



MANUAL DE USUARIO

Documentación orientada a gestionar la Base de Datos, solventar los errores en la ejecución del programa e interpretar la codificación en Java.

Versión 1.0



Por Miguel Olmo Hernando



GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

1. PRESENTACIÓN

Este manual presenta una propuesta de la estructura desarrollada en el programa elaborado por el Equipo 6, cuyo integrante en solitario Miguel Olmo Hernando es un estudiante de Primero de Desarrollo de Aplicaciones Web del Instituto Egibide Arriaga de Vitoria-Gasteiz (Araba).

- **Entre los objetivos que persigue esta propuesta se tienen tres puntos clave:**
 - Identificar El Problema - Enunciado:
 - Definir Y Representar El Problema Del Enunciado - Fases Del Proyecto:
 - Temporización - Rúbricas - Criterios De Evaluación Del Proyecto:
- **Para completar el manual tienen a su disposición una galería de imágenes de mayor calidad y resolución en el repositorio de GitHub, carpeta Imágenes. De igual forma pueden acceder al Blog en el que se ha generado semanalmente entradas, en las cuales puede ir viendo la evolución del proyecto.**
 - poner link mig
- **A destacar en este manual:**
 - Se desarrolla mediante la siguiente estructura:

https://es.wikipedia.org/wiki/Partes_del_libro

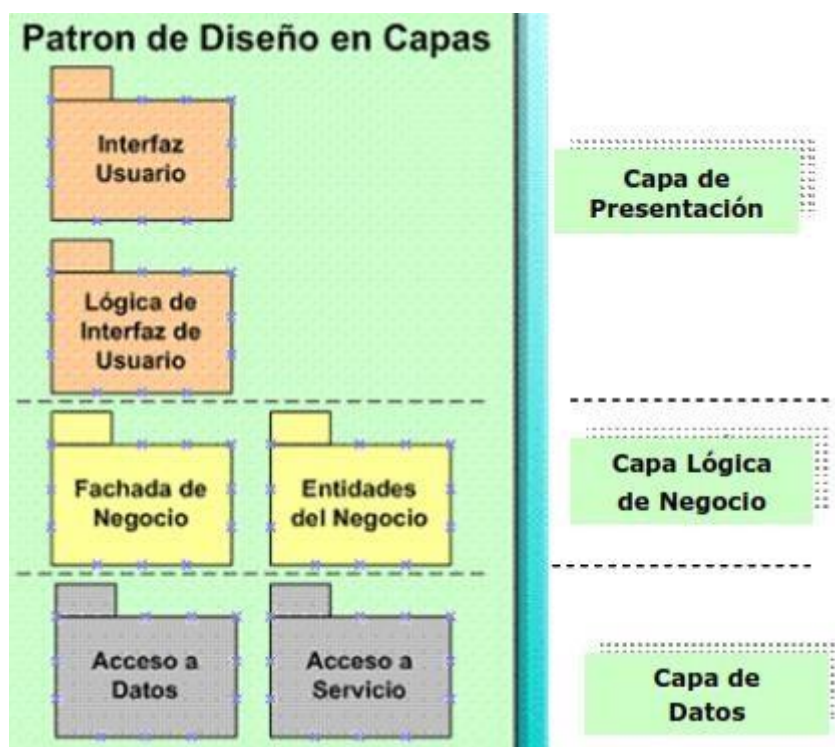
 - Esta creado mediante las herramientas de Microsoft Office 2016 y NitroProPDF. Si tienen problemas con la plantilla de Word utilicen el PDF para este visionado. Recuerden que en este caso puede ser que no tengan disponible el índice multimedia generado.

2. INTRODUCCIÓN

2.1. ARQUITECTURA DE LA APLICACIÓN

Uno de los patrones arquitectónicos más usados en la actualidad para el desarrollo de aplicaciones de porte empresarial es el denominado “arquitectura de 3 o más capas”.

Este estilo arquitectural separa en algunos casos físicamente, los aspectos de presentación de la aplicación (interfaz de usuario), la lógica de negocios (automatización del flujo trabajo) y la gestión de los datos (bases de datos), tal como se muestra en la siguiente imagen:



La capa de presentación es la encargada de manejar la interfaz del usuario, controlando la captura y presentación de los datos y recibiendo los eventos accionados por los usuarios a través de la interfaz.

Esta capa se comunica únicamente con la capa lógica de negocios.

La capa lógica de negocios tiene la responsabilidad de manejar la funcionalidad del sistema, implementando a través de objetos de negocio las reglas de negocio que deben cumplirse. Esta capa se comunica con la capa de presentación para recibir solicitudes y presentar resultados y con la capa de datos, para solicitar al gestor de base de datos que almacene o recupere datos.

GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

La capa de datos (llamada en algunos casos capa de persistencia) es la responsable del almacenamiento y recuperación de los datos. Se comunica únicamente con la capa de lógica de negocios.

Un ejemplo de la descripción y estructura de la arquitectura para nuestra aplicación es la siguiente:



El sistema empleará el estilo arquitectónico de capas y será organizado en tres capas: la capa de interfaz, la capa de la aplicación y la capa de almacenamiento.

La capa de interfaz contendrá la interfaz gráfica del usuario que le permitirá a los usuarios interactuar con la aplicación. Esta capa será implementada usando principalmente Java y gracias al software NetBeans.

La capa de la aplicación contendrá la lógica y reglas para almacenar datos en la capa de la base de datos y también para recuperar éstos de acuerdo con las necesidades del usuario.

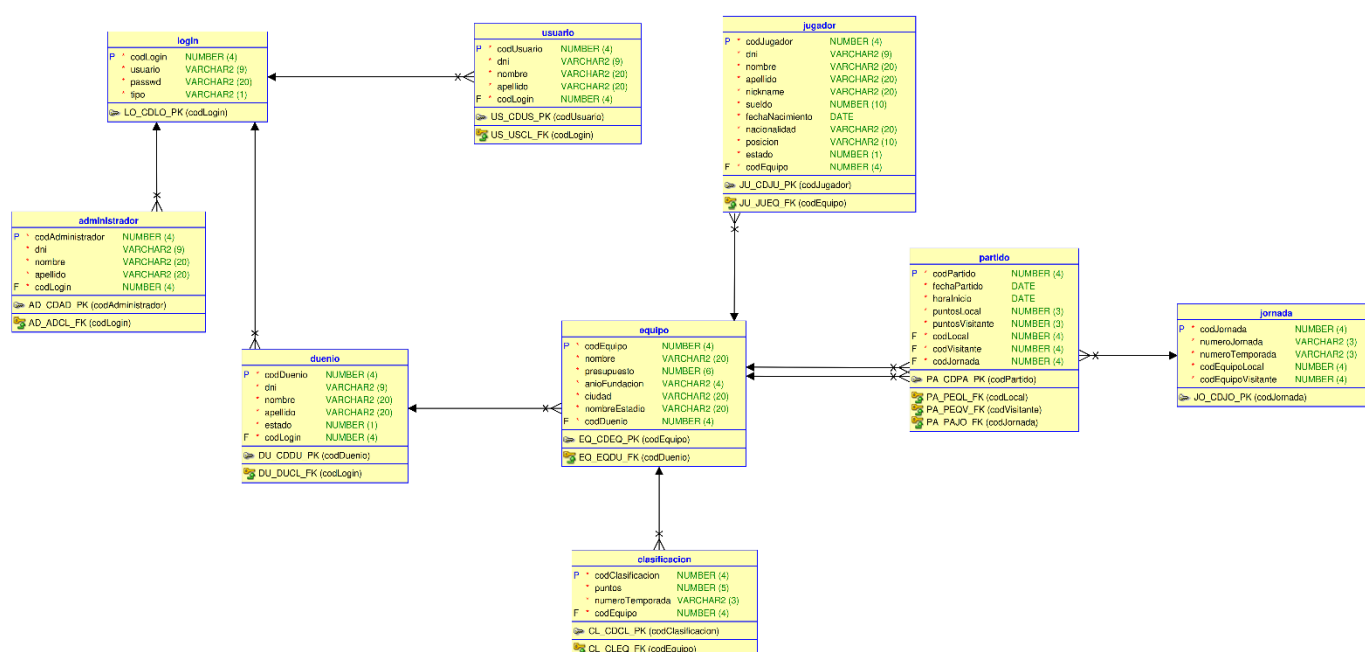
Finalmente, la capa de almacenamiento guardará los datos requeridos por el sistema. Para la gestión de la Base de Datos usamos Vagrant que es quien nos permite generar un entorno de desarrollo mediante una Máquina Virtual en la que a posteriori junto con SQL Developer insertaremos las tablas, procedimientos, paquetes y disparadores necesarios para generar, obtener o almacenar los datos.

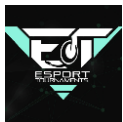
GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

2.1.1. MODELO UML

Se presenta a continuación el modelado de la aplicación, basado en la metodología UML. UML (Lenguaje de Modelado Unificado) es una especificación de notación orientada a objetos, el cual se compone de diferentes diagramas, los cuales representan las diferentes etapas del desarrollo del proyecto.

La aplicación usada para generar este modelo es Data Modeler.



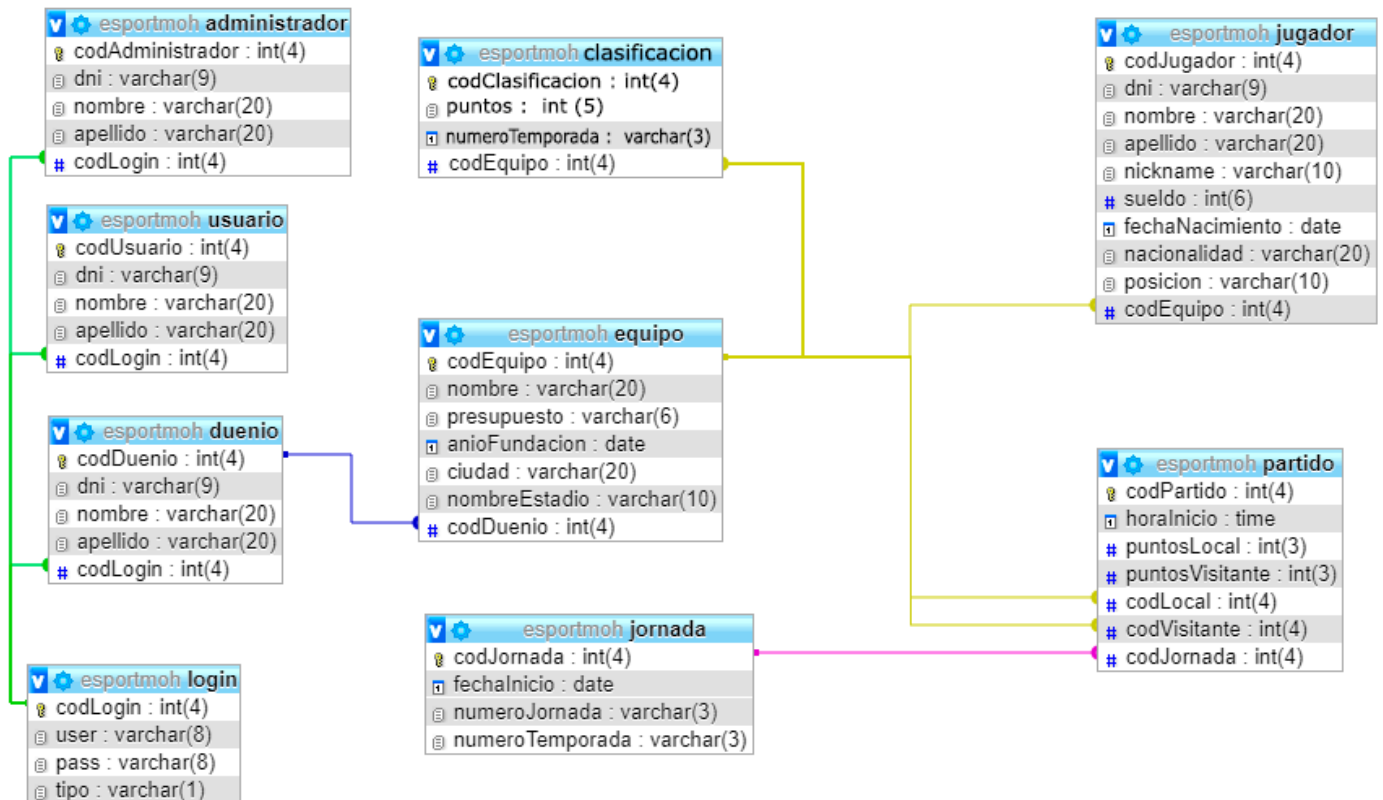


GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

2.1.2. MODELO MER

Turno del Diagrama de Entidad Relación, gracias a este diagrama representamos las entidades relevantes de nuestra aplicación y obtenemos así la información de sus interrelaciones y propiedades.

La aplicación usada para generar este modelo es Data Modeler.

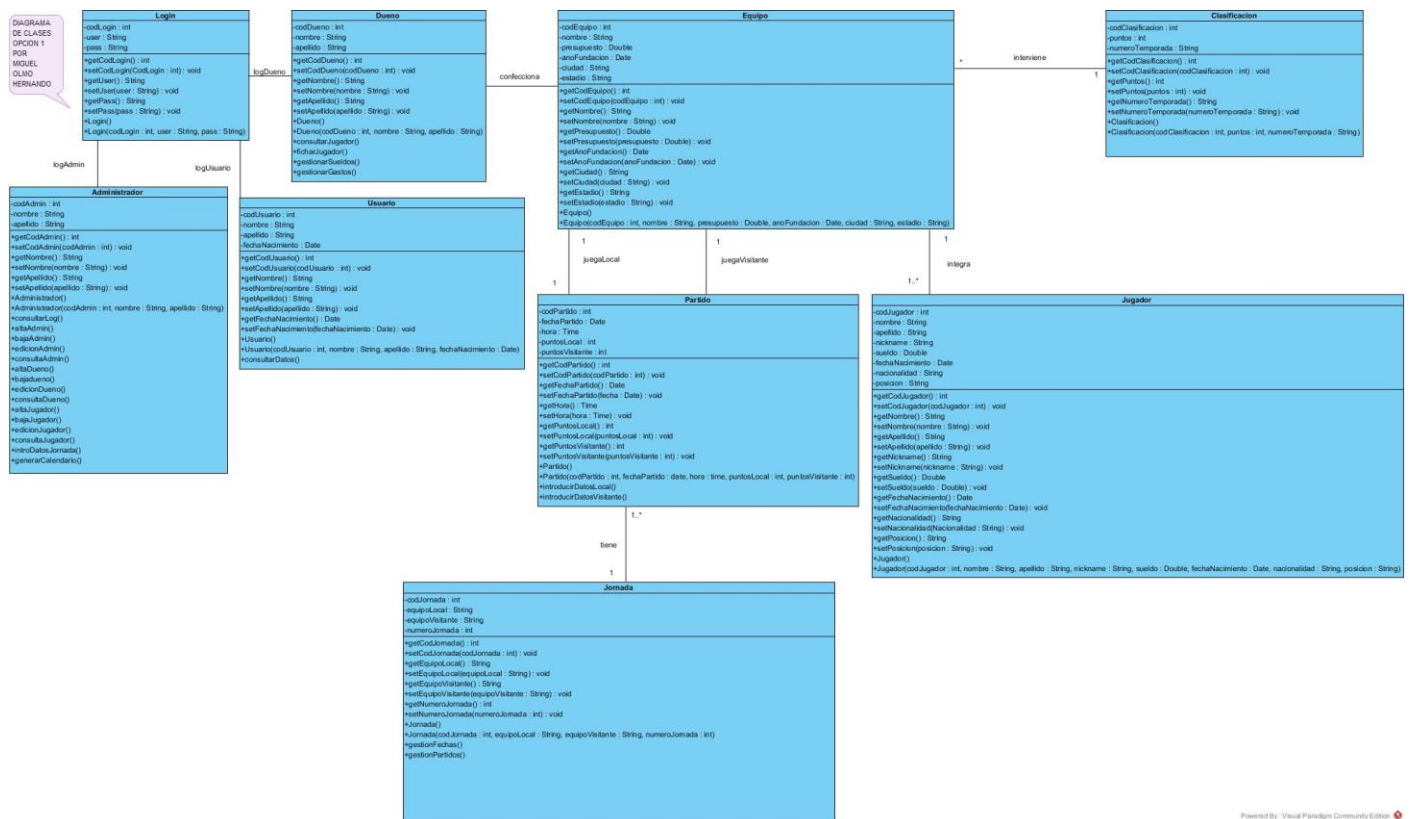


GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

2.1.3. DIAGRAMA DE CLASES

Generamos mediante la aplicación Visual Paradigm este diagrama de clases que nos permitirá representar gráficamente y de manera estática la estructura general de nuestra aplicación, mostrando cada una de las clases y sus interacciones (como herencias, asociaciones, etc.), representadas en forma de bloques, los cuales son unidos mediante líneas y arcos.

Este diagrama es fundamental del modelado con UML, siendo ampliamente utilizados tanto para análisis como para diseño de sistemas y software en general.



- **NOTA IMPORTANTE:** Alguno de los atributos de las tablas han podido cambiar ligeramente, un ejemplo son unos nuevos, los estados, tanto en la tabla jugadores como en la de dueños. También he quitado la hora de la jornada que no me ha dado tiempo a codificar.



GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

3. ÍNDICE

1.	PRESENTACIÓN	2
2.	INTRODUCCIÓN.....	3
2.1.	ARQUITECTURA DE LA APLICACIÓN	3
2.1.1.	MODELO UML.....	5
2.1.2.	MODELO MER.....	6
2.1.3.	DIAGRAMA DE CLASES.....	7
3.	ÍNDICE	8
4.	INSTALACIÓN DEL SOFTWARE.....	9
4.1.	VAGRANT.....	9
4.2.	SQL DEVELOPER	11
4.3.	NETBEANS	13
5.	GESTIÓN Y CARGA DE LA BASE DE DATOS	14
5.1.	EJECUTAR SCRIPT ORACLE.....	15
5.2.	CARGAR LOS PAQUETES Y PROCEDIMIENTOS.....	16
5.3.	CARGAR LOS TRIGGERS / DISPARADORES.....	18
5.4.	PAQUETE – TRIGGER - CONTROLAR EL PRESUPUESTO DEL EQUIPO - MUTANTES.....	19
6.	GESTIÓN PREVIA DEL PROGRAMA	21
6.1.	ABRIR EL PROYECTO CON NETBEANS	21
6.2.	SOLUCIONAR PROBLEMAS – LIBRERIAS	22
6.3.	EDITAR LOS PARAMETROS DE CONEXIÓN.....	26
6.4.	MOSTRANDO LA ESTRUCTURA INTERNA DEL PROYECTO	27
7.	PROGRAMA EN EJECUCIÓN	28
7.1.	GESTIÓN DE ACCESO AL PRGRAMA.....	28
7.2.	IDENTIFICACIÓN SEGÚN EL TIPO DE LOGIN INSERTADO	29
7.2.1.	ADMINISTRADORES	29
7.2.1.1.	PANELES CRUD – ALTAS – BAJAS – CONSULTAS - MODIFICACIONES.....	30
7.2.1.2.	PANELES DE GESTIÓN DE TEMPORADAS	31
7.2.1.3.	PANELES DE GESTIÓN DE CLASIFICACIONES	32
7.2.1.4.	PANELES DE GESTION PARTIDO / INSERTAR PUNTOS.....	33
7.2.2.	DUEÑOS.....	35
7.2.3.	USUARIOS	36
8.	BIBLIOGRAFIA.....	37
9.	EPÍLOGO	38
7.3.	EN EL TINTERO	38
7.4.	CONCLUSIÓN.....	38



4. INSTALACIÓN DEL SOFTWARE

4.1. VAGRANT

Antes de ponernos con la Interfaz Gráfica de nuestra Aplicación y explicar cómo navegar a través de ellas, es imprescindible comentar este punto, sin tener las herramientas que se requieren, no podrán ejecutar correctamente la aplicación.

Este proceso, se resumen en cinco puntos clave:

- 1) Deben de acceder al siguiente enlace y seguir los pasos que se muestran para poder **instalar Vagrant** <https://github.com/Egibide/vagrant-oracle-12-ee-linux>
- Si en este punto tienen problemas al cargar el Vagrant fuera de Egibide, eliminen la siguiente línea del archivo Vagrantfile. Utilicen un editor de texto como Notepad++

Buscar y quitar:

```
db12102.vm.box_url = ["http://nunki.diocesanas.org/mv/vagrant/centos-6.5-x86_64.box",  
"https://dl.dropboxusercontent.com/s/np39xdpw05wfmv4/centos-6.5-x86_64.box"]
```

Para dejar la siguiente:

```
db12102.vm.box_url = ["https://dl.dropboxusercontent.com/s/np39xdpw05wfmv4/centos-  
6.5-x86_64.box"]
```

- Quitando la URL de nunki.diocesanas está comprobado que CentOS se descarga antes y no genera errores si se pretende instalar fuera de las instalaciones de Egibide.
- Una vez tengan realizados todos los pasos que se indican en el link del repositorio anteriormente mostrado: <https://github.com/Egibide/vagrant-oracle-12-ee-linux> ya podrán ejecutar Vagrant muy fácilmente:

```
Directorio de D:\INSTALACIONES\MAQUINA_VIRTUAL\vagrant-oracle-12-ee-linux-master  
24/01/2017 23:22 <DIR> .  
24/01/2017 23:22 <DIR> ..  
11/09/2016 02:35 26 .gitignore  
24/01/2017 23:22 <DIR> .vagrant  
24/01/2017 23:54 <DIR> oracle12ee  
11/09/2016 02:35 <DIR> puppet  
11/09/2016 02:35 2.200 README.md  
11/09/2016 02:35 1.635 Vagrantfile  
3 archivos 3.802 bytes  
5 dirs 392.387.780.608 bytes libres  
D:\INSTALACIONES\MAQUINA_VIRTUAL\vagrant-oracle-12-ee-linux-master>vagrant up
```

GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

- Comprueben mediante el comando *vagrant status* que efectivamente la máquina está cargando:

```
D:\INSTALACIONES\MAQUINA_VIRTUAL\vagrant-oracle-12-ee-linux-master>vagrant status
Current machine states:

b12102                running (virtualbox)

The VM is running. To stop this VM, you can run `vagrant halt` to
shut it down forcefully, or you can run `vagrant suspend` to simply
suspend the virtual machine. In either case, to restart it again,
simply run `vagrant up`.

D:\INSTALACIONES\MAQUINA_VIRTUAL\vagrant-oracle-12-ee-linux-master>_
```

- Si desean parar Vagrant, siguiendo los pasos indicados (*vagrant halt*) es igual de sencillo:

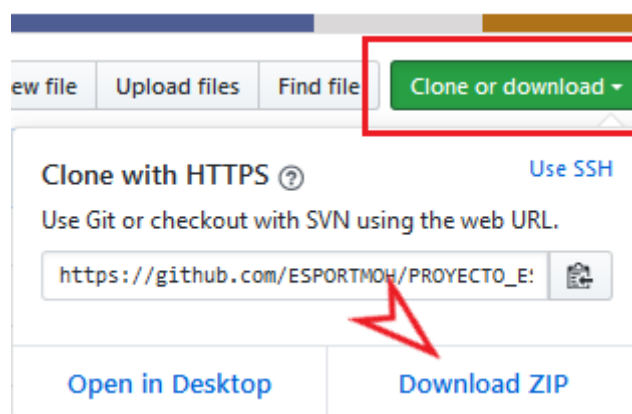
```
D:\INSTALACIONES\MAQUINA_VIRTUAL\vagrant-oracle-12-ee-linux-master>vagrant halt
==> db12102: Attempting graceful shutdown of VM...
db12102: Guest communication could not be established! This is usually because
db12102: SSH is not running, the authentication information was changed,
db12102: or some other networking issue. Vagrant will force halt, if
db12102: capable.
==> db12102: Forcing shutdown of VM...

D:\INSTALACIONES\MAQUINA_VIRTUAL\vagrant-oracle-12-ee-linux-master>vagrant status
Current machine states:

b12102                poweroff (virtualbox)

The VM is powered off. To restart the VM, simply run `vagrant up`
```

- Llegados a este punto, accedan al repositorio y comiencen su descarga. Esta descarga se recomienda gestionar en una carpeta previamente creada en el escritorio o el home de su Sistema Operativo, luego pueden descomprimirlo y tener acceso a todo el proyecto.



- Link al Repositorio de la Organización E-SPORT:



https://github.com/ESPORTMOH/PROYECTO_ESPORT_G6

- ## PROYECTO_ESPORT_G6
- Por Miguel Oliva Hernandez
- ### NOVEDADES
- Esta versión Paso 4 V7 Incluye lo siguiente:
 - Se ha codificado lo necesario para la autogeneración de las temporadas/jornadas.
 - Todos los CRUDs fundanados (pero a falta de unos pocos registros).
 - Creación del layout.
 - Se ha mejorado la interfaz gráfica.
 - Codificada la plantilla y Perfilado (Actualmente trabajando en ello).
 - Trabajando en los diferentes manuales y la programación.
- ### CONTENIDO DEL REPOSITORIO
- La Carpeta **BASES_DE_DATOS** contiene las siguientes:
 - **DIAGRAMAS:** Igual se encuentra el Diagrama de Clase, el UML y el Modelo Relacional, se irán actualizando en función a los cambios que vaya realizando en la codificación del programa. Se incluyen los editables de estos.
 - **PROCEDURES:** Contiene los dos **PROCEDURES** que se piden para el proyecto y otros. Se incluyen los editables de estos.
 - **SCRIPTS_ORACLE:** Contiene el **Script SQL** para crear la base de datos mediante **Oracle SQL Developer**. Este genera la base de datos utilizada en el proyecto. Se incluye el **Script SQL** para ser cargado mediante **MySQL** al servidor.
 - **TRIGGERS:** Contiene los dos **TRIGGERS** que se piden para el proyecto y otros. Se incluyen los editables de estos.
 - La Carpeta **WIDGETS** contiene cuando las imágenes usadas en WordPress a modo de logo de seguridad. Pueden editar el blog mediante el siguiente link [Proyecto de seguridad](#).
 - La Carpeta **CUON_INTRODUCCION** es donde se ha creado el protocolo que han dado de copyright. Se incluye una copia del programa **USWIDGETS**, esta es una versión que no regula de permisos y que funciona perfectamente en el aula del centro. **NOTA:** las versiones más nuevas de este no funcionan.
 - La Carpeta **ENUNCIADO** tiene una copia actualizada del mismo.
 - La Carpeta **PROGRAMACION_JAVA** es donde tiene lugar la codificación del programa. Esta codificación la realice mediante **Netbeans**.
- ### REQUISITOS PARA EJECUTAR EL PROGRAMA
- [1ª.-] Descargarse el navegador o donarlo en tu PC (brillio o descargarlo, etc).
 - [2ª.-] Realizar las siguientes pasos para la instalación de **Oracle SQL Developer** en **Virtual**.
 - [3ª.-] Cargar **SQL Developer** para generar la Base de datos mediante el **SCRIPT SQL** de **ORACLE**, cargar los **PROCEDURES** y los **TRIGGERS** mencionados anteriormente.
 - [4ª.-] Instalar **Netbeans**.
 - [5ª.-] Ejecutar **Netbeans** e importar **MyCalendar** y **MyLibrary**.
 - [6ª.-] Editar la Clase **Generador** con los datos que utilizas en el paso 2ª.
 - [7ª.-] Si quieren ampliar el funcionamiento de la aplicación al servidor, realizar la siguiente carpeta:
 - **WIDGETS** Esta contiene la documentación de la API en formato **HTML**. Si abren el **INDEX HTML** con un Navegador Web de su elección, le permitirá analizar la estructura de la aplicación. Detallar que la aplicación se encuentra con comentarios, procedimientos, funciones, datos y métodos desarrollados de tal forma que explican al detalle que hace cada uno y cómo.
 - [8ª.-] Ejecutar el programa. Si usuario y clave por defecto del usuario **Administrador** con **root / root**.
-

2) Ahora requieren de la herramienta SQLDeveloper de Oracle, pueden seguir los pasos del punto 1) o bien pueden descargarla mediante otros medios.

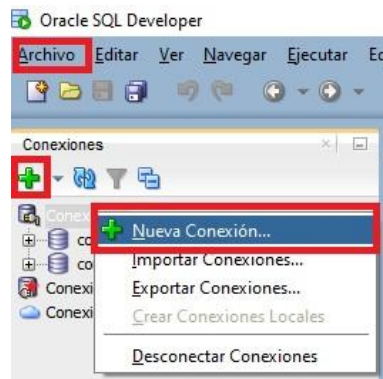
<http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>

Con los pasos 1), 2) terminados y estando Vagrant ejecutado, ahora es necesario mediante SQL Developer crear una conexión, pueden al igual que en el punto 2) usar los datos de conexión que se muestran o bien generar otros, lo que sí tienen que tener muy en cuenta es recordar dichos datos, pues luego nuestra

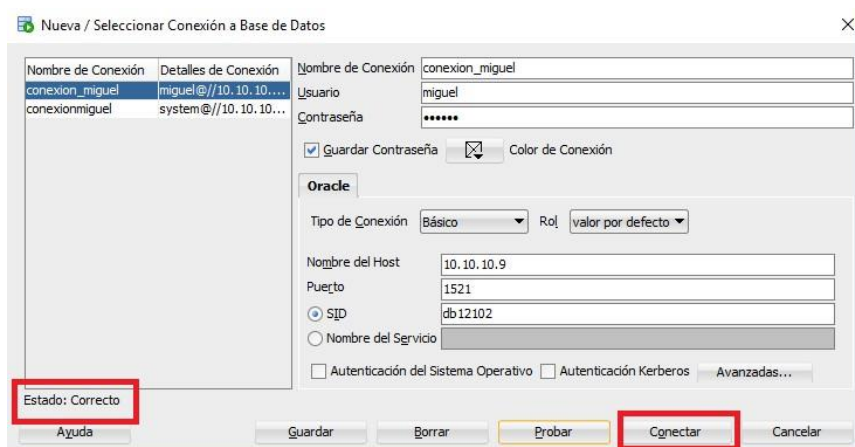
Aplicación requerirá de ellos para conectarnos a la Base de Datos. Si ya tienen una conexión establecida con anterioridad o ya conoce el funcionamiento de este punto, diríjase al siguiente apartado – **Gestión y Carga de la Base de Datos** -, si no continúe.

GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

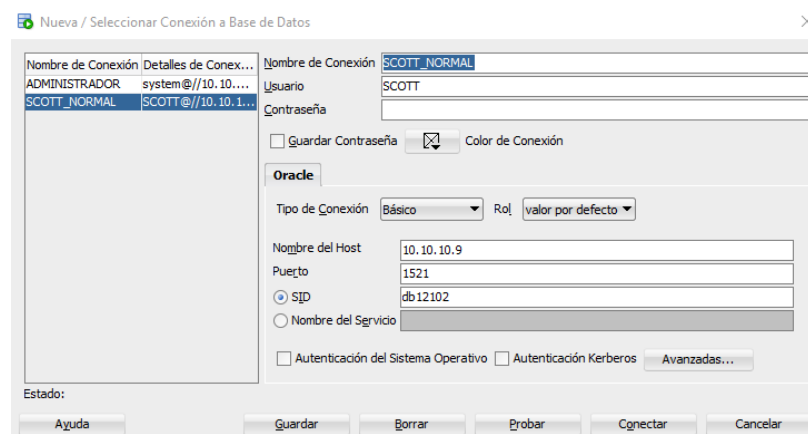
- Abrimos SQL Developer y tenemos varias formas de crear una conexión, desde archivo nueva conexión, desde el + verde o bien haciendo clic en el botón derecho del ratón y Nueva Conexión.

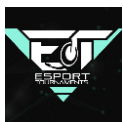


- Introduzca los datos de su nueva conexión, recuerde que inicialmente tiene que entrar como usuario administrador así que ojo con lo que hacen. Rellenados los campos, pulsar el botón Probar y así podréis ver si os funciona, si funciona conectarlos. Una vez acceden como usuario Administrador pueden crear un nuevo usuario con menos permisos y evitar riesgos.



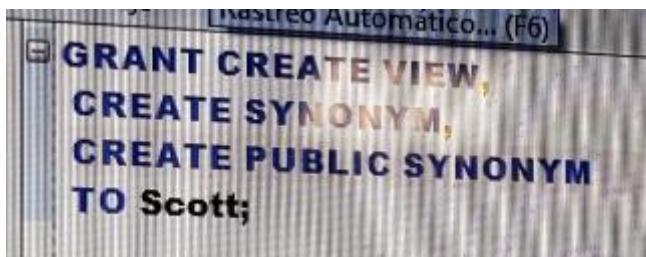
- Nota: La contraseña de sys y system es oracle.





GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

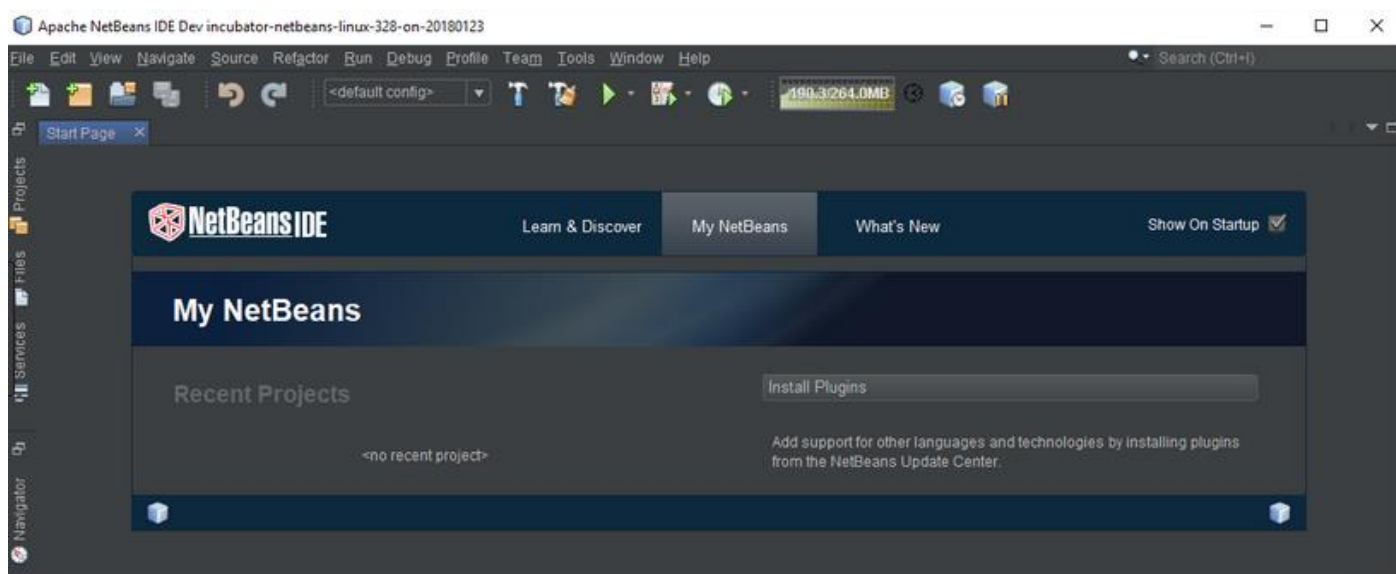
- 3) Ultimo paso, cargar nuestra Base de Datos, paquetes, procedimientos y disparadores en SQL Developer. Para este paso se recomienda generar un nuevo usuario y no utilizar el System, en este caso se ha generado el de SCOTT.



- Ejemplo de cómo dar permisos a un usuario de SQL Developer
- Como crear un nuevo usuario: <http://cursoanpefernandoortiz.blogspot.com.es/2014/12/crear-un-nuevo-usuario-oracle-con-sql.html>

4.3. NETBEANS

- 4) Es el momento de instalar NetBeans, para ello accedan a la Web para su descarga y sigan los pasos que se indican:
- Link descarga: <https://netbeans.org/>
 - Manual de instalación: <https://netbeans.org/community/releases/82/install.html>





5. GESTIÓN Y CARGA DE LA BASE DE DATOS

- Si ya disponen de una conexión con anterioridad y han pasado a este punto sin necesidad de hacer los anteriores. - o bien quiere cargar de nuevo nuestra Base de Datos le recomendamos borre cualquier tabla que tenga creada antes. En caso de tener la misma, el Script se encargará de ello por sí solo, pero os mostro como a cómo hacerlo de forma manual.
- Diríjanse a la carpeta en donde descargaron y descomprimieron el repositorio, localicen la siguiente carpeta *BASES_DE_DATOS* en ella se ubican las siguientes:



- DIAGRAMAS, contiene los diferentes diagramas mencionados anteriormente y sus editables.
- PROCEDURES, contiene los Paquetes y Procedimientos que se cargaran en la Base de Datos, también se incluyen pruebas y editables.
- SCRIPTS_ORACLE, contiene el fichero DDL de ORACLE que nos permitirá crear la Base de Datos, se incluye otro para MySQL que no usaremos en este caso y editables y pruebas y comprobaciones varias.
- TRIGGERS, contiene los disparadores que se requieren para el correcto funcionamiento de la Base de Datos, de igual forma se incluyen los editables y pruebas varias.

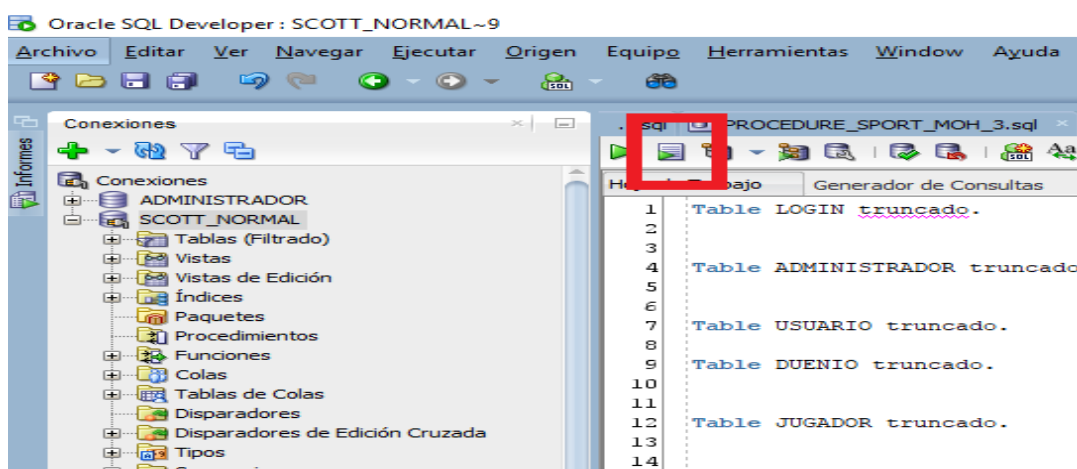


GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

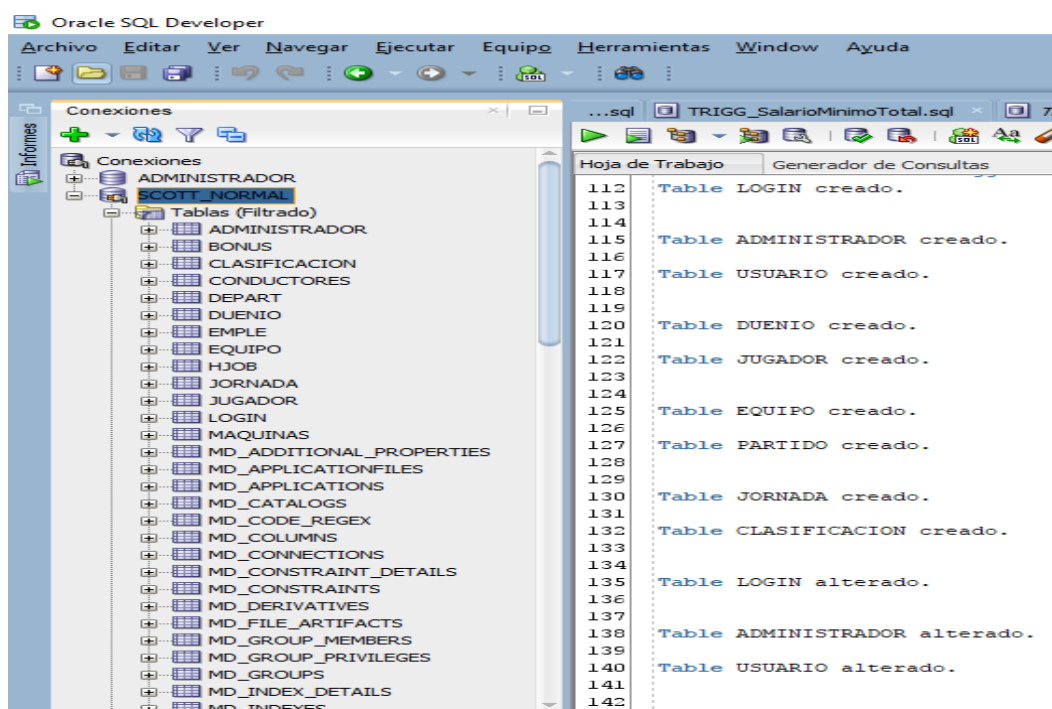
5.1. EJECUTAR SCRIPT ORACLE

Entramos en la carpeta *SCRIPTS_ORACLE*, seleccionamos el *ScriptDDL_ESPORT_ORACLE_OK_MOH* y lo arrastramos para soltarlo en el SQL Developer (recordamos que la conexión a de establecerse como mencionamos en pasos anteriores para que este paso nos funcione).

Una vez lo tengamos, simplemente ejecutamos el Script (ver icono remarcado en rojo). Inicialmente nos borra cualquier tabla que pueda llamarse igual a las que ya pueda tener en su BD, por lo que deberá de revisar este si quiere mantener su información. De igual forma pasa con los paquetes, procedimientos y disparadores.



Este Script genera todas las tablas, restricciones e inserciones de datos necesarias para cargar a posteriori el programa con un contenido base con el que poder empezar a trabajar.



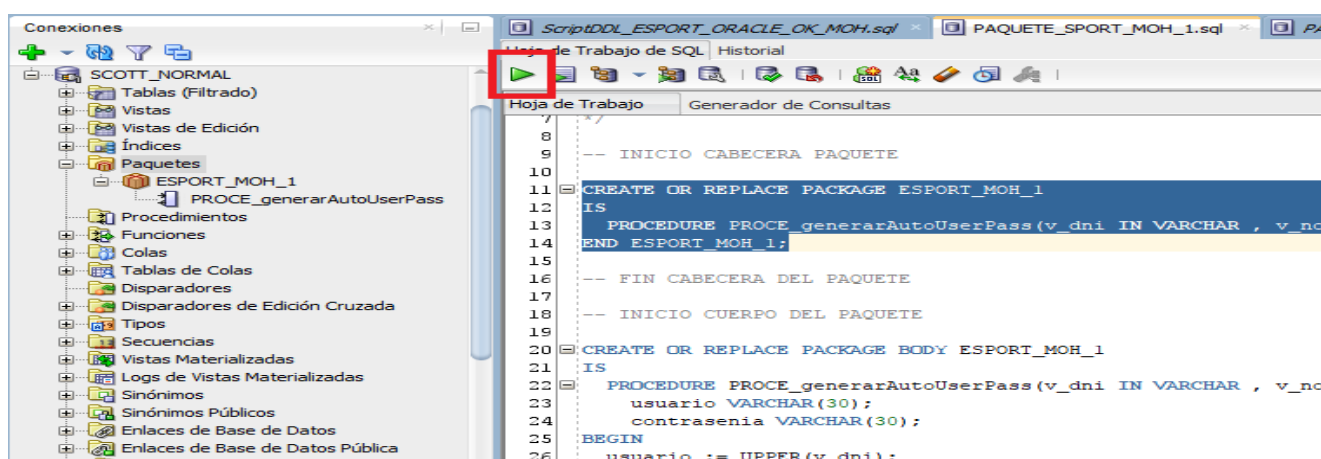
GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

5.2. CARGAR LOS PAQUETES Y PROCEDIMIENTOS

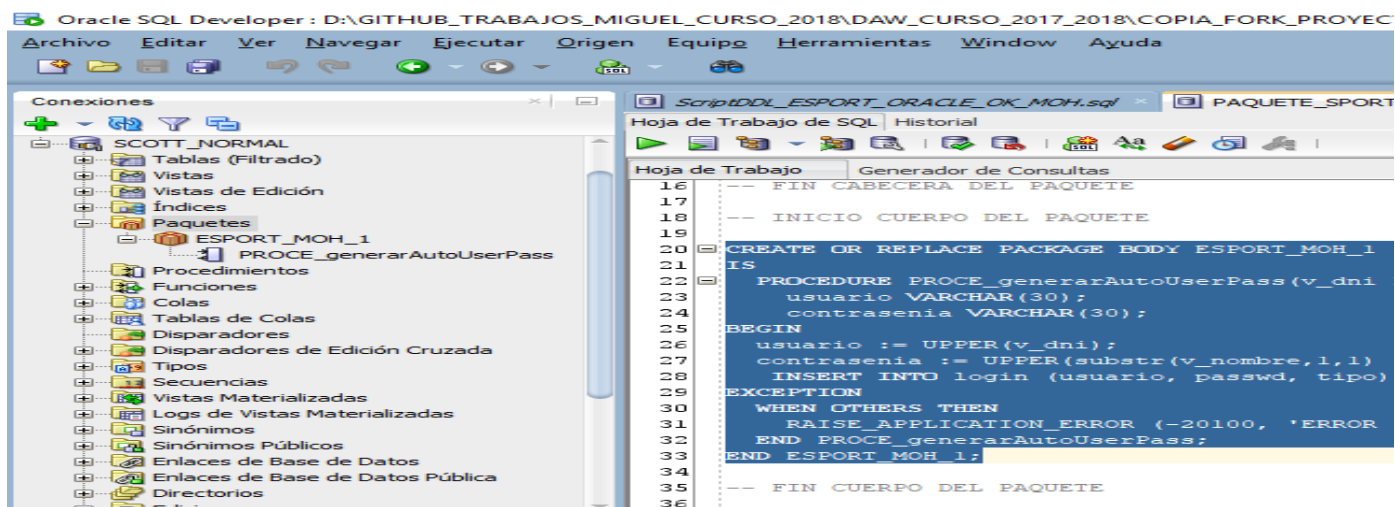
No vamos a complicarnos y como anteriormente, nos vamos a la carpeta correspondiente *PROCEDURES*, seleccionamos el *PAQUETE_SPORT_MOH_1*, *PAQUETE_SPORT_MOH_2* y el *PROCEDURE_SPORT_MOH_3*, los arrastramos y soltamos en el SQL Developer.

A continuación, vamos a cargar estos en nuestra BD, pueden empezar por el que quieran, os mostramos como cargar un paquete y un procedimiento.

En el caso de los Paquetes, cargamos primero la **cabecera**, la seleccionamos y le damos al icono remarcado en rojo.

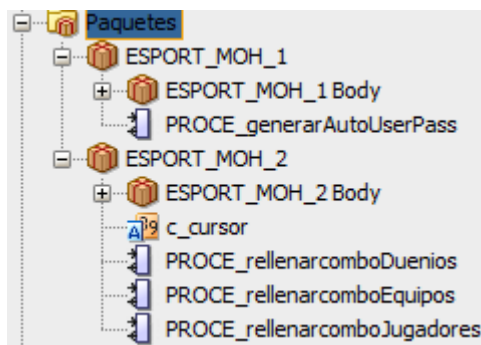


Después hacemos lo mismo, pero con el **cuerpo** del paquete.



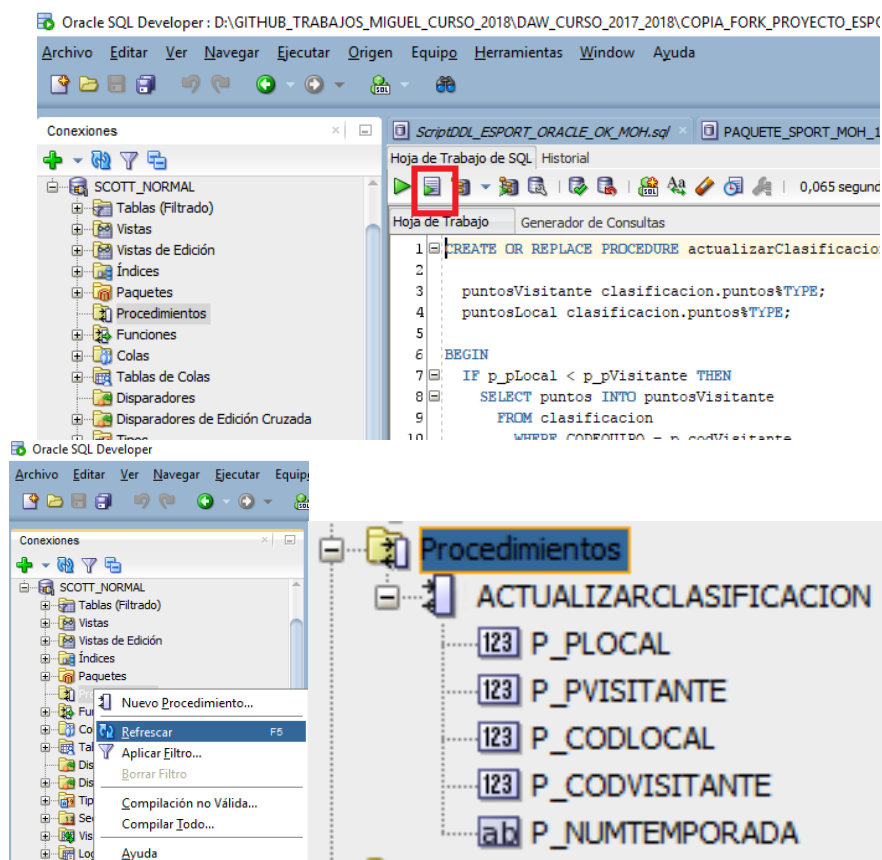
GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

Por último, comprobamos que se han cargado correctamente en nuestra Base de Datos.



- Este paso lo vamos a realizar con los dos Paquetes que se muestran.
- El paquete *PAQUETE_SPORT_MOH_1* es quien nos va a generar a partir del alta de un administrador, dueño, usuario y su DNI, nombre y apellido un Login cuyo usuario será el DNI y el Login la concatenación de la primera letra de su nombre y el apellido completo.

Para cargar el Procedimiento, basta con ejecutarlo, no se requiere de una cabecera o cuerpo.



- Si no ven a la primera que los paquetes o procedimientos están cargados, refresque.

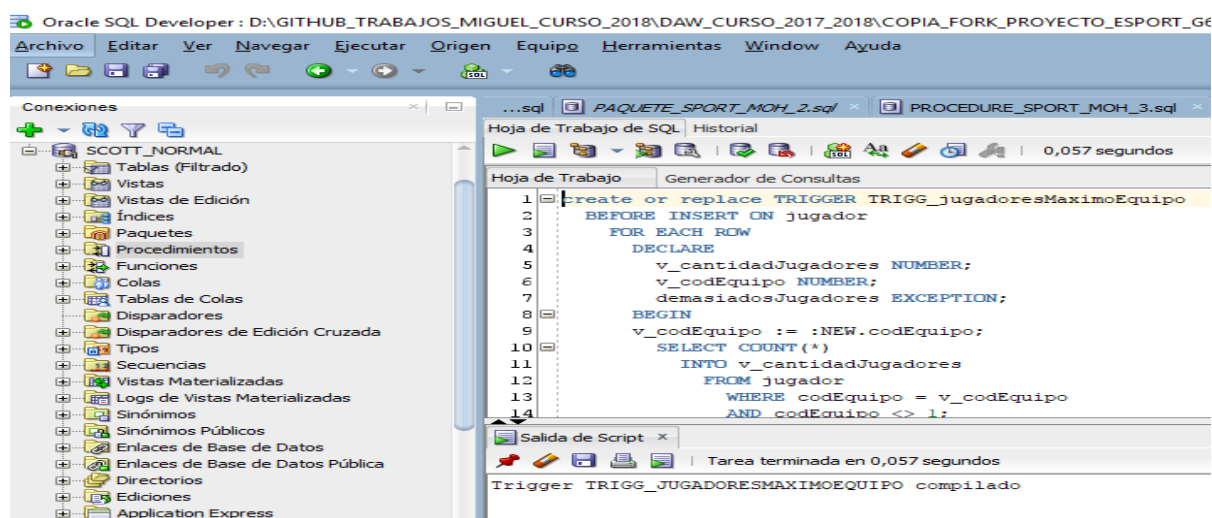


GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

5.3. CARGAR LOS TRIGGERS / DISPARADORES

Vamos a la carpeta *TRIGGERS* y como con los paquetes, seleccionamos su contenido *TRIGG_JugadoresMaximoEquipo* y *TRIGG_SalarioMinimoTotal* y los arrastramos y soltamos en el SQL Developer.

Ejecutamos *TRIGG_JugadoresMaximoEquipo* y podemos ver que este se compila perfectamente.





GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

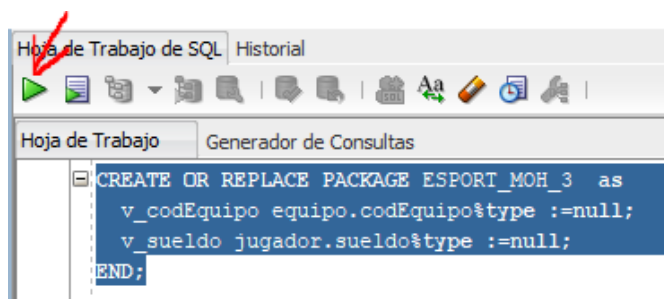
5.4. PAQUETE - TRIGGER - CONTROLAR EL PRESUPUESTO DEL EQUIPO - MUTANTES

Este punto se ha decidido dejar separado de los demás, puesto que el fichero *TRIGG_SalarioMinimoTotal* nos permite controlar que la Tabla Equipo no Mute. ¿Qué quiere decir, Mutante? Una tabla mutante es una tabla que está siendo modificada. Si un 'trigger' hace referencia a la propia tabla sobre la que se dispara se producirá una excepción con error ORA-04091.

Según esto, cualquier 'Trigger' que haga una simple consulta sobre la tabla que lo dispara generará dicho ORA-04091.

¿Qué hacemos para controlarlo? Para controlarlo es necesario utilizar una colección definida en un paquete. Luego demuestro cómo usar el disparador compuesto y agregarlo en nuestra Base de Datos, para resolver el problema de forma mucho más simple:

Con el *TRIGG_SalarioMinimoTotal* en SQL Developer vamos a seleccionar y ejecutar el Paquete *ESPORT_MOH3* como se muestra en la siguiente imagen:



Luego hacemos lo mismo con el Trigger llamado *jugadorMutante*:

```
CREATE OR REPLACE TRIGGER jugadorMutante
AFTER INSERT OR UPDATE ON jugador
FOR EACH ROW
BEGIN
    ESport_MOH_3 .v_codEquipo := :new.codEquipo;
END jugadorMutante;
```

Y con el Trigger *sueldoMutante*:

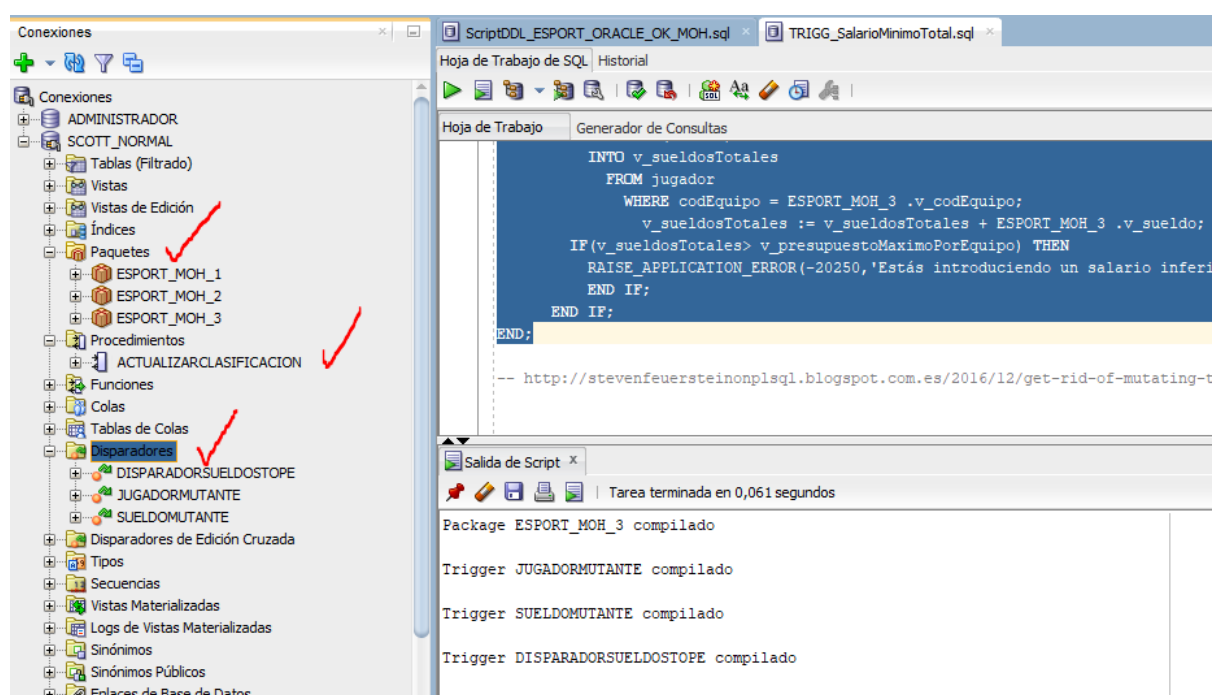
```
CREATE OR REPLACE TRIGGER sueldoMutante
AFTER INSERT OR UPDATE ON jugador
FOR EACH ROW
BEGIN
    ESport_MOH_3 .v_sueldo := :new.sueldo;
END sueldoMutante;
```

GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

Por último, cargamos el Trigger *disparadorSueldosTope* que nos controlará la suma total de los salarios de los jugadores con el presupuesto máximo de 200000€ del equipo, llegados a este máximo el Dueño no podrá fichar un jugador.

```
CREATE OR REPLACE TRIGGER disparadorSueldosTope
BEFORE INSERT on jugador
FOR EACH ROW
DECLARE
    v_sueldosTotales NUMBER;
    v_presupuestoMaximoPorEquipo NUMBER:=200000;
    v_error EXCEPTION;
BEGIN
    IF ESPORT_MOH_3 .v_codEquipo <> 1 THEN
        SELECT SUM(sueldo)
        INTO v_sueldosTotales
        FROM jugador
        WHERE codEquipo = ESPORT_MOH_3 .v_codEquipo;
        v_sueldosTotales := v_sueldosTotales + ESPORT_MOH_3 .v_sueldo;
        IF(v_sueldosTotales> v_presupuestoMaximoPorEquipo) THEN
            RAISE_APPLICATION_ERROR(-20250,'Estás introduciendo un salario inferior al salario mínimo');
        END IF;
    END IF;
END;
```

Finalmente podemos ver que se han generado e insertado correctamente en nuestra Base de Datos:



Los Triggers saltarán automáticamente en nuestra Base de Datos según la tabla en la que intervengan, dicho mensaje de error se verá reflejado en la aplicación. Los procedimientos y paquetes son los que llamaremos directamente al cargar una función de la aplicación usando la **Interfaz CallableStatement**.

```
CallableStatement cS = con.prepareCall("{call ESPORT_MOH_2.PROCE_rellenarcomboDuenios(?)}");
```

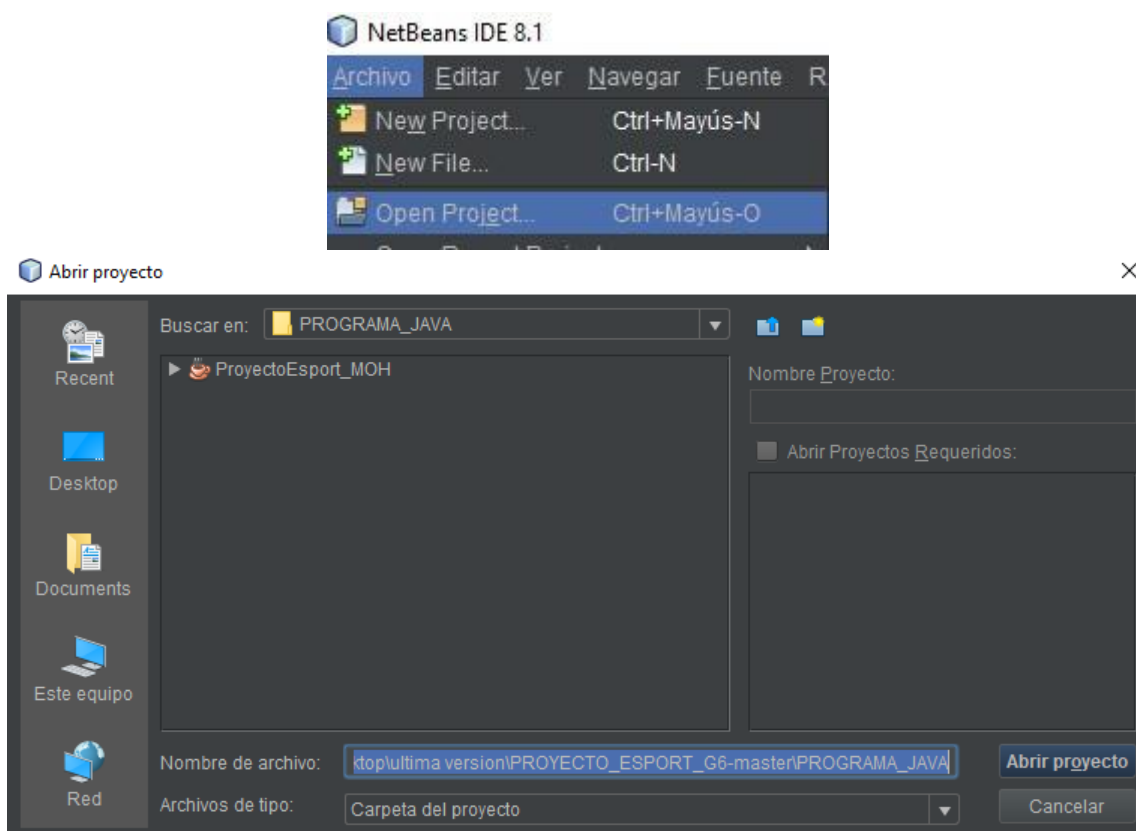
- Ejemplo de llamada al procedimiento ESPORT_MOH_2

6. GESTIÓN PREVIA DEL PROGRAMA

6.1. ABRIR EL PROYECTO CON NETBEANS

Ya con nuestra Base de Datos preparada, solo nos queda ejecutar el programa creado en Java, en este caso y como se menciona al principio de este manual, usaremos Netbeans como plataforma de carga.

Ejecutamos Netbeans y vamos a abrir el Proyecto.



Como se ve en la imagen, el Proyecto (*ProyectoEsport_MOH*) se encuentra en la carpeta PROGRAMA_JAVA del repositorio extraído.

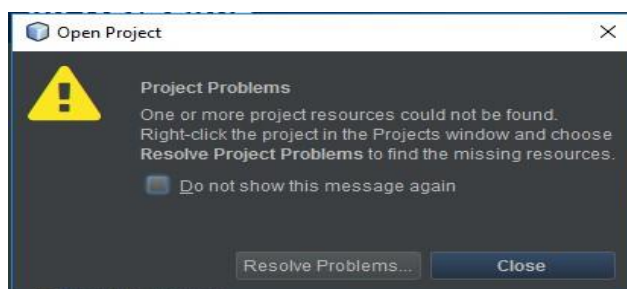
- Si se van a realizar cambios en el proyecto, es de vital importancia el generar un fichero gitignore para decirle a Git qué archivos o directorios completos debe ignorar y no subir al repositorio de código. Su implementación es muy sencilla, por lo que no hay motivo para no usarlo y para ello en el Blog aportado al proyecto se ha dejado un manual a modo de que puedan crear dicho archivo y de cómo poder usar GitHub y GitKraken:

<https://proyectodawsport.wordpress.com/2018/04/20/crear-organizacion-fork-gitignore/>

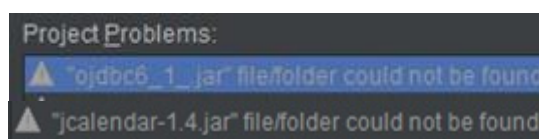
6.2. SOLUCIONAR PROBLEMAS – LIBRERIAS

La primera vez que abramos el proyecto, Netbeans nos mostrara un error, este indica que algunas de las librerías que se han usado para codificar el programa no las tiene referenciadas/establecidas Netbeans por lo que tendrán que seguir los siguientes pasos.

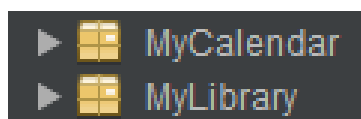
- Este es el mensaje de error, clic en Resolve Problems o Resolver Problemas



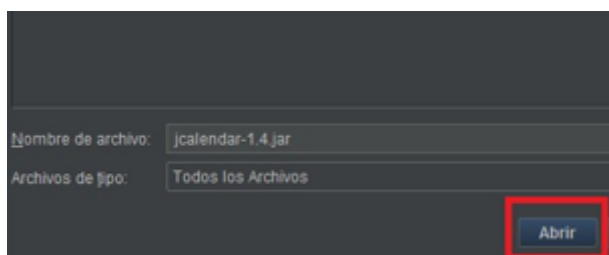
- La segunda ventana nos indicara los errores concretos que tenemos, bien uno de ellos tiene que ver con el tipo de librería usada para generar los calendarios y el otro con la conexión.

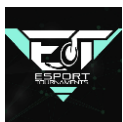


- Vamos en orden, seleccionamos el primero y clicamos en RESOLVE, nos abrirá una ventana en la que tendremos que seleccionar la ruta en la que se encuentran las librerías que se requieren (estas están dentro del mismo proyecto, no hay problema), concretamente se encuentran en estas dos carpetas de la imagen:



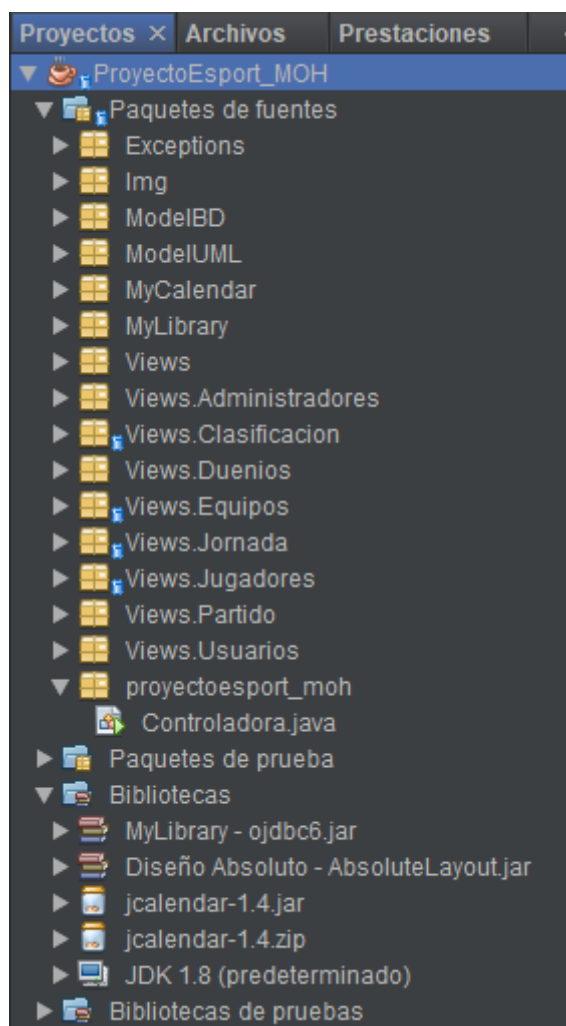
- Una vez ubicada la carpeta seleccionan la librería y hagan clic en Abrir. Este paso obviamente lo tendremos que hacer dos veces, una por cada error de librería.



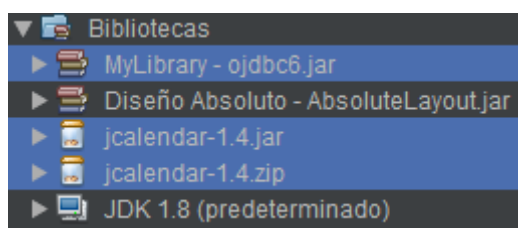


GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

Estos pasos nos han permitido poder abrir el proyecto, el cual tendrá la siguiente estructura:



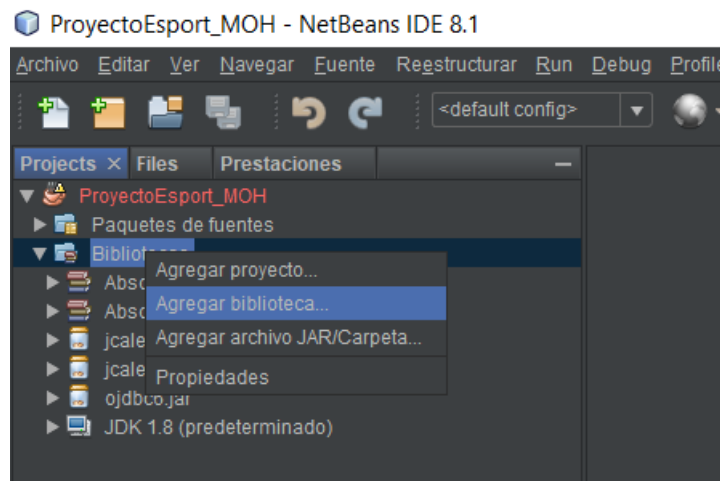
- Y en la parte final, las librerías que hemos agregado.



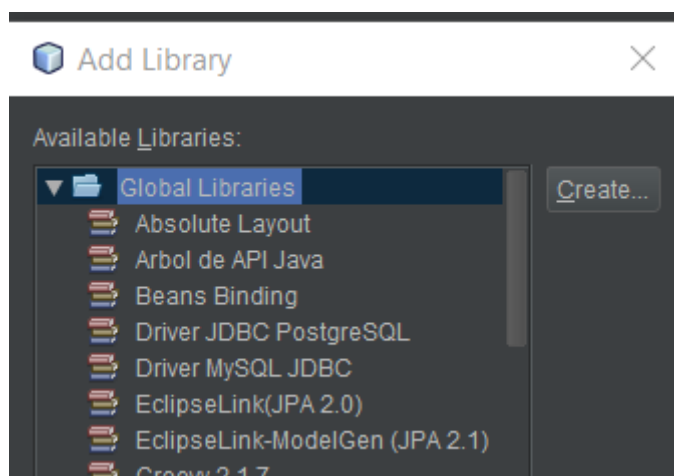


GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

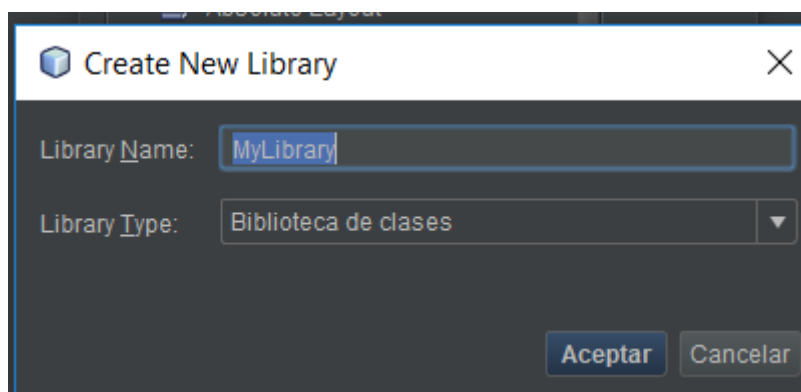
- Si aún le sigue dando un problema con la librería MyLibrary, cárguela manualmente, botón derecho sobre Bibliotecas y clic en Agregar biblioteca..



- Create...

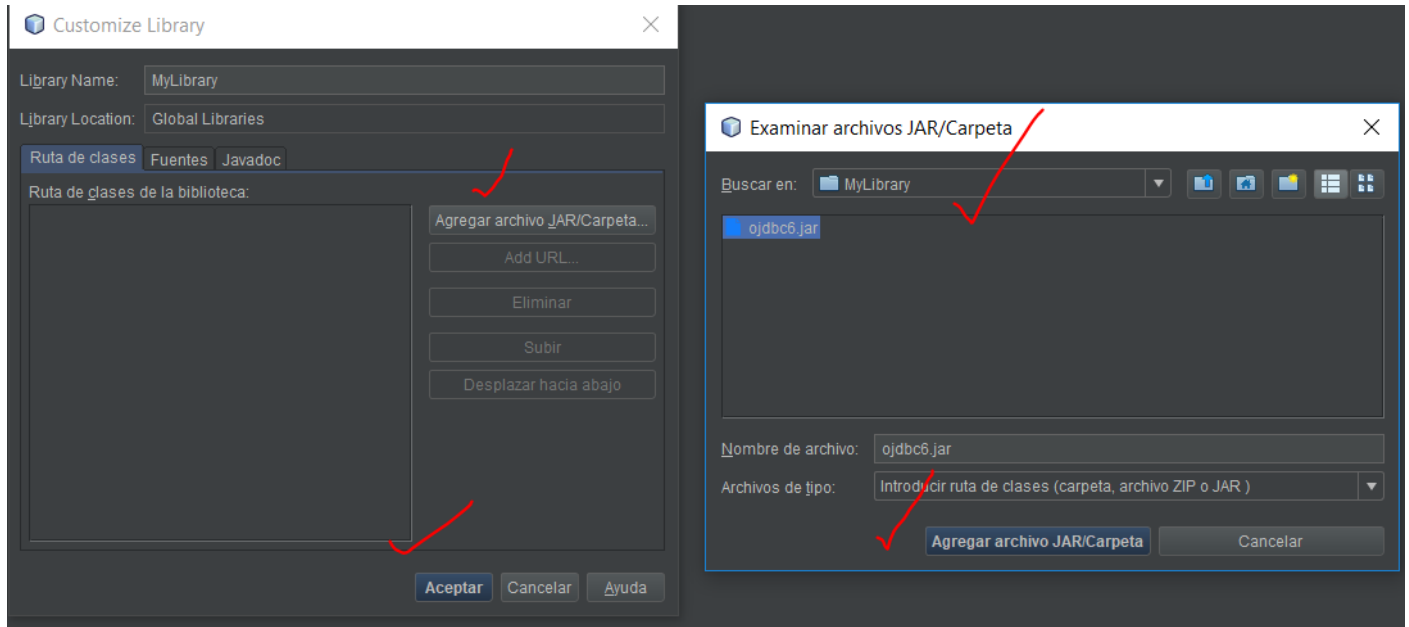


- MyLibrary

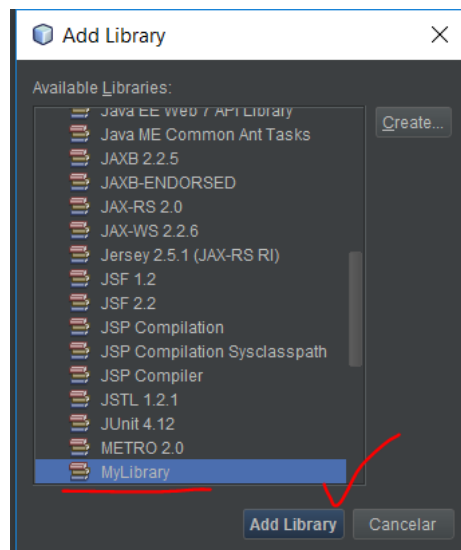


GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

- Luego click en agregar archivo JAR/Carpeta, se abrirá la ventana de la derecha, busquen MyLibrary dentro del proyecto, seleccionen el driver ojdbc6.jar y agréguenlo, luego acepten.



- Finalmente, en la última ventana, seleccionen esa librería y clic en Add



- En el caso de la librería del Calendario, si necesitan hacer un cambio en los calendarios combos de las vistas, deberán de añadir a la paleta dicha selección de calendarios. Sigán estos pasos si lo desean:

[Agregar nuevos componentes a la paleta de NetBeans](#)

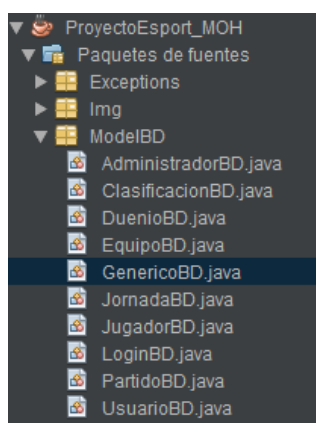


GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

6.3. EDITAR LOS PARAMETROS DE CONEXIÓN

Para que las operaciones realizadas al ejecutar el programa tengan efecto en nuestra Base de Datos o viceversa se requiere una clase que nos permita establecer o cerrar la conexión, para ello vamos a editar dicha clase con el fin de poder trabajar en la Base de Datos generada anteriormente.

- Nos vamos a ModelBD y hacemos doble clic en GenericoBD.



- Es esta clase en donde vamos a poder poner los parámetros de nuestro servidor BD o los datos generados en los puntos 4.1 y 4.2 de la Instalación del Software previa. Ver recuadros marcados en rojo.

```
* Class.forName("oracle.jdbc.OracleDriver"); String user = "system"; String
* pass = "oracle"; String url =
* "jdbc:oracle:thin:@10.10.10.9:1521:db12102";
*
* @param conexion
* @return conexion / null
* @throws SQLException
* @throws ConexionProblemas
*/
public Connection abrirConexion(Connection conexion) throws SQLException, ConexionProblemas {

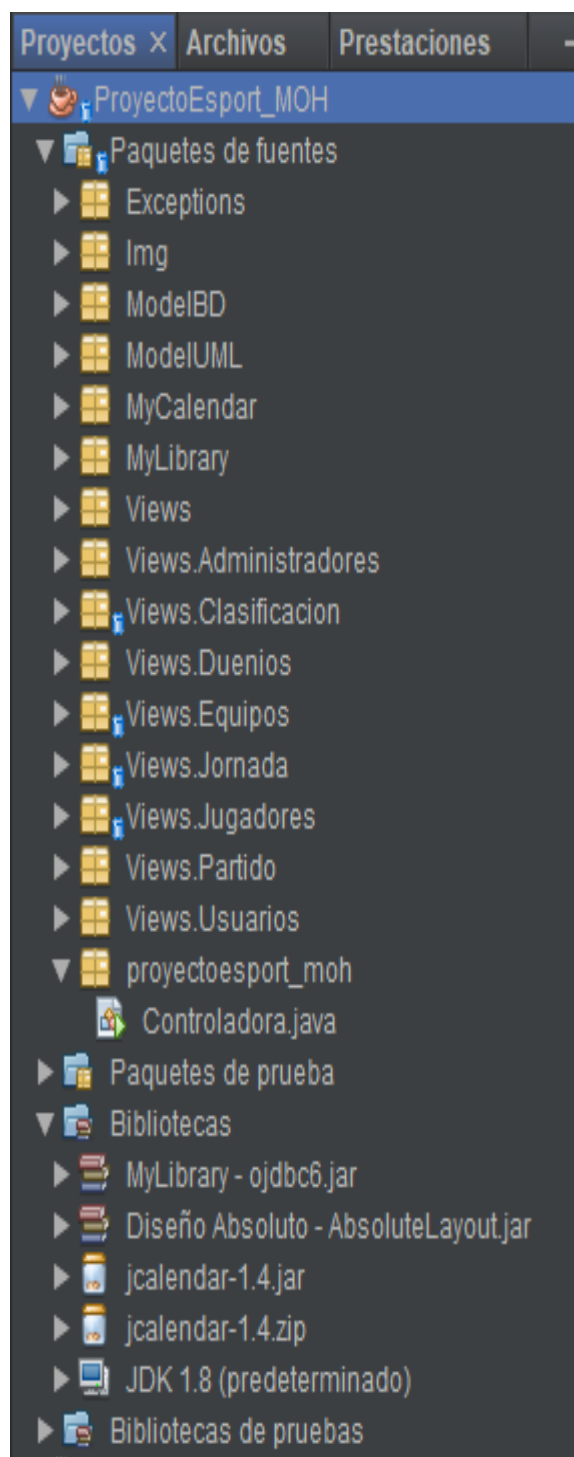
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        conexion = DriverManager.getConnection("jdbc:oracle:thin:@10.10.10.9:1521:db12102" "SCOTT" "miguel");

        if (conexion != null) {
            System.out.println("Conexion Establecida");
            return conexion;
        } else {
            System.out.println("Error Estableciendo Conexion");
            return null;
        }
    } catch (ClassNotFoundException | SQLException E) {
        return null;
    }
}
```

- Recuerdo, tienen que usar sus datos, no los que se muestran, estos son orientativos.

6.4. MOSTRANDO LA ESTRUCTURA INTERNA DEL PROYECTO

La estructura del proyecto en NetBeans no debe mostrarnos ningún elemento de fallo y ha de ser exacta a la de la imagen anteriormente mostrada:



En la carpeta **Exceptions** se recogen todos los posibles errores que pueda generar una función o un usuario, por ejemplo, que falten datos para generar un equipo o si en el caso anterior no establecemos una correcta conexión esta se verá reflejada en pantalla.

Las carpetas **Img**, **MyCalendar**, **MyLibrary** contienen por orden las imágenes usadas en la aplicación, la librería para el calendario y el driver de Oracle que ya mencionados.

En la carpeta **ModelUML** es donde tienen lugar las Clases que se han presentado en el Diagrama de Clases, sus atributos, constructores, métodos getter y setter con los que poder traer o acceder a los atributos.

En la carpeta **ModelBD** se encuentran las mismas clases que en ModelUML pero que nos permitirán realizar todas las llamadas programadas a nuestra Base de Datos para realizar las operaciones CRUD, rellenar combos, etc...

La carpeta **Views** contiene las vistas (ventanas) de Home e Inicio de Sesión, esta carpeta a su vez contiene otras ocho carpetas. Estas carpetas también contienen las ventanas que nosotros como administradores vamos a poder controlar y ver al igual que las propias para los usuarios registrados, así como para los dueños de los equipos que realicen los fichajes o consultas oportunas.

La carpeta **proyectoessport_moh** contiene la clase controladora (el musculo motor) que nos permitirá recoger todas las solicitudes generadas en las ventanas por los usuarios y pasarlas según corresponda a una clase BD para realizar x consulta o que nos abra otra ventana. Es el núcleo inicial de un camino por el cual nos permite acceder a otros. Mientras que lo generado en la Base de Datos es para nosotros (el cerebro) quien guarda la información.

Por ultimo podrá ver las **bibliotecas** con las librerías importadas con antelación.

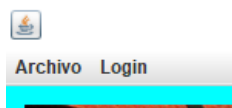


7. PROGRAMA EN EJECUCIÓN

7.1. GESTIÓN DE ACCESO AL PRGRAMA

El programa en ejecución muestra una Interfaz sencilla, amigable y de fácil comprensión. De primeras nos mostrará un Home en segundo plano y la ventana de **Login** en primer plano.

El Home cuenta con una barra superior lo suficiente legible para entender que hace, Archivo / Cerrar la Sesión y Login / Iniciar la Sesión, ojo, si la inician esta cerrara todas las ventanas que tenga abiertas y tendrá que volver a logearse con su usuario y contraseña, por lo que pueden perder todo lo que estén haciendo en esos e momento.



- El **usuario y clave** por defecto del **Administrador** es **ROOT / ROOT**.
- El **usuario y clave** por defecto del **Dueño** es **SUPER / DUEÑO**.
- El **usuario y clave** por defecto del **Usuario** es **SUPER / USUARIO**.

Estos usuarios son los que nosotros como administradores deberemos de usar para poder gestionar de primera mano tanto el correcto funcionamiento del programa y comprobaciones.



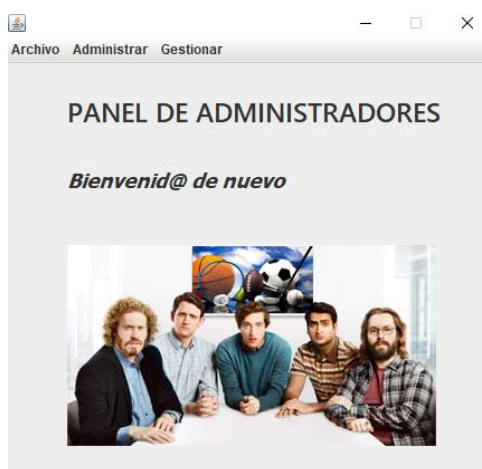
- **NOTA INFORMATIVA**, el aspecto visual del programa puede verse afectado a cambios de última hora que pueden afectar al tipo de fuente y colores, puesto que este manual se ha ido creando acorde a los cambios en el mismo programa, el resto de codificación y funcionamiento no se ve alterado. Disculpen las molestias, gracias.

GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

7.2. IDENTIFICACIÓN SEGÚN EL TIPO DE LOGIN INSERTADO

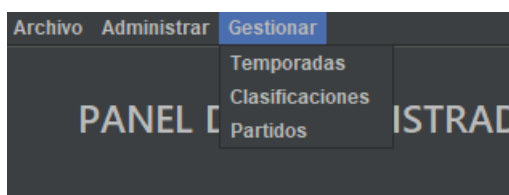
7.2.1. ADMINISTRADORES

Como Administradores de la aplicación vamos a poder tener acceso a lo más importante de la aplicación, gestionar la base en la que se basa el proyecto, todas las operaciones de alta, baja, modificación y consulta (CRUD) de más administradores, para insertar los dueños, usuarios y equipos, en definitiva, controlar el acceso a las personas que van a poder acceder al programa y preparar los equipos que formaran parte del programa afiliado a nuestra suscripción.



Nuestro panel de gestión cuenta con una barra superior para dichas acciones y además como Administradores vamos a ser quienes deberemos de Gestionar las Temporadas (darlas de alta), las Clasificaciones (iniciar una clasificación) e insertar la puntuación de los partidos jugados.

Se dejará explicado en este caso lo que hay que hacer exactamente en cada momento.

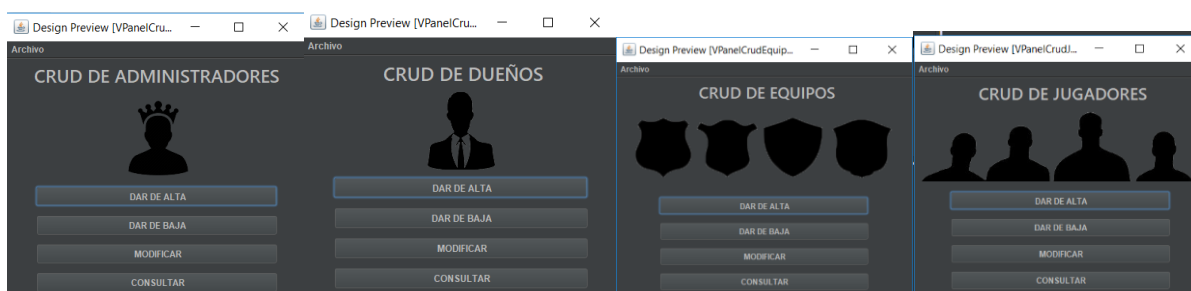


Por lo general el resto de las ventanas no tienen mayor dificultad, aun así, resumidamente se van a explicar los datos quizás más importantes de cada una de ellas.

GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

7.2.1.1. PANELES CRUD – ALTAS – BAJAS – CONSULTAS - MODIFICACIONES

- Los Paneles CRUD de Administradores, Usuarios, Dueños y Equipos, son la antesala a cada una de estas operaciones alta, baja, modificación y consulta, destacar en los Paneles una barra superior que nos permitirá retroceder al Panel inicial de Administradores, la opción de la X para cerrar la ventana se ha controlado para evitar el cierre accidental de la misma y posible pérdida de información, por lo que deberá de retroceder hasta poder cerrar su sesión con seguridad.



- IMPORTANTE:**
 - El orden correcto de creación es el siguiente. ADMINISTRADORES (si se desean más), luego los DUEÑOS, siguiendo de los JUGADORES y por último los EQUIPOS.
- El resto de las ventanas son las que se encargan ya de cada una de las operaciones, las flechas de retroceso nos permiten volver al panel CRUD base y en cada una de las operaciones de baja o edición primero deberá de consultar para estar seguro/a de lo que borra o edita, si no está seguro siempre podrá resetear:



- Recuerden que, de cara al usuario final, hay una serie de validaciones y de igual manera para nosotros, hay unas condiciones de uso en cada una de las ventanas, por ejemplo, los DNI han de ser correctos 8 números y 1 letra, no puede haber campos vacíos, etc.
- Para generar el Login recuerden que hemos cargado un procedimiento que nos lo genera automáticamente.

GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

7.2.1.2. PANELES DE GESTIÓN DE TEMPORADAS

Bien otra de nuestras funciones como administradores son las de poder crear una temporada y esta ser generada de forma automática y aleatoria para que los equipos puedan jugar entre sí. Dicho código lo pueden ver en la Clase Controladora del proyecto, aquí se muestra un pedacito de este:

```
* INICIO SENTENCIAS ALTA / BAJA / CONSULTA / MODIFICACION > JORNADA
*
* ALTA
*
* @param numeroTemporada
* @throws Exception
*/
public static void generarJornadas(String numeroTemporada) throws Exception {
    equipoBD = new EquipoBD();
    jornadaBD = new JornadaBD();

    ArrayList<Equipo> listaEquipos = equipoBD.getAllEquipos();
    System.out.println("Arraylist : " + listaEquipos.size());

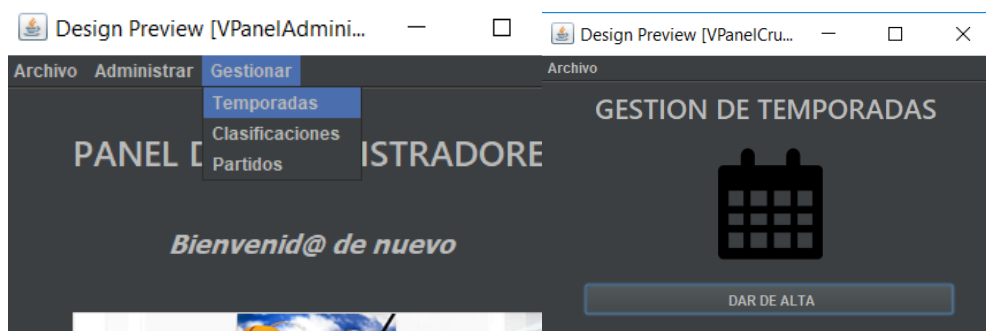
    int[] listacodEquipos = new int[listaEquipos.size()];

    for (int i = 0; i < listacodEquipos.length; i++) {
        listacodEquipos[i] = listaEquipos.get(i).getCodEquipo();
    }

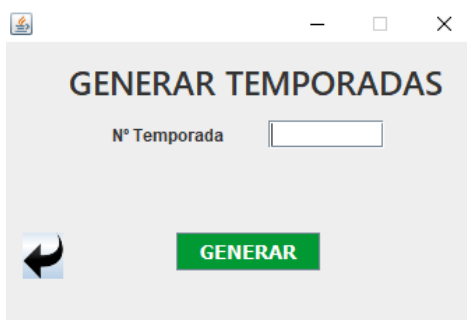
    int N = listacodEquipos.length;
    System.out.println("array: " + listacodEquipos.length);
    String[][] matriz1, matriz2, jornadas, jornadas2;

    matriz1 = new String[N - 1][N / 2];
    matriz2 = new String[N - 1][N / 2];
    jornadas = new String[N - 1][N / 2]; // PRIMERA VUELTA
    jornadas2 = new String[N - 1][N / 2]; // SEGUNDA VUELTA
}
```

- Dicho código no solo nos va a permitir hacer una sola temporada si no que vamos a poder generar las que nos dé la gana. Para ello es cuestión de irnos a Gestión / Temporadas.



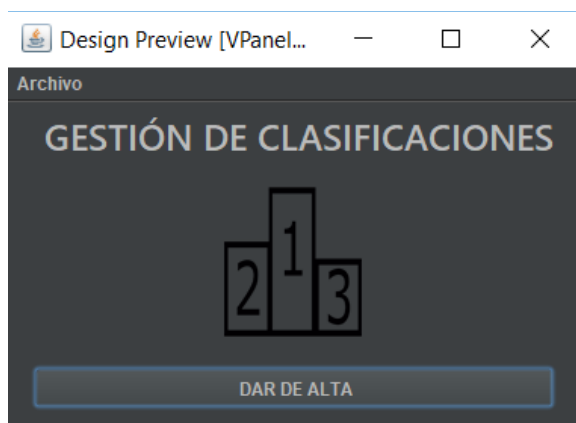
- Nos abrirá el Panel de Temporadas, clicamos en Dar de Alta para luego introducir el numero T01. Al clicar en el botón GENERAL considere tomar 30 segundos para que esta se genere correctamente.



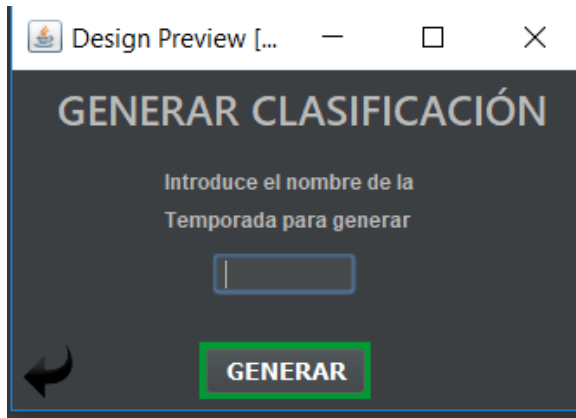
GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

7.2.1.3. PANELES DE GESTIÓN DE CLASIFICACIONES

Repetimos el paso anterior, pero ubicándonos en la pestaña Clasificaciones para que nos muestre la vista, en ella tendremos nuevamente su propio panel:



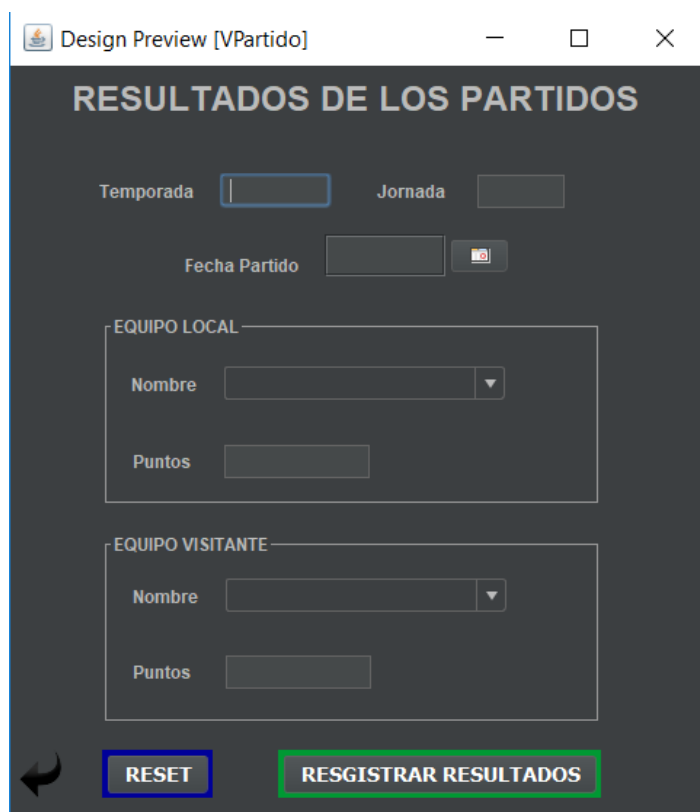
- Al clicar en el alta es cuando podemos inicializar la tabla de las clasificaciones, está por defecto estará sin valor alguno. Al igual que con las temporadas, esta alta requerirá de unos segundos para su implementación.



GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES**7.2.1.4. PANELES DE GESTION PARTIDO / INSERTAR PUNTOS**

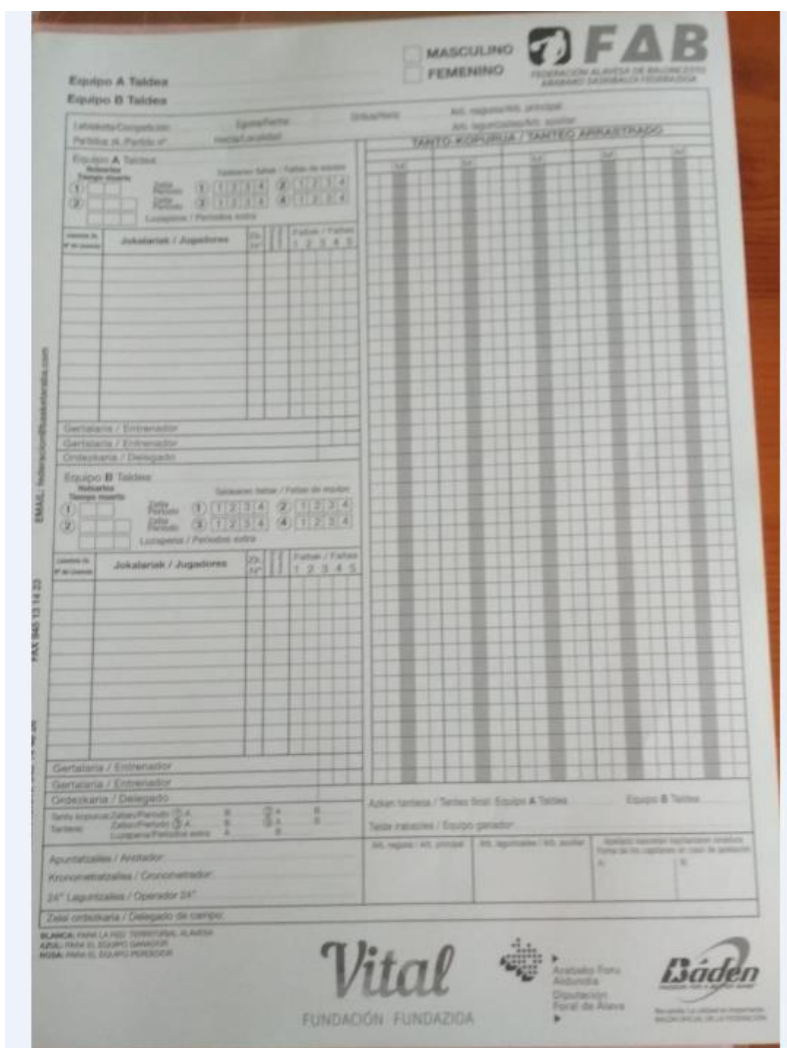
Al clicar en la ventana del Panel de Administradores / Gestión / Partidos nos abrirá una ventana en la que vamos a ir recopilando la información de las plantillas de papel en las que hemos anotado las puntuaciones de los equipos en el partido. En la ventana rellenaremos con esos mismos datos, la temporada, jornada, fecha y obviamente quienes han sido los equipos locales y visitantes.

Esta puntuación mediante la codificación internar nos va a hacer una conversión, el equipo que gana tendrá un 1 y el que pierda un 0 y así sucesivamente, no habrá empate posible. Esta luego mediante el procedimiento *PROCEDURE_SPORT_MOH_3* nos hará que la clasificación anteriormente generada se actualice con esos datos.

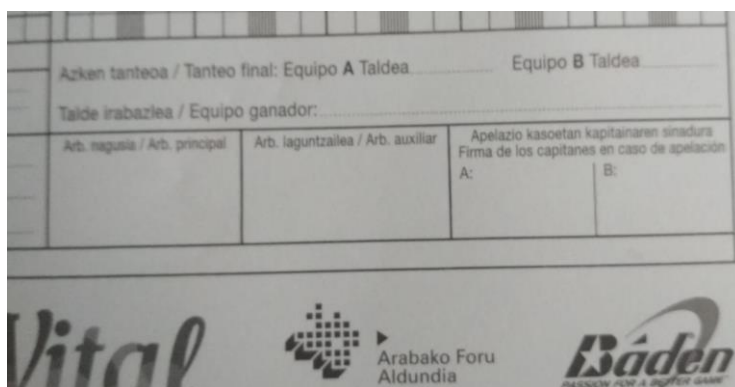


- La plantilla en la que nos basamos para esta codificación es en la oficial de la ACB

GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES



Formulario de Registro de Partido de Baloncesto (FAB) para el Grupo 6. El formulario incluye secciones para el Equipo A y el Equipo B, con campos para el nombre del equipo, el entrenador, el delegado, y los jugadores. También incluye una sección para el árbitro principal y el árbitro auxiliar. El formulario está diseñado para ser completado antes del partido y se utiliza para registrar los resultados y los jugadores que participan.



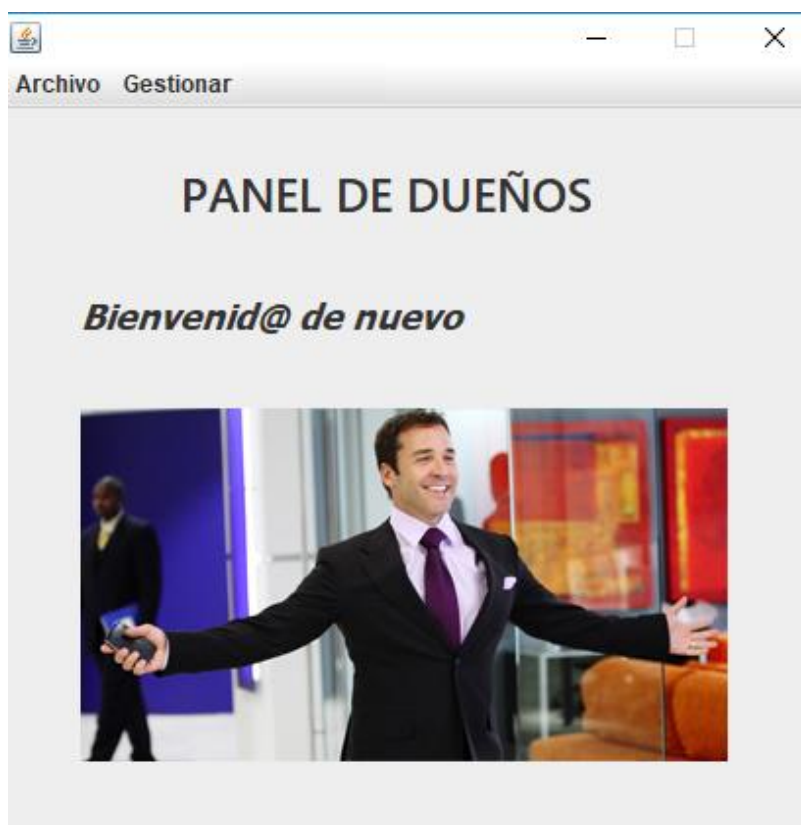
Formulario de Resultados y Firmas. Incluye campos para el resultado final del partido (Equipo A Taldea vs. Equipo B Taldea), el equipo ganador, y las firmas de los capitanes de ambos equipos en caso de apelación. También incluye logos de patrocinadores como Vital, Arabako Foru Aldundia, y Baden.

- Imágenes cedidas por un Árbitro profesional de la Liga Nacional ACB.
- **IMPORTANTE:**
 - El orden correcto de creación es el siguiente. Primero las TEMPORADAS, luego las CLASIFICACIONES y por último se podrán meter los PUNTOS de los PARTIDOS.

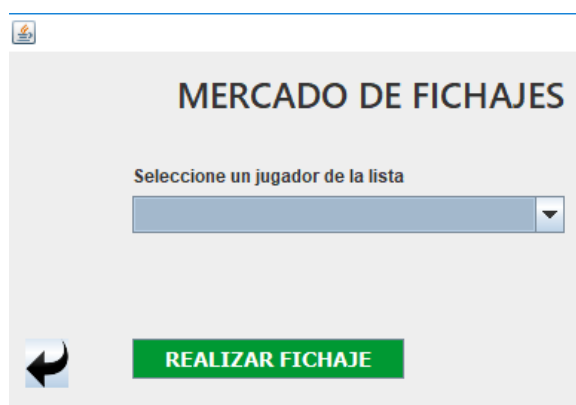
GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

7.2.2. DUEÑOS

Si entramos como usuarios Dueños, nos aparece su propio panel, en este panel el Dueño podrá realizar los fichajes estipulados en el enunciado del Proyecto, máximo 6 jugadores por equipo, obviamente no se incluye como tal el propio Dueño. Para esto tenemos los triggers que nos van a controlar tanto el fichaje máximo como el salario máximo del equipo, 200.000€.



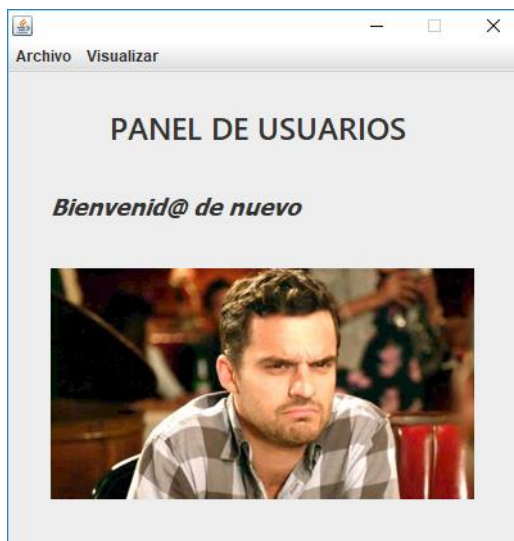
- El Panel de Dueños cuenta al igual que el de Administradores una barra de menú con las funciones de retroceso al Login y la gestión de fichajes.
- Para realizar los fichajes clicamos en el centro del combo y por cada jugador clicaremos en el botón de fichar. El combo ira disminuyendo en base a los jugadores que estén disponibles.



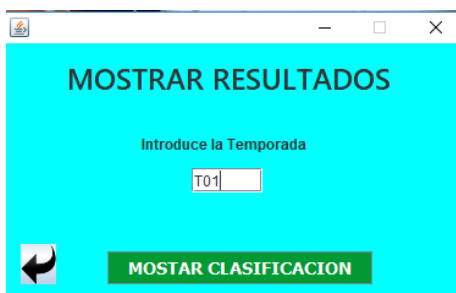
GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

7.2.3. USUARIOS

Los Usuarios obviamente también tienen su Panel, en este podrán ver la clasificación general de los equipos.



- El Panel de Usuarios cuenta al igual que el de Administradores y Dueños una barra de menú con las funciones de retroceso al Login y la gestión de fichajes.
- También cuenta con una pestaña de Visualizar que le mostrara la clasificación mencionada.



- Esto le generará una tabla con la clasificación de la temporada.

EQUIPOS	PUNTOS
NOM	1
BASKONIA	0
BARCELONA	0
BULLS	0
UNICAJA	0
CELTICS	0
LAKERS	0
MADRID	0



GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

8. BIBLIOGRAFIA

- Gran parte de la información que he recopilado tanto para hacer la codificación en JAVA la he obtenido de las siguientes páginas web.

<https://stackoverflow.com/>

<https://docs.oracle.com/javase/tutorial/uiswing/components/table.html>

<https://stackoverflow.com/questions/26471421/round-robin-algorithm-implementation-java>

https://www.tutorialspoint.com/java/util/calendar_compareto.htm

<https://stackoverflow.com/questions/1737627/java-how-to-represent-graphs>

<https://docs.oracle.com/middleware/11119/jdev/api-reference-esdk/javax/ide/util/Graph.html>

<https://stackoverflow.com/questions/7355025/create-pdf-with-java>

- La otra parte necesaria para realizar el proyecto (Bases de Datos) la he obtenido de las siguientes:

<http://stevenfeuersteinonplsql.blogspot.com.es/2016/12/get-rid-of-mutating-table-trigger.html>

<https://www.youtube.com/watch?v=bJy155CAr6c>

http://www.dba-oracle.com/t_avoiding_mutating_table_error.htm

- Sin descontar por supuesto la información aportada por el profesorado durante todo el curso, manuales, etc. Y de igual forma la consulta y ayuda de varios compañeros



GRUPO 6
CONSULTORIA E-SPORT
MANUAL PARA ADMINISTRADORES

9. EPÍLOGO

7.3. EN EL TINTERO

Como es normal ante un proyecto de tal magnitud para una sola persona, pues hay cosas que siempre se le escapan a uno de su capacidad física y mental, el cansancio durante todo el curso y este último sprint pesa por lo que me hubiera gustado hacer...

1. Me hubiera gustado poder realizar la parte en la que un Dueño podría ver las ultimas jornadas y clasificaciones, ya estaban codificadas para ser cargadas, pero a falta de pulir las consultar las he tenido que descartar y centrarme en el plato fuerte del Dueño que es la posibilidad de realizar los fichajes.
2. Otra de las cosas que no he podido terminar ha sido como he comentado las consultas de las clasificaciones, por lo que en este caso el Usuario también se ve afectado en este caso. Este es capaz de poder ver la clasificación final de los equipos en base a una temporada completa pero no me ha dado tiempo a codificar el consultar por jornada y varias partes más que se pedían.
3. El código en JAVA, estoy muy contento con lo que he realizado, pero quizás si debiera de haberlo refactorizado, reutilizado código me ha sido posible pero no en todas las ocasiones. Con lo que si estoy muy contento es con el método que he seguido, el denominado MODELO – VISTA – CONTROLADOR.
4. Otro aspecto indiscutible de lo que me hubiera gustado hacer es un entorno grafico más elaborado, espero que de verdad como se indicaba en el enunciado no afecte mucho al resultado, puesto que mas sencillo y amigable de lo que lo he pensado no puede ser. Pero se que siempre pueden mejorarse cosas y si es algo que me hubiera gustado hacer.
5. No he podido crear un gráfico que ya tenía más o menos preparado para mostrar los resultados como “se pedía” pero lo descarte por falta de tiempo, aunque luego he conseguido que los resultados de la clasificación me los muestre en una ventana con una tabla de forma ordenada.
6. Volviendo al tema “visual o funcional” también tenia en mente hacer una barra de progreso para algunas de las funciones como vienen siendo la creación de la temporada / jornadas y la clasificación, por lo menos mostrar un mensaje con un espere o un contador de cuenta atrás que también tenía localizado para implementar y el generar un PDF con esos resultados.
7. Se que puede haber cosas que quizás en JAVA no sean necesarias o que se deberían de hacer de alguna otra forma, pero estando solo tenia que hacer lo que sabia y como me lo había pensado y quizás algo sea bastante rudimentario.
8. No voy a negar que otra de mis prioridades era desquitarme del año pasado con la creación de combos mediante la llamada de Procedimientos, hacer Triggers y creo que lo he “logrado”. Por lo menos la llamada a los Procedimientos si me ha funcionado.
9. Siempre me hubiera gustado documentar mejor tanto el proyecto como realizar los manuales que se pedían, pero bueno estoy contento con el resultado de este. Recuerdo que si no se ve correctamente es necesario usar MS Office o un PDF que permita un índice con links que permita moverse por este manual.
10. Mas que algo en el tintero es una espinita, la de poder hacer alguna otra cosilla que quizás ahora no recuerdo pero...

7.4. CONCLUSIÓN

No voy a negar que ha sido un proyecto complicado para solo una persona y que encima no haya podido estar muchos días al 100%, pero esto no lo voy a tomar como excusa alguna puesto que estoy muy orgulloso de lo entregado hasta minutos antes de la hora límite.

Como dato a destacar ha sido el planteamiento de este por parte del profesorado ha sido excelente, aunque si es cierto que hay cosas, que quizás dada mi situación pues tenía asinaturas que no me inflúan para el proyecto, al final si me he visto obligado a hacer.