



Proyecto de Laboratorio: scoutingFootballApp



Antonio Patón Rico

Gestión de Sistemas de Información

José Luis Cabezas Velasco

ÍNDICE

1. Proyecto de laboratorio	1
1.1. Descripción general del proyecto de laboratorio	1
1.2. Clases del proyecto	4
2. LabBook.....	7
3. Manual de Usuario.....	9
4. Anexo: Diseño de la Interfaz con Balsamiq (documento DiseñoInterfaz.pdf)	15

1. Proyecto de laboratorio

1.1. Descripción general del proyecto de laboratorio

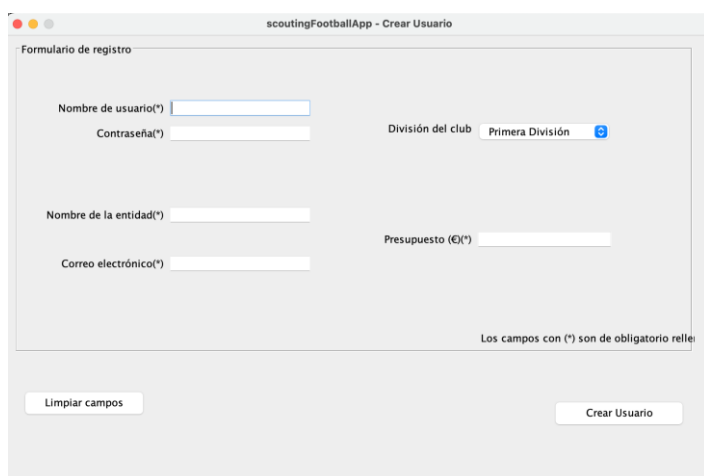
La aplicación creada llamada ScoutingFootballApp mide el rendimiento de los jugadores de fútbol y aconseja a los equipos sobre realizar el fichaje de cierto jugador o si es mejor valorar otras opciones. A continuación, veremos todos los detalles de la aplicación.

Al iniciarla lo primero que encontraremos es una pantalla de login.



The login screen of the scoutingFutbolApp is displayed in a window titled "scoutingFutbolApp - Login". It features the app's logo, which consists of a line graph with a soccer ball at the end of the line. Below the logo, there are two input fields: "Nombre de usuario" and "Contraseña". Underneath these fields are two buttons: "Crear Cuenta" and "Entrar". At the bottom of the form, there is a small text prompt: "Introduce tus datos de autenticación". In the bottom left corner, the text "No logeado" is visible.

Desde esta pantalla tenemos dos opciones. La primera es si no tenemos cuenta, crearla.



The user creation form is shown in a window titled "scoutingFutbolApp - Crear Usuario". The form is titled "Formulario de registro" and contains several input fields: "Nombre de usuario(*)", "Contraseña(*)", "División del club" (a dropdown menu currently showing "Primera División"), "Nombre de la entidad(*)", "Presupuesto (€)(*)", and "Correo electrónico(*)". A note at the bottom right of the form states: "Los campos con (*) son de obligatorio rellenar". At the bottom of the form, there are two buttons: "Limpiar campos" and "Crear Usuario".

Y la segunda opción es introducir nuestros datos para logearnos y poder entrar en la aplicación. Una vez dentro nos encontraremos el menú principal.



Desde aquí se podrá acceder directamente a todas las funciones de la aplicación. También se han incluido dos formas de hacerlo. Una más clásica desde la barra de herramientas superior y otra más visual desde la propia pantalla. Además en cualquier momento se podrá acceder al menú principal desde la pestaña aplicación si así se desea. En el menú principal también podremos editar nuestro perfil o salir de la aplicación




Detallando ahora las funciones a las que se puede acceder desde la barra de herramientas y de izquierda a derecha nos encontramos:

Pestaña Aplicación: Desde aquí se puede volver al menú principal y salir de la aplicación.

Pestaña Futbolistas: Desde aquí se puede acceder a la base de datos de los futbolistas.

scoutingFootballApp - Base de datos de jugadores

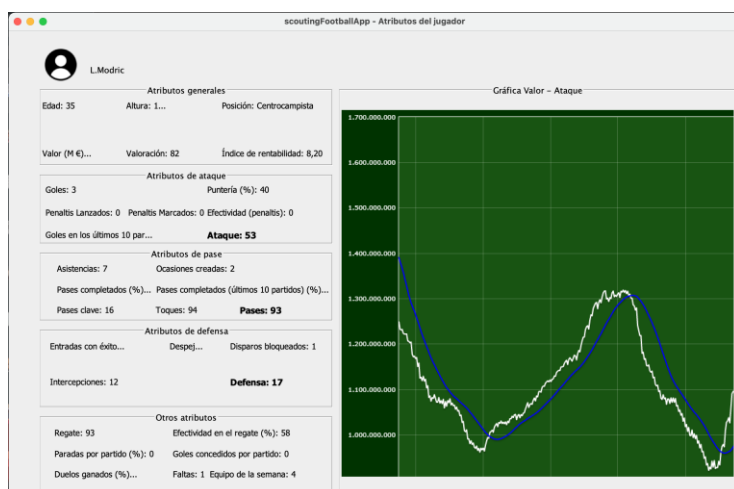
Aplicación Futbolistas Tu equipo Chillos

Nombre Posición Precio (M) 

Nombre	Posición	Disparo	Pase	Regate	Defensa	Goles encajados	Velocidad	Valor
K.Mbappe	Delantero	87	73	90	10	0	94	160
E.Haaland	Delantero	80	51	85	10	0	89	110
L.Messi	Delantero	90	87	95	10	0	89	97
Marquinhos	Defensa	0	42	0	0	0	70	68
R.Varane	Defensa	0	25	0	0	0	80	63
D.Upamecano	Defensa	0	23	0	0	0	79	55
L.Modric	Centrocampista	0	93	0	0	0	70	10

Mostrar datos


Esta herramienta permite visualizar todos los jugadores con sus estadísticas principales. Se pueden realizar búsquedas detalladas con los filtros según nombre, posición y precio. Además, si se pincha sobre uno de los jugadores y pulsamos sobre el botón “Mostrar datos” saldrán más detalles del mismo.



Pestaña Tu equipo: Desde aquí se podrá administrar tu propio equipo.

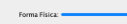
scoutingFootballApp - Administrador del equipo

Aplicación Futbolistas Tu equipo Chillos

Plantilla Actual de Jugadores del Equipo 

Información del jugador

Nombre: V.Dolz Posición: Portero Edad: 25

Forma Física:  Estado: Muy contento Lesiones: 2

Ataque: 15 Defensa: 78 Pase: 70

Paradas/partido: 6 Goles concedidos/partido: 4

Muy buen rendimiento

Comentarios:

Añadir jugador UD Alzira

Jugador seleccionado: V.Dolz

Como se ve, al pinchar sobre un jugador también se pueden ver los detalles de este. También es posible agregar jugadores por si ha habido algún fichaje nuevo en el equipo.

scoutingFootballApp - Añadir nuevo jugador

Formulario para añadir nuevo jugador

Nombre: Posición: Edad:

Forma Física: Estado: Lesiones:

Atributos de ataque

Goles: Puntería (...): Goles en los últimos 10 ...:

Penaltis (...): Penaltis ...: Efectividad Penaltis (%):

Atributos de pase

Asistencias: Toques: Ocasiones creadas:

Pases Clave: Pases Completad...: Pases completados (ult 10 ...):

Atributos de defensa y portería

Entradas con éxito: Despejes: Disparos bloqueados:

Intercepciones: Paradas por partido: Goles Concedidos:

Ataque: Pase: Defensa:

Comentarios:

Calcular campos

Los campos Ataque, Defensa, Pase y Efectividad Penalti se calculan de forma automática (al pulsar el botón).

Limpiar Campos Añadir Jugador

Pestaña Chollos: En esta ventana se pueden ver los chollos de mercado con respecto al índice de rentabilidad que se ha creado. También se puede filtrar según ese índice para buscar los jugadores que pueden resultar más interesantes de fichar por el equipo. Por si desde esta ventana se ve un jugador interesante, también se pueden mostrar sus datos detallados sin tener que cambiarla.

scoutingFootballApp - Chollos

Aplicación

Futbolistas

Tu equipo

Chollos

Nombre de la entidad: UD Alzira

Presupuesto del club: 200 M

Índice de rentabilidad mínimo:

Actualizar

Nombre	Posición	Ataque	Pase	Regate	Defensa	Valor	Valoracion	Índice de rentabilidad
K. Mbappe	Delantero	92	73	93	4	160	91 0.57	
E. Haaland	Delantero	88	51	70	13	110	82 0.75	
L. Messi	Delantero	92	87	95	9	97	93 0.96	
Marquinhos	Defensa	41	42	54	72	68	110 0.16	
R. Varane	Defensa	22	25	50	90	63	70 0.11	
D. Upamecano	Defensa	25	23	49	81	55	70 0.13	
L. Modric	Centrocampista	53	93	93	17	10	82 8.20	

Mostrar datos

1.2. Clases del proyecto

El proyecto de laboratorio realizado (*scoutingFootballApp*) está distribuido en 3 paquetes (siguiendo los estándares ISO recomendados): Dominio, Presentación y Persistencia.

- **Dominio:** En este paquete se encuentran las clases necesarias para poder crear objetos en la parte de presentación, nos encontramos 3 clases (*GestorUsuario.java*, *Usuario.java* y *Jugador.java*):

- **Usuario.java:** Clase en la que se declaran los atributos necesarios para que el cliente de la aplicación pueda tanto autenticarse como crear un usuario nuevo. Consta de los métodos necesarios para poder realizar el proceso de autenticación, creación y actualización de valores de un usuario determinado. Estos métodos lo que hacen es interactuar con el ConectorBD, introduciendo la sentencia SQL necesaria en cada funcionalidad para poder realizar acciones con las bases de datos.
 - La clase está formada por los siguientes atributos: *mLogin*, *mPassword*, *mNombre_Entidad*, *mCorreo*, *mDivision* y *mPresupuesto*, atributos que tiene cada usuario.
 - En cuanto a los métodos está formada por los siguientes:
 - Usuario(): Constructor de la clase.
 - read(): Método que rescata un usuario de la base de datos.
 - insert(): Método para introducir un nuevo usuario en la base de datos.
 - updateUser(): Método para actualizar una serie de valores de un usuario que el cliente introduce.
 - Métodos get(): Método para devolver un determinado atributo del usuario.
 - Métodos set(): Método que cambia el valor de un atributo determinado.
- GestorUsuario.java: Clase para interactuar entre la clase Usuario y las clases del paquete Presentación.
- Jugador: Clase más importante de este paquete. En esta clase están presentes los atributos necesarios para el manejo de la parte central de la aplicación, ya que constantemente se van a estar manejando objetos de esta clase. Posee los atributos necesarios para cada jugador y los métodos necesarios para poder comunicarse, tanto con el paquete Presentación como con la Persistencia.
 - La clase tiene los siguientes atributos: nombre, asistencias, toques, *pases_clave*, *int ocasiones_creadas*, *pases_completados*, *pases_completados_last*, *pase*, *goles*, *punteria*, *penaltis_lanzados*, *penaltis_marcados*, *efectividad_penaltis*, *goles_last*, *ataque*, *entradas_exito*, *despejes*, *tiros_bloqueados*, *intercepciones*, *defensa_gen*, *valor*, *posicion*, *edad*, *altura*, *regate*, *efectividad_regate*, *paradas_partido*, *goles_concedidos_partido*, *duelos_ganados*, *faltas*, *equipo_semana*, *valoración*, *índice_rendimiento*, *equipo*, *forma_fisica*, *estado*, *lesiones*, *comentarios*. Son clases necesarios para el manejo de las diferentes funcionalidades de la aplicación, aunque los 5 últimos sólo se utilizan en la funcionalidad de la gestión del equipo.
 - Los métodos que se usan son los referentes para insertar un nuevo jugador, a obtener un vector de jugadores según unos valores dados y los métodos get y set para obtener el valor de un atributo de un jugador determinado.

- **Persistencia:** Paquete que se encarga de interactuar directamente con la base de datos.
 - ***ConectorBD:*** Clase que se encarga de realizar la conexión de forma directa con las bases de datos creadas con MySQLWorkbench. Tiene los métodos y atributos necesarios para realizar la conexión y la desconexión de forma correcta y realizar cada una de las operaciones necesarias, introducidas mediante sentencias SQL desde cada una de las clases situadas en el paquete Dominio.
- **Presentación:** Paquete donde se encuentra toda la interfaz de la aplicación (parte que va a manejar el usuario para interactuar con esta) Desde este paquete se llama a cada clase del paquete Dominio según el uso que se le quiera dar para poder dar la funcionalidad requerida en cada una de las secciones de la aplicación. Tiene las siguientes clases:
 - ***VentanaLogin.java:*** Clase que se encarga de cargar la interfaz para que el usuario se pueda autenticar mediante un usuario y una contraseña y acceder a los servicios que proporciona la aplicación. También permite acceder al formulario para crear un nuevo usuario.
 - ***FormularioRegistro.java:*** Clase que permite realizar un insert en la base de datos de usuario mediante los datos introducidos por el usuario.
 - ***Menu.java:*** Se trata de un jDeskPane en el que aparecen las distintas opciones que tiene la aplicación en forma de “Internal Frames” (Chollos, base de datos de jugadores y administración del equipo). Desde este frame se puede acceder a cada una de las funcionalidades mencionadas anteriormente. Además, también se puede acceder al formulario de editar datos de un usuario determinado o volver de nuevo a la pantalla de login.
 - ***IF_Futbolistas.java:*** Se trata de un Internal Frame que tiene la funcionalidad de ver los datos de los jugadores de fútbol que aparecen en la base de datos, consta de una tabla en la que aparecen los jugadores con algunos de sus atributos y un botón de mostrar datos, en el que aparecerán más datos. Para el renderizado de las tablas hemos usado otra clase: *MiTablaJugadores.java*
 - ***IF_Chollos.java:*** Internal Frame que tiene la misma forma que IF_Futbolistas, pero en este caso aparecen filtrados por índice de rentabilidad y ordenados por valor. También tiene el botón mostrar datos. Para el renderizado de la tabla hemos usado otra clase: *MiTablaChollos.java*
 - ***IF_Equip.java:*** En este Internal Frame aparecen cada uno de los jugadores en una lista en la que también se pueden ver los datos. Para el renderizado de las listas hemos usado otra clase: *ListCellRender.java*. Además hay posibilidad de añadir nuevos jugadores.

- **FormularioNuevoJugador.java:** Aparecen campos para introducir nuevos datos (algunos se calculan de forma automática mediante fórmulas introducidas en el código mediante el botón calcular campos), para añadir el jugador a la base de datos se realiza con el botón añadir jugador y con limpiar campos se quedan todos los campos vacíos.
- **FormularioEditarPerfil.java:** Se permite editar algunos campos del usuario. Tiene también 2 botones: borrar campos y confirmar los cambios realizados.
- **VentanaAtributosJugador.java:** Frame que muestra los atributos de un jugador determinado.

2. LabBook

22/02/2021

Quedamos para comenzar la primera iteración en la que crearemos las fórmulas con las que se decidirá qué jugador es más recomendable fichar. También recopilaremos información sobre estadísticas de jugadores en diferentes páginas para añadirlas en nuestra base de datos posteriormente. Además, pensaremos qué herramientas vamos a utilizar para el desarrollo de la aplicación.

01/03/2021

Tras terminar con la primera iteración, tenemos pensadas las fórmulas que emplearemos en el cálculo de estadísticas de los jugadores y hemos decidido utilizar el entorno de desarrollo Eclipse junto con el plugin de WindowBuilder para el desarrollo de la aplicación. Además, usaremos la herramienta de MySQLWorkbench para manejar la base de datos MySQL. Ahora, quedamos para empezar la segunda iteración en la que crearemos el diseño de la aplicación. También pensaremos en el diseño que tendrá nuestra base de datos.

09/03/2021

Hemos terminado el diseño de la aplicación con la herramienta Balsamiq. También hemos creado las tablas que usaremos en nuestro programa. Ahora comenzaremos con la iteración 3 en la que desarrollaremos el login y el menú principal.

16/03/2021

Tenemos el login y el menú principal terminados. En la iteración 4 desarrollaremos la ventana que nos cargará la base de datos con los futbolistas y la que nos permite ver más datos de estos.

30/03/2021

Tenemos completada la iteración 3. En la iteración 4 vamos a crear las ventanas donde administraremos nuestro propio equipo y en la que podremos añadir nuevos jugadores a nuestro equipo.

06/04/2021

Hemos acabado la iteración 4. En la iteración 5 tenemos pensado hacer la ventana de chollos para ver qué jugadores pueden suponer un buen fichaje para nuestro equipo.

13/04/2021

Hemos completado la iteración 5. Ahora queda corregir errores pendientes para pulir el programa y crear la documentación.

24/04/2021

El programa y su documentación han sido finalizados a la espera de realizar la entrega de este.

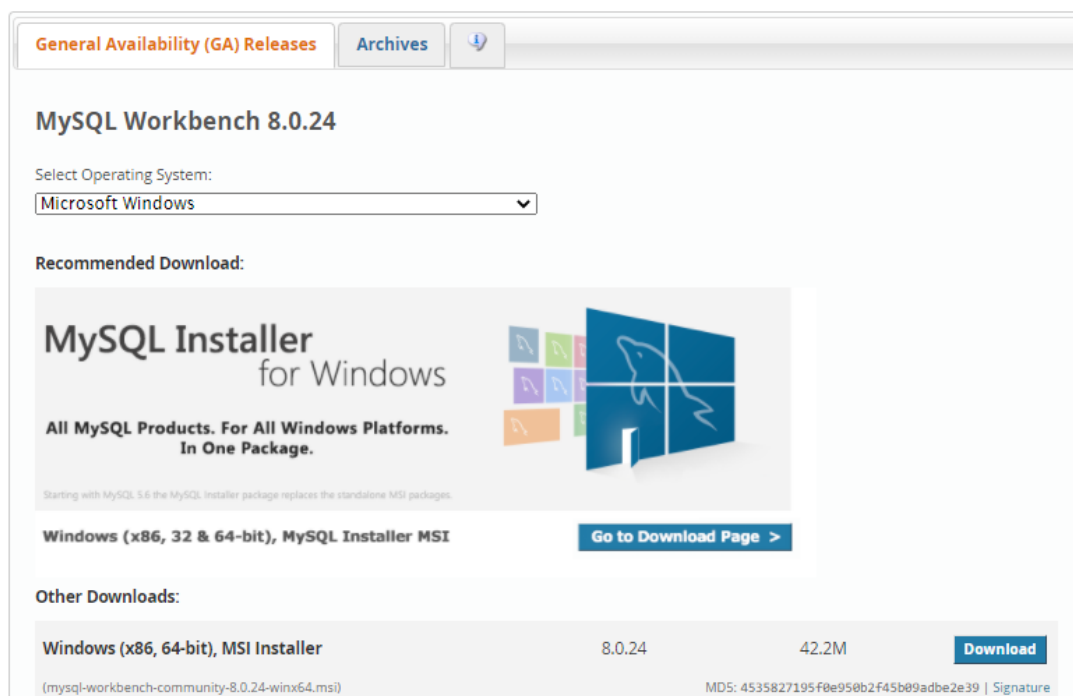
3. Manual de Usuario.

El programa ha sido desarrollado con Java y con MySQLWorkBench y a la hora de la compatibilidad en distintos equipos ha habido algunos problemas, pero mediante el correcto seguimiento de este manual de usuario no tiene que haber ningún problema para su utilización.

En caso de que se quiera mirar el código METER el jar de MySQL en el buildpath de java (está dentro del zip) para que funcione bien.

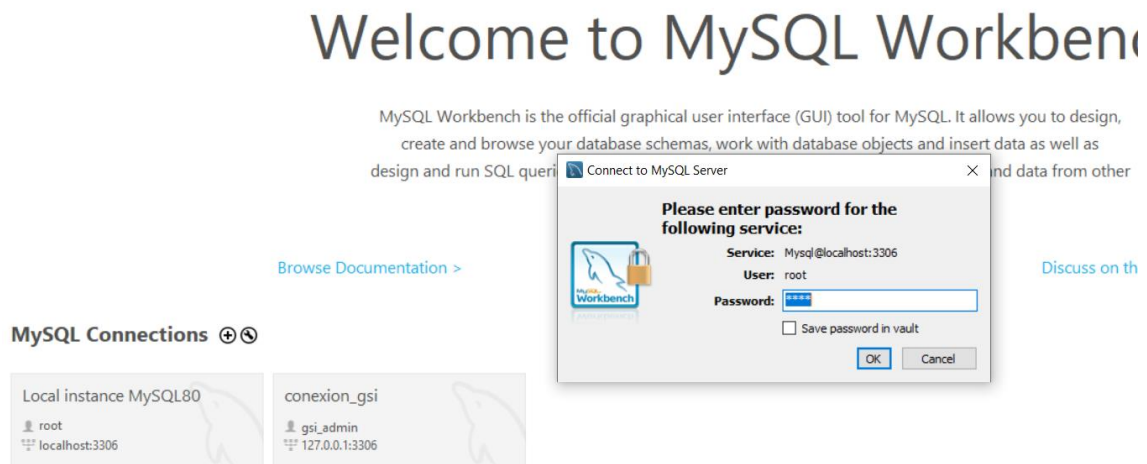
1. Instalar MySQLWorkbench (se trata de un software abierto:
<https://dev.mysql.com/downloads/workbench/>) y crear un usuario.

Al pulsar en el enlace nos aparecerá algo como esto:



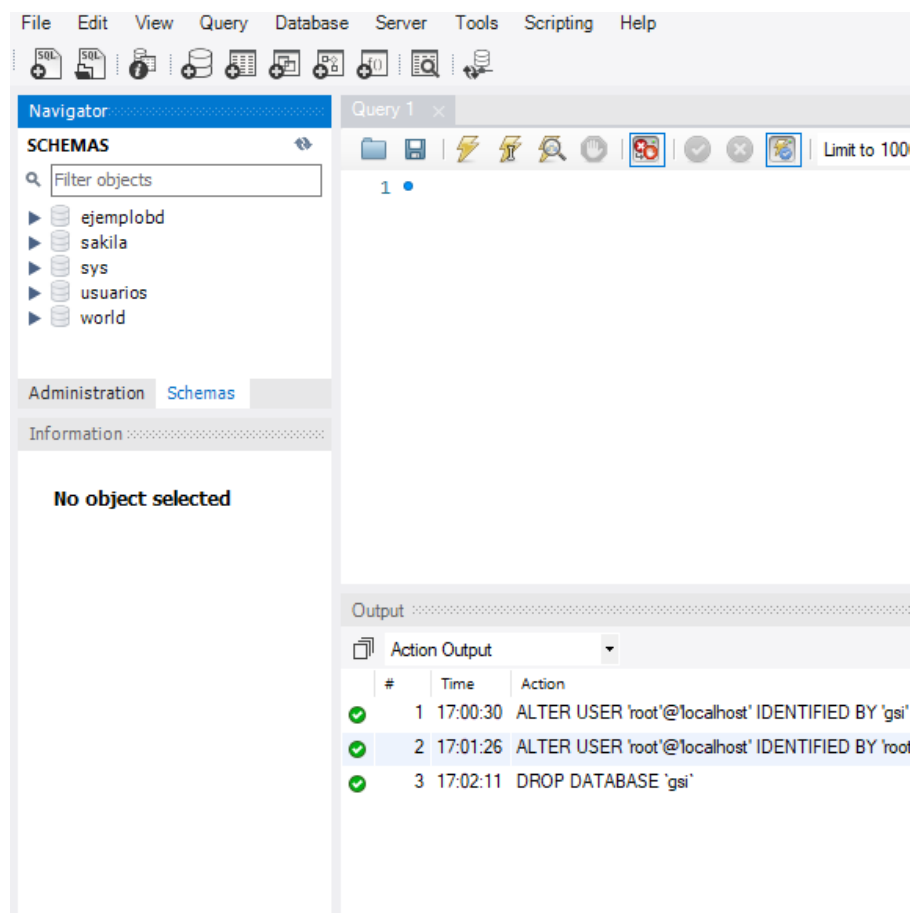
Seleccionar sistema operativo y proceder a la descarga. Cuando se descargue seguir los pasos de instalación y se deberá crear un usuario root, con un usuario y una contraseña (en mi caso he elegido como login root y como password, root).

2. Una vez hecho todo eso tendremos en la pantalla principal de MySQL Workbench una conexión con usuario root y contraseña root:

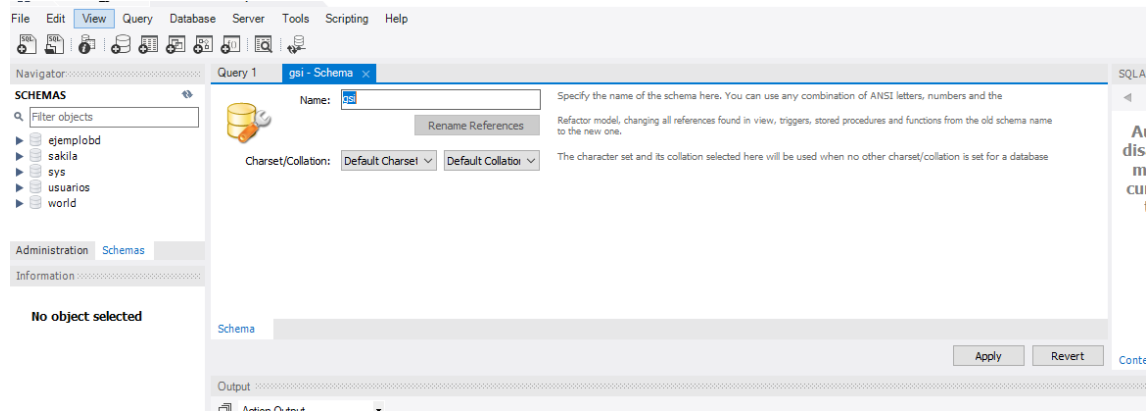


Se trata de la denominada “Local instance MySQL80” a la que podremos acceder con el usuario y contraseña mencionado anteriormente.

3. Al entrar a la conexión nos encontraremos lo siguiente:

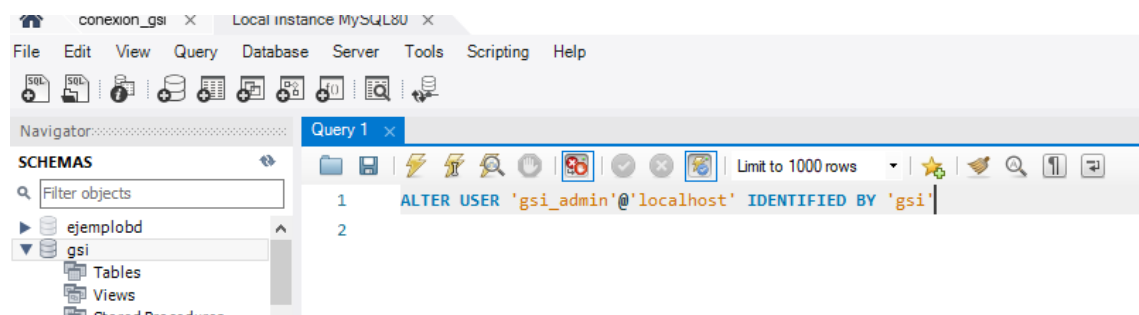


Debemos crear una nueva base de datos denominada *gsi*, esto se hace en el icono de crear base de datos (4º icono debajo del menú, entre View y Query), y le ponemos de nombre '*gsi*':

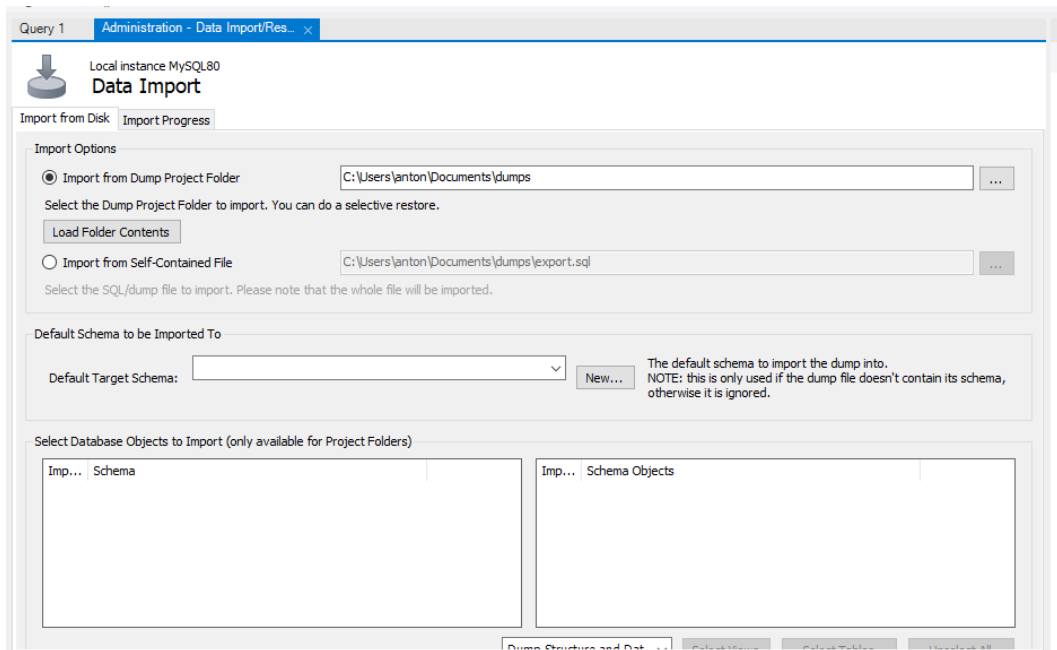


Pulsamos 'Apply' y en la siguiente pestaña también debemos de pulsar 'Apply', que se trataría de la sentencia SQL necesaria.

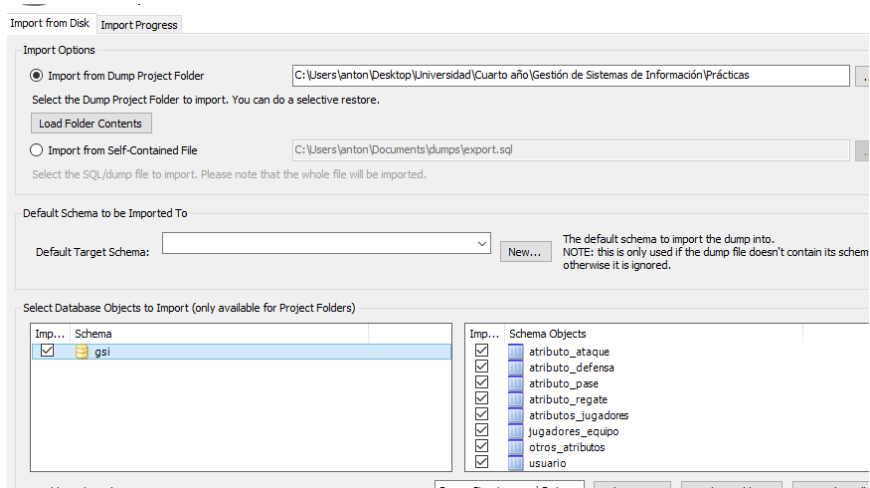
Ahora lo que tenemos que hacer es en un script de MySQL debemos poner el siguiente comando: `ALTER USER 'gsi_admin'@'localhost' IDENTIFIED BY 'gsi'`, en el que '*gsi_admin*' es el usuario y '*gsi*' la contraseña de una nueva conexión:



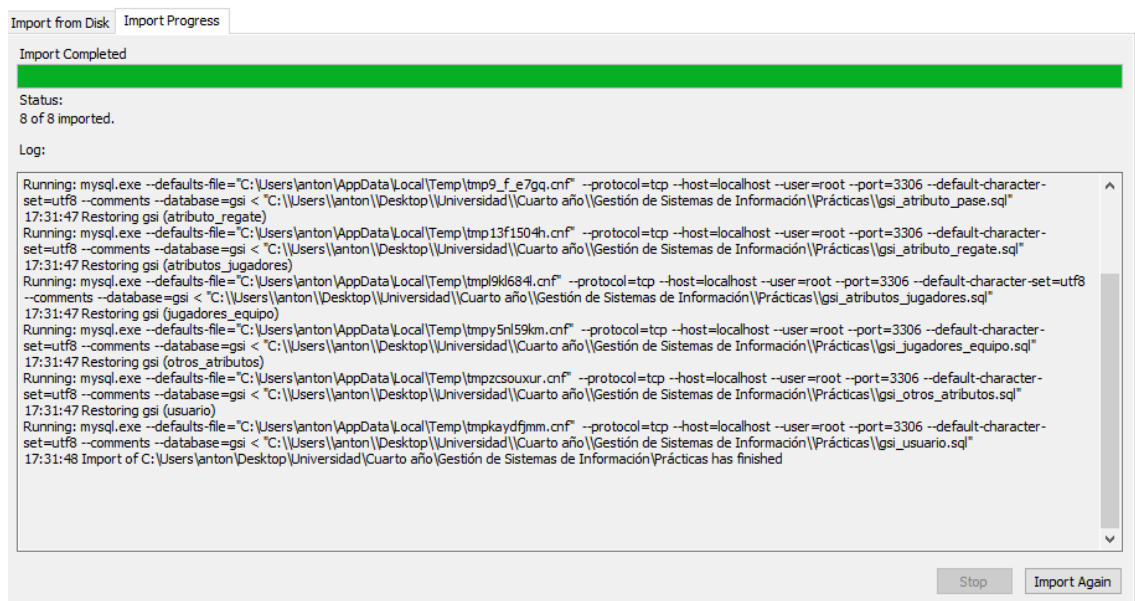
4. Ahora tenemos que exportar las bases de datos que se encuentran dentro del zip, por lo tanto se descomprime el zip, y en Server -> Data Import. Sale la siguiente pantalla:



Seleccionamos la localización de las bases de datos (que se han descomprimido del zip), en mi caso en la carpeta de la asignatura de mi equipo (en los ... se puede seleccionar la carpeta). Al meternos en la carpeta nos aparecerá directamente la base de datos para agregarla. Seleccionamos las tablas y comenzamos la importación:



Import Progress -> Start Import:

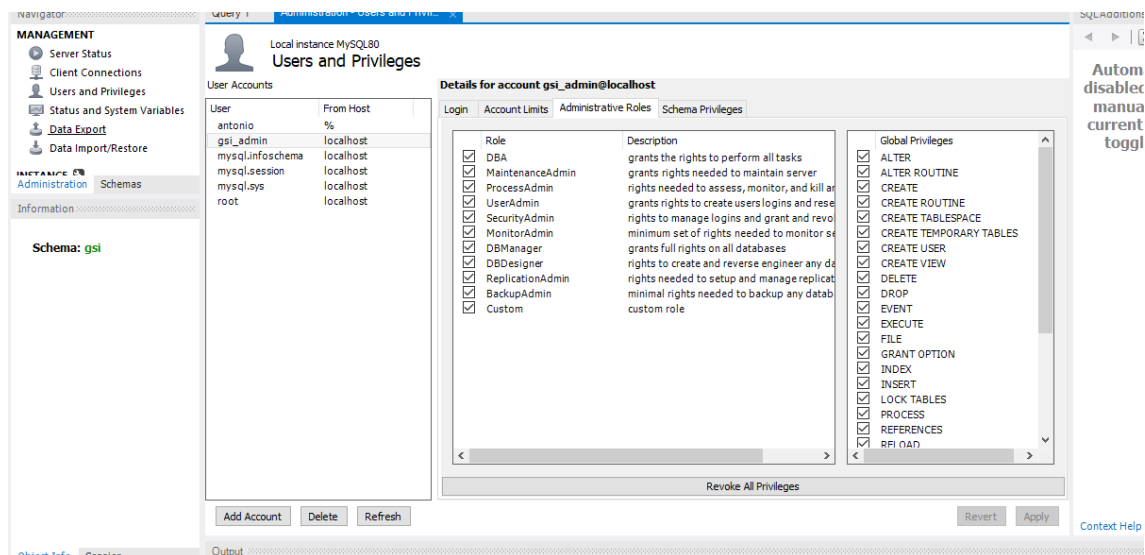


Si no hay ningún error debe salir lo de la ilustración.

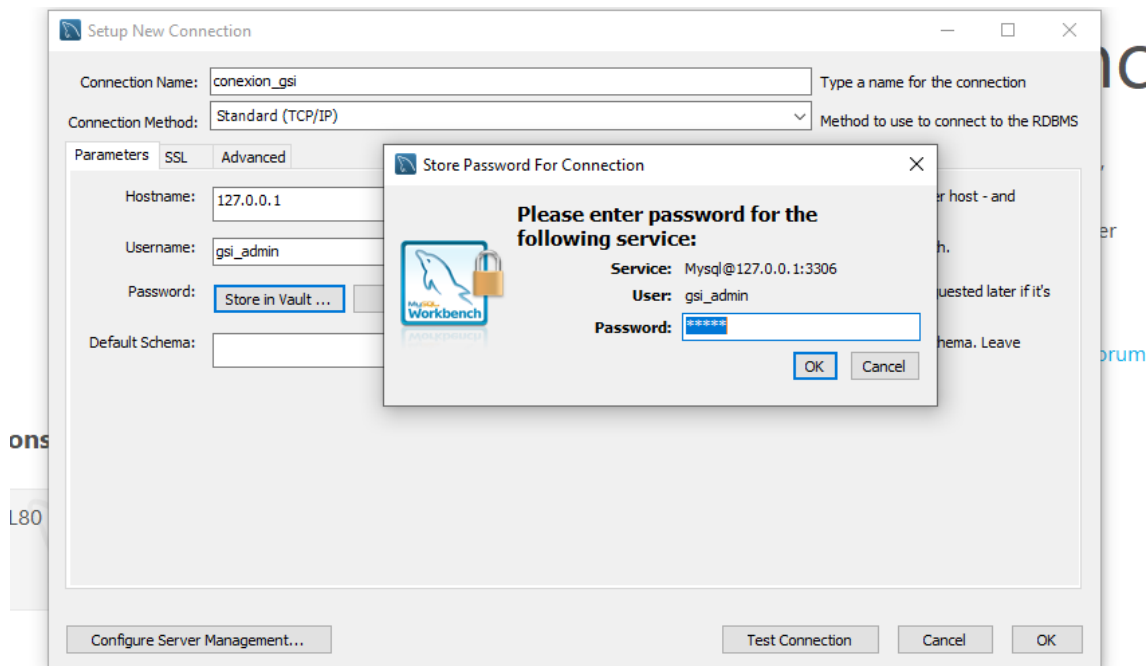
5. Ahora debemos dar permisos al otro usuario para que pueda acceder a la base de datos gsi.

Esto lo hacemos en el apartado Administration -> Users and Privileges -> seleccionamos gsi_admin -> Administrative Roles. Le marcamos todas las opciones y aceptamos mediante 'Apply'.

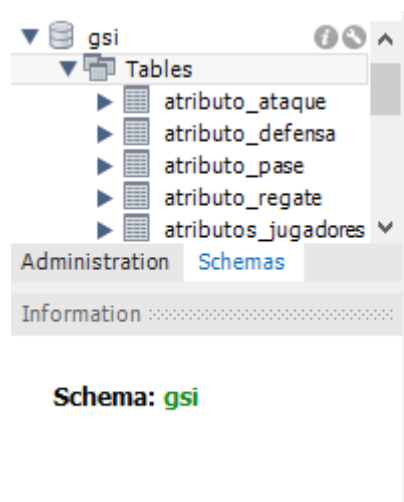
Debe quedar así:



6. Una vez hecho eso creamos una conexión con el usuario 'gsi_admin' y contraseña 'gsi'. La base de datos debe de haberse añadido automáticamente.



Se hace una prueba de conexión, que no debe de dar ningún error y dentro de la conexión debe de aparecer esto:



7. Una vez hecho todo esto podemos ejecutar la aplicación (scoutingFootballApp.jar) con usuario 'udalzira' y contraseña 'udalzira'.

4. Anexo: Diseño de la Interfaz con Balsamiq (documento DiseñoInterfaz.pdf)