

Universidad Politécnica de Valencia

Aplicación software para el diseño y gestión de ejercicios deportivos y jugadas de equipo, aplicado al baloncesto

David García Juan

Profesor Asignado: Oscar Vega Gisbert

Grado en ingeniería informática

Curso académico: 2016 - 2017



UNIVERSIDAD
POLITECNICA
DE VALENCIA

Resumen

El objetivo principal del proyecto es implementar una aplicación que sea de código abierto para el diseño, elaboración y clasificación de ejercicios de carácter deportivo y jugadas de equipo, así como la planificación de entrenamientos. La aplicación permite una fácil catalogación de los ejercicios y generar documentos para su mejor uso. La aplicación está especializada en un deporte, el baloncesto.

Palabras Clave: diseño, planificación, baloncesto, entrenamiento deportivo.

Abstract

The main purpose of the project is implement an open source application to design, elaborate and classification of sport drills and team plays, as well as create and planning of sport trainings. The application let you an easy way of sorting the drills and generate documents to use it. This app is specialized in one sport, in basketball.

Keywords: design, planning, basketball, sport training.

Índice

Índice de contenidos

Índice.....	1
Índice de figuras.....	2
Introducción.....	3
Estado actual de las aplicaciones.....	4
Objetivos.....	6
Descripción del proyecto.....	7
Netbeans.....	8
Java.....	8
JavaFX.....	9
Scene Builder.....	9
CSS.....	10
SQLite.....	10
Itext 5.....	11
Github.....	11
Desarrollo del proyecto.....	12
Instalación y preparación del entorno.....	13
Lógica de datos.....	14
Modelos de vistas.....	17
Desarrollo de los controladores.....	22
Funcionalidades adicionales.....	24
Presupuesto.....	27
Conclusiones.....	28
Mejoras para el futuro.....	28
Bibliografía.....	30
Anexos.....	32
Localización del repositorio en Github.....	32
Manual de usuario.....	32

Índice de figuras

Índice de ilustraciones

Ilustración 1: Entorno de trabajo de la aplicación Scene Builder.....	9
Ilustración 2: Modelo de comunicación del patrón MVP.....	12
Ilustración 3: Estructura de la base de datos.....	14
Ilustración 4: Operación de insertar un registro nuevo.....	15
Ilustración 5: Operación de eliminar un registro.....	15
Ilustración 6: Operación de renombrar una jugada.....	16
Ilustración 7: Operación de actualizar la información.....	16
Ilustración 8: Relación entre las diferentes vistas.....	17
Ilustración 9: Secciones de la vista del Frame.....	18
Ilustración 10: Botones de creación, edición y eliminado de jugadas.....	18
Ilustración 11: Botones de guardar, borrar, deshacer y rehacer.....	19
Ilustración 12: Botones de imprimir ejercicio y plan.....	19
Ilustración 13: Botones de edición de jugadas.....	20
Ilustración 14: Campo de ataque.....	20
Ilustración 15: Campo de defensa.....	20
Ilustración 16: Campo entero.....	20
Ilustración 17: Botones de manejo de las páginas de las jugadas, y texto asociado.....	21
Ilustración 18: Vista del Plan.....	21
Ilustración 19: Vista del PlayEditDialog.....	22
Ilustración 20: Ejemplo de declaración con las etiquetas @FXML.....	23
Ilustración 21: Ejemplo declaración de acción de botón.....	23
Ilustración 22: Ejemplo de ejercicio impreso.....	25
Ilustración 23: Ejemplo de una planificación de entrenamiento.....	25
Ilustración 24: Ventana principal una vez iniciado el programa.....	32
Ilustración 25: Ventana de creación de jugada.....	33
Ilustración 26: Pantalla de creación de una planificación.....	34

Introducción

Hoy en día hay una gran cantidad de programas dedicados al mundo del deporte. Cuantificar todo lo que ocurre dentro del terreno de juego se hace más importante y más difícil, ya que cada vez es más la cantidad de dinero que mueve esta forma de ocio. Por ello tanto equipos como medios de comunicación desglosan todo lo que ha ocurrido en el partido, haciendo análisis más detallados.

Dentro del baloncesto, se usa la tecnología para decidir acciones importantes en los partidos, como es la repetición de jugadas polémicas, determinar el tiempo que queda de posesión o el final de esta, y los tableros de información de estadísticas, como son los puntos o faltas, en los marcadores.

Pero también se emplea tecnología que no se aprecia a simple vista. Es la que se usa en el día a día para el trabajo de un equipo deportivo profesional, que alcanza un punto más de detalle y que puede marcar la forma de juego que se llevará a cabo.

El uso de la tecnología dentro de un equipo, puede separarse en dos grandes apartados:

El análisis: Esta agrupación permite conseguir un mayor detalle de desglose de lo que ha ocurrido en el campo de juego, extrayendo datos que no son apreciables a la vista. Hay desde generadores de estadísticas avanzadas de diferentes aspectos del juego a cuadros de tiro con sus porcentajes de acierto. Programas de vídeo especializados para hacer estudios de rivales o incluso propio, recreación de jugadas mediante sensores, e incluso ya se está empezando a incluir el big data.

La gestión de un equipo: Esta agrupación ayuda a simplificar todas las tareas que envuelven a la vida de un equipo, que pueden ser más o menos relacionadas con el mundo del baloncesto. En este grupo se incluyen programas para la gestión de viajes y alojamientos, gestión de instalaciones deportivas, plataformas oficiales de las distintas federaciones, creación de entrenamientos y su planificación, canales de comunicación con padres y jugadores...

En la gestión de ejercicios y planificación entrenamientos deportivos hay una gran variedad de programas y con soporte para diferentes plataformas. Estos programas ayudan a tener una clasificación clara de los ejercicios, permitiendo tenerlos así agrupados formando una biblioteca y ahorrando tiempo a la hora de diseñar un entrenamiento.

El gran problema de la oferta actual de este tipo de software es que o bien son de pago, como es el ejemplo del software FastDraw de la empresa FastModel, o bien de tipo freemium como el software Jes-soft Basketball Playbook. Por lo tanto, ante la situación actual de este mercado, la intención de este proyecto es dar una alternativa 100% gratuita a estos programas.

La aplicación ha sido diseñada para ser intuitiva, sencilla y sobre todo útil, permitiendo la creación de documentación asociada para poder usarlo fácilmente en los entrenamientos, todo con una visualización correcta.

Estado actual de las aplicaciones

Como se ha mencionado anteriormente, ya existen aplicaciones que realizan esta función, y algunas de ellas con un largo recorrido.

Existen algunas que son más conocidas, como son el caso de Fast draw o Jes-soft Playbook, pero existen más alternativas en este mercado.

El gran problema de muchas de estas aplicaciones es el precio, muchas veces prohibitivo para sus usuarios, siendo usado mucho en un mundo donde es difícil ser profesional con una remuneración digna.

Las aplicaciones de escritorio para la creación de jugadas son las siguientes:

Fast draw: aplicación que pertenece a la empresa Fast Model. Puede funcionar en varios sistemas operativos, incluido IOS. Es una aplicación muy completa y actualmente es la referente dentro de este campo. Su suscripción anual va desde los 25\$ para la versión de IOS sólo, hasta 148\$ en su versión más completa. Es la opción ideal para equipos profesionales o clubes con de gran tamaño.

Jes-soft Playbook: Esta aplicación cuenta con versiones para Windows y macOS para la creación de jugadas y entrenamientos. Además, cuenta con aplicación

para móvil que actúa como visor de los ejercicios creados. Su precio es de 35\$, siendo una opción más accesible que la anterior, e incluye un año de acceso a su base de datos on-line de jugadas y de actualizaciones. Esta aplicación es bastante completa y cumple de sobras su finalidad.

Oneplaybook: aplicación web que permite la gestión de un equipo deportivo. Dentro de sus opciones permite la creación de jugadas y ejercicios, aunque con una interfaz bastante pobre y limitada. Se ha parado su desarrollo y no presenta mejoras desde 2009. Su precio anual es de 48\$

Playmanager: Aplicación de escritorio cuyo desarrollo no ha continuado desde 2012, cuyo última versión es para Windows XP. Es gratuita, pero presenta pocas funcionalidades y cubre sólo lo más básico.

También hay diferentes aplicaciones que realizan esta función y que son para dispositivos móviles, pero la mayoría de estas solo enlazan vídeos o ejercicios prediseñados. En este campo destacan: *Basketball Blueprint*, *Basketball coach's clipboard* y *Coachbase*.

Tras esto, se observa que en la actualidad no hay ninguna opción que ofrezca este servicio de forma totalmente gratuita, algo necesario para los clubs con poco presupuesto y entrenadores noveles o no profesionales.

Objetivos

El objetivo principal de este proyecto, es la creación de una aplicación de escritorio que permita dibujar un ejercicio deportivo, en este caso baloncesto, con la iconografía propia del baloncesto (lenguaje universal aceptada mundialmente), permitiendo asociar texto explicativo al ejercicio dibujado. Todos los ejercicios dibujados son almacenados en el programa, creando una biblioteca personal.

Así mismo, otro objetivo principal es la planificación de los entrenamientos, permitiendo usar los ejercicios anteriormente diseñados y asociando los datos necesarios para organizar de forma óptima la práctica deportiva.

Otros objetivos secundarios son:

- Clasificación de los ejercicios por categorías y subcategorías, permitiendo una rápida búsqueda.

- Creación de documentación en formato pdf, tanto de los ejercicios como del entrenamiento planificado. Al entrenamiento planificado se podrá adjuntar los ejercicios previamente almacenados.

- Disponibilidad de la aplicación en varios idiomas.

- Una interfaz sencilla y clara para los usuarios no expertos.

Descripción del proyecto

Como base para el proyecto, se emplea el IDE Netbeans, creado originalmente por Sun Microsystems, y mantenido por la comunidad.

La arquitectura de la aplicación seguirá el patrón MVP (modelo - vista - presentador). La elección de este patrón se basa en tener separados las diferentes partes del software. La interfaz mediante la que interactúa el usuario es pasiva, enviando las operaciones de interacción al presentador, que es el encargado de cargar los datos necesarios y prepararlos según se necesite para mostrarlos. Por tanto es el presentador en el que recae principalmente el peso de la aplicación.

Esto también facilita la tarea de desarrollo, ya que permite cierta independencia entre las partes, y un mantenimiento de la aplicación más eficiente.

Durante la realización del proyecto, se van a emplear diferentes tecnologías:

Para la lógica del programa se empleará Java, en su versión 8, ya que permite que se pueda utilizar en cualquier sistema operativo, y su gran repertorio de librerías permite un fácil desarrollo de la solución planteada.

Para almacenar los datos generados durante el uso de la aplicación se usará una base de datos SQL, empleando SQLite para este caso, siendo la alternativa que mejor se adapta a la solución elegida.

Para la parte visual de la aplicación se empleará JavaFX, en su versión 8, y CSS, soportado por Netbeans, junto al editor de ventanas Scene Builder de JavaFX.

Para la creación de documentos, se emplea la librería iText, en su versión 5.

Netbeans

Netbeans es un IDE (integrated development environment) de código abierto para desarrollar con Java, PHP, C++ entre otros lenguajes de programación. Es un IDE programado en Java y puede funcionar en cualquier sistema operativo que tenga la maquina virtual de java (JVM).

Este IDE está diseñado para limitar los errores de código y facilitar su corrección analizando lo escrito en el momento. Para ello tiene numerosas herramientas como *FindBugs*, para encontrar errores, el *Debugger*, para ayudar en la monitorización del código, o la herramienta para refactorizar de una forma más eficiente.

Facilita la escritura con su editor de texto, resaltando las estructuras del lenguaje y variables, las plantillas o el autocompletado. Su forma de gestionar los archivos vinculados a un proyecto, hace que este proceso sea sencillo y claro, con una organización clara de carpetas.

Por último, permite instalar módulos complementarios para facilitar la integración con otras herramientas ajenas al IDE. Estos módulos se instalan de forma totalmente dinámica, lo que permite que siempre estén actualizados.

Java

Java es un lenguaje de programación ampliamente conocido y usado por la comunidad, desarrollado originalmente por Sun Microsystems.

Es un lenguaje de alto nivel, orientado a objetos y multiplataforma. También es uno de los lenguajes más usados a nivel mundial. Tiene una gran variedad de librerías, lo que hace que sea muy completo. Este lenguaje es muy versátil, siendo útil tanto para desarrollo web, aplicaciones de escritorio e incluso aplicaciones móviles.

En este proyecto se empleará la versión Java SE 8, que incluye la librería JavaFX completa, una nueva funcionalidad como es el uso de las expresiones lambdas, además de mejoras en seguridad y nuevas APIs.

Desarrollar en este lenguaje es absolutamente gratis, ya que no se requiere ningún tipo de licencia.

JavaFX

JavaFX es una librería GUI para Java, con intención de ser el reemplazo definitivo de Swing. Esta librería permite la creación de interfaces, permitiendo que los usuarios puedan tener una interacción más sencilla con la aplicación. Permite la inclusión de lenguajes de hojas de estilo como CSS para mejorar la apariencia.

La versión 8 de javaFX se incluye ya en el jdk oficial. Esta versión añade nuevos elementos, como el *TreeTableView*, mejoras de elementos anteriores, como el *DatePicker*, soporte para Swing embebido, el soporte de las expresiones lambdas.

Con JavaFX se determina la estructura de toda la apariencia de la aplicación, la distribución de los elementos de la pantalla, las funcionalidades de los botones y la utilización de la sección canvas.

Scene Builder

Scene Builder es una aplicación gráfica adaptada a la librería JavaFX.

Esta aplicación permite una fácil creación del aspecto gráfico de aplicaciones mediante el uso de una interfaz del tipo drag&drop, arrastrando los elementos que se quiere añadir y el uso de etiquetas FXML para asignar las funciones a los elementos. Esta herramienta facilita la edición de la interfaz gráfica, generando automáticamente el código.

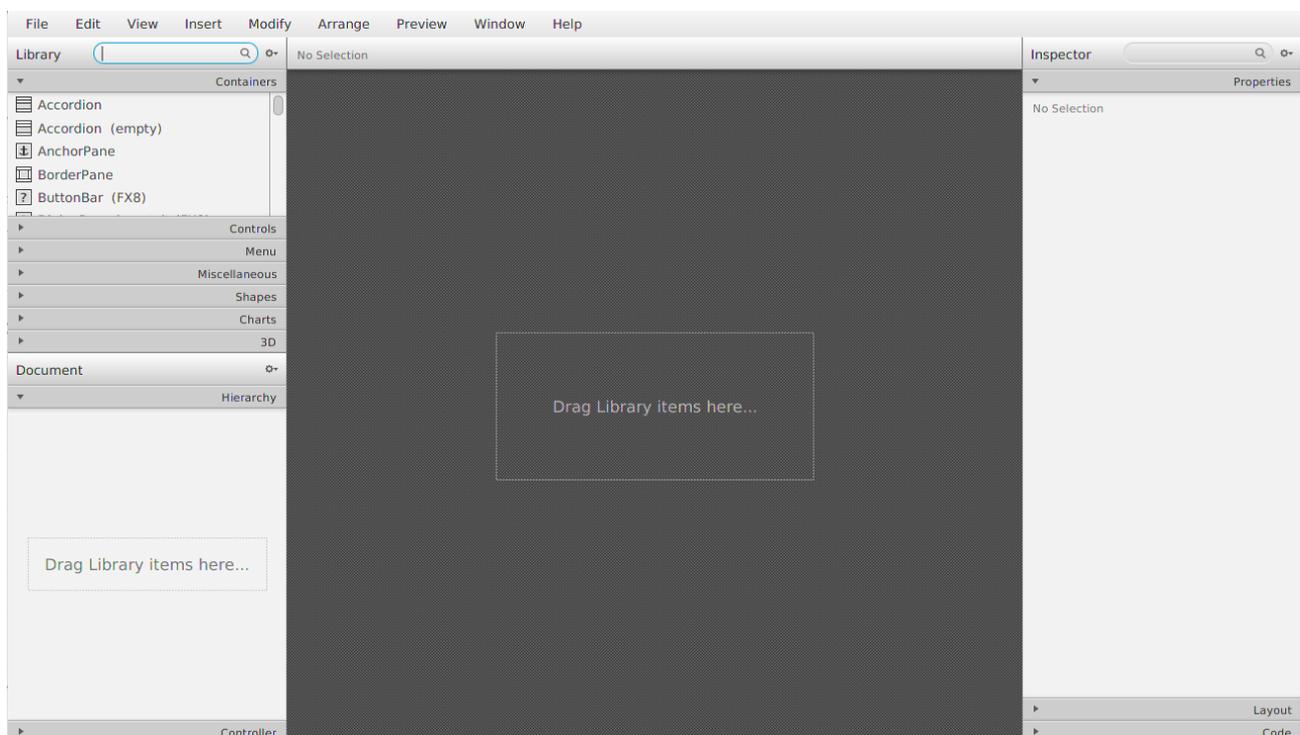


Ilustración 1: Entorno de trabajo de la aplicación Scene Builder

CSS

CSS es un lenguaje de hojas de estilo, diseñado para mejorar la estética de los programas, haciéndolos más atractivos para el público. Este lenguaje es leído por los navegadores o programas para entender el aspecto que el diseñador propone a los diferentes elementos, como puede ser la posición, alineación, tamaño, opciones de fuentes, entre muchas otras.

El uso de este lenguaje en aplicaciones ofrece varios beneficios como es la separación del contenido y la presentación, optimización del tamaño de los archivos, mejoras en la eficiencia de presentación, accesibilidad y personalización según dispositivos.

Este lenguaje se emplea sobre un documento estructurado escrito con un lenguaje de marcado, en este caso generado automáticamente por el Scene Builder.

Se empleará junto a JavaFX, para mejorar la apariencia de la aplicación, y que mantenga unas líneas de estilo que lo unifique.

SQLite

SQLite es una librería que implementa un motor de base de datos SQL. Esta es autocontenida, sin necesidad de servidores y sin configuración. Esto permite que la propia base de datos se enlace al programa pasando a ser parte integral del mismo y reduciendo los tiempos de latencia en las operaciones. Esta librería puede usarse con diferentes lenguajes de programación. También puede usarse en sistemas embebidos debido a su pequeño tamaño y facilidad de uso. Esta escrita en C y es *ACID compliant*.

SQL (Structured Query Language) es un lenguaje específico estandarizado que da acceso a un sistema de gestión de base de datos relacionales, permitiendo especificar varios tipos de operaciones en ellos. Los usos de SQL incluyen la modificación de las tablas y estructuras de índices: añadiendo, actualizando o borrando filas de datos, devolviendo subsets de información de la base de datos para operar después con ellos.

Para su uso se empleará el driver SQLite JDBC, que permite la comunicación entre el presentador y la base de datos.

Itex 5

Itex es una API para la creación de documentos pdf de forma automática. Esto agiliza la creación de pdf, sin intervención alguna, usando un formato definido previamente y haciendo una petición de los parámetros que se son necesarios.

Se usará la versión 5.10, que contiene una gran documentación y numerosos ejemplos de su uso.

Esta librería se ofrece de forma gratuita, bajo la licencia AGPL. Esto no causa ningún problema de licencias, ya que la es aplicación de código abierto.

Gracias a esta librería se va a poder generar los documentos de los ejercicios y las planificaciones de los entrenamientos.

Github

Github es un software de control de versiones vía web y una plataforma colaborativa para programadores. Está basada en Git, que fue creado por Linus Torvalds para desarrollar software de una forma más rápida.

Git es usado para almacenar el código fuente de los proyectos y poder analizar la historia de todos los cambios realizados en el código. Permite a los desarrolladores colaborar de forma más efectiva proveyéndolos de herramientas para manejar posibles conflictos en los cambios. Los repositorios pueden ser públicos o privados, y estos contienen todos los archivos del proyecto y su historial.

Durante el proyecto se ha ido usando este servicio, actualizando cada vez que se conseguía una nueva funcionalidad, o bien corregir algún error.

Desarrollo del proyecto

Para una mejor realización del proyecto, el desarrollo se ha dividido en cinco fases diferentes, para poder ofrecer una solución mejor modularizada, siguiendo los patrones del MVP. Las cinco fases principales son:

Instalación y preparación del entorno de programación, así como documentación necesaria para llevar a cabo el proyecto.

Lógica de datos, donde se diseñará la base de datos a emplear y las consultas para interactuar con ella.

Modelo de vistas, mediante el uso del SceneBuilder se diseñará la apariencia final del software con la que interactuará el usuario.

Desarrollo del presentador, asignando una funcionalidad a los botones de la interfaz. También se marcará las pautas para el tratamiento de los datos, tanto para su presentación en la vista como para su almacenamiento en la base de datos.

Funcionalidades adicionales, como es la internacionalización de la aplicación, la creación de los documentos asociados



Ilustración 2: Modelo de comunicación del patrón MVP

Con estas tres partes bien diferenciadas, se facilita el posterior mantenimiento de la aplicación, tanto para realizar mejoras como para corregir errores.

Instalación y preparación del entorno

La preparación del entorno es el primer paso para la puesta en marcha del proyecto. Una instalación y preparación cuidadosa eliminará posteriores problemas a la hora de programar y agilizará el proceso.

Todo está instalado sobre una máquina con el sistema operativo Ubuntu, en su versión 14.04.

Como IDE se ha elegido usar Netbeans, ya que su uso durante la carrera hace que sea ya conocidas sus funcionalidades, permitiendo un desarrollo más rápido y tiene una fácil integración con otras herramientas. Otros IDEs que se valoraron fueron Eclipse e IntelliJ, pero se descartaron por no conocer todas las funcionalidades que ofrecen y evitar un desarrollo más lento.

Para su descarga, se accede a la página oficial del proyecto, y en su sección de descargas, entre las múltiples opciones de paquetes IDE, se elige la versión Java SE, que incluye la plataforma SDK Netbeans, Java SE y Java FX.

Con el instalador descargado, se ejecuta el fichero .sh mediante la terminal. Al ejecutarlo, se abre un instalador en el cual se acepta la licencia del producto, se configura la ruta de instalación del IDE y del jdk y se confirma su instalación.

Tras la instalación, se abre el IDE para realizar la configuración de los directorios de proyectos. El siguiente paso es SceneBuilder.

Para ello hay que acceder a la página de gluonhq, que es su sección de productos ofrece esta herramienta compilada y adecuada para la versión de Java 8 y preparada para su integración con Netbeans. Se descarga su versión de 64 bits preparada para Linux. Su instalación se gestiona mediante el centro de software. Una vez instalado, hay que asegurarse de que Netbeans detecta la instalación realizada, para comprobarlo, hay que acceder al menú opciones del entorno, y en la categoría de JavaFX, ver si la ruta de Scene Builder es correcta.

Para la creación de la base de datos, se empleará una extensión de firefox, llamada SQLiteManager, que permite la creación y manejo de este tipo de base de datos de una forma fácil e intuitiva. El uso de una extensión para la creación de la base de datos, se debe a la sencillez de esta.

Una vez instalado todos los componentes necesarios, el paso final es la creación del proyecto. Primero se crea el repositorio en Github del proyecto, donde se almacenaran las funcionalidades completadas.

En la página web de Github, después de la creación de la cuenta, da la opción de iniciar un proyecto, al que se asigna un nombre y si se publica o se almacena en privado. Una vez creado, hay que vincularlo al proyecto que se

crea dentro de Netbeans. Se escogerá dentro de la categoría JavaFX, una aplicación JavaFX FXML. Finalmente se vincula el repositorio con el proyecto, gracias a la integración existente entre ambas herramientas.

Por último, queda la creación de la estructura inicial de carpetas que tendrá el proyecto, creando inicialmente tres carpetas llamadas *modelo*, *controller* y *view*.

Lógica de datos

Una parte muy importante de la aplicación es poder generar un banco de ejercicios y jugadas que tener disponibles para visualizar en el momento. Con los años, la cantidad de ejercicios y jugadas que se generan puede llegar a ser un número elevado de información, y más si se usa a nivel de club o federación. Para poder gestionarlo de forma eficiente, se empleará una base de datos SQLite, donde se registra toda la información necesaria. Esta base de datos nos servirá como soporte de la aplicación.

Para ello se accede a la aplicación SQLite Manager, desde la cual con su asistente, se crea la base de datos que se define a continuación.

La base de datos constará de una sola tabla, con los campos siguientes:

Identificador, nombre de la jugada, tipo de jugada, subtipo de jugada, número de páginas, campo usado, descripción de la jugada y coordenadas de los elementos.

Column ID	Name	Type	Not Null	Default Value	Primary Key
0	id	INTEGER	1	null	1
1	Name	TEXT	1	null	0
2	Type	TEXT	1	null	0
3	subType	TEXT	1	null	0
4	number	INTEGER	1	1	0
5	pista	TEXT	1	null	0
6	descripcion	TEXT	0	null	0
7	coordenadas	TEXT	0	null	0

Ilustración 3: Estructura de la base de datos

Se añade una restricción por la cual no puede haber dos jugadas con el mismo nombre, tipo y subtipo, además de configurar el autoincremento del id, y marcar que campos no pueden ser nulos y los valores por defecto.

Tras la creación de la base de datos, el siguiente paso es crear los accesos a esta.

Para ello se crea una serie de operaciones que permita manipular la información de la base de datos, ya sea para crear nuevos registros, eliminar

alguno o bien actualizar información de alguna de ellas.

Para ello se definen 4 operaciones básicas:

Insertar:

```
public int guardarRegistro(Connection connection) {
    try {
        PreparedStatement instruccion
            = connection.prepareStatement("INSERT INTO `playbookData` (`N
            + "VALUES (?, ?, ?, ?, ?, ?, ?)");
        instruccion.setString(1, nombreJugada.get());
        instruccion.setString(2, tipoJugada.get());
        instruccion.setString(3, subtipoJugada.get());
        instruccion.setInt(4, contadorJugada.get());
        instruccion.setString(5, pistaJugada.get());
        instruccion.setString(6, descripcionJugada.get());
        instruccion.setString(7, coordenadasJugada.get());
        instruccion.close();
        return instruccion.executeUpdate();
    }
}
```

Ilustración 4: Operación de insertar un registro nuevo

Eliminar:

```
public int eliminarRegistro(Connection connection) {
    try {
        PreparedStatement instruccion = connection.prepareStatement(
            "DELETE FROM `playbookData` "
            + " WHERE Name = ? AND Type = ? AND subType = ?;"
        );
        instruccion.setString(1, nombreJugada.get());
        instruccion.setString(2, tipoJugada.get());
        instruccion.setString(3, subtipoJugada.get());
        instruccion.close();

        return instruccion.executeUpdate();
    }
}
```

Ilustración 5: Operación de eliminar un registro

Renombrar jugada:

```
public int updateRegistro(Connection connection, String anteriorName, String nombreJugada, String tipoJugada, String subtipoJugada, String pistaJugada) {
    try {
        PreparedStatement instruccion
            = connection.prepareStatement(
                "UPDATE `playbookData` "
                + " SET     `Name` = ?, "
                + " `Type` = ?, "
                + " `subType` = ?, "
                + " `pista` = ? "
                + " WHERE Name = ? AND Type = ? AND subType = ?;"
            );
        instruccion.setString(1, nombreJugada.get());
        instruccion.setString(2, tipoJugada.get());
        instruccion.setString(3, subtipoJugada.get());
        instruccion.setString(4, pistaJugada.get());
        instruccion.setString(5, anteriorName);
        instruccion.setString(6, anteriorTipo);
        instruccion.setString(7, anteriorSubtipo);
        instruccion.close();
    }
}
```

Ilustración 6: Operación de renombrar una jugada

Actualizar:

```
public int updateInformacion(Connection connection) {
    try {
        PreparedStatement instruccion
            = connection.prepareStatement(
                "UPDATE `playbookData` "
                + " SET     `descripcion` = ?, "
                + " `coordenadas` = ?, "
                + " `number` = ? "
                + " WHERE Name = ? AND Type = ? AND subType = ?;"
            );
        instruccion.setString(1, descripcionJugada.get());
        instruccion.setString(2, coordenadasJugada.get());
        instruccion.setInt(3, contadorJugada.get());
        instruccion.setString(4, nombreJugada.get());
        instruccion.setString(5, tipoJugada.get());
        instruccion.setString(6, subtipoJugada.get());
        instruccion.close();
        return instruccion.executeUpdate();
    }
}
```

Ilustración 7: Operación de actualizar la información

Como último paso, se define una clase para poder realizar la conexión a la base de datos. En esta clase carga el driver JDBC específico para esta base de datos, y define el método por el cual establece conexión y otro método para cerrar la conexión. Detecta en todo momento donde está almacenado el fichero sqlite,

configurando la ruta de búsqueda a él.

Modelos de vistas

Después de la creación del modelo de datos, el siguiente paso es definir como el usuario interactuará con la aplicación. Para no saturar al usuario con demasiada información, se ha decidido dividir la aplicación en dos principales partes:

Frame: Esta es la sección principal, donde se puede dibujar el ejercicio o jugada, teniendo al alcance las herramientas. Esta contendrá otra vista en su interior, que será la vista Sketch, donde se dibujará la jugada

Plan: Se accede desde el botón de planificación, desde el cual abre una nueva vista para poder mostrar una ventana con solo la información necesaria.

También tenemos otra vista accesoria para introducir los datos necesarios para almacenar nuevas jugadas, o bien editarlas.

Por tanto, tenemos una vista principal y tres complementarias:

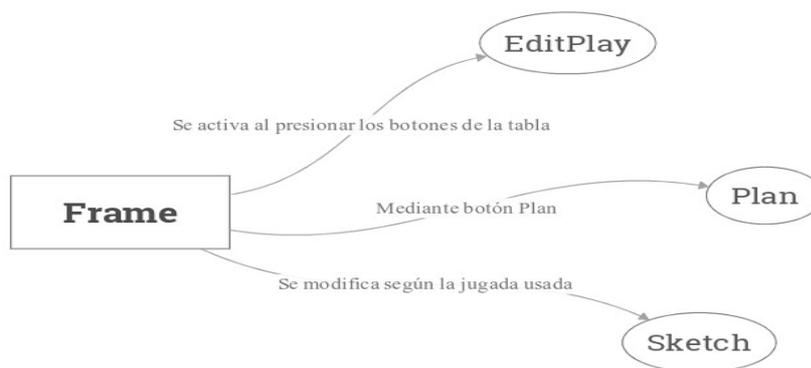


Ilustración 8: Relación entre las diferentes vistas

Vista Frame

La vista del frame puede distinguirse cinco secciones diferentes, donde se agrupan los elementos según sus funcionalidades de los botones.

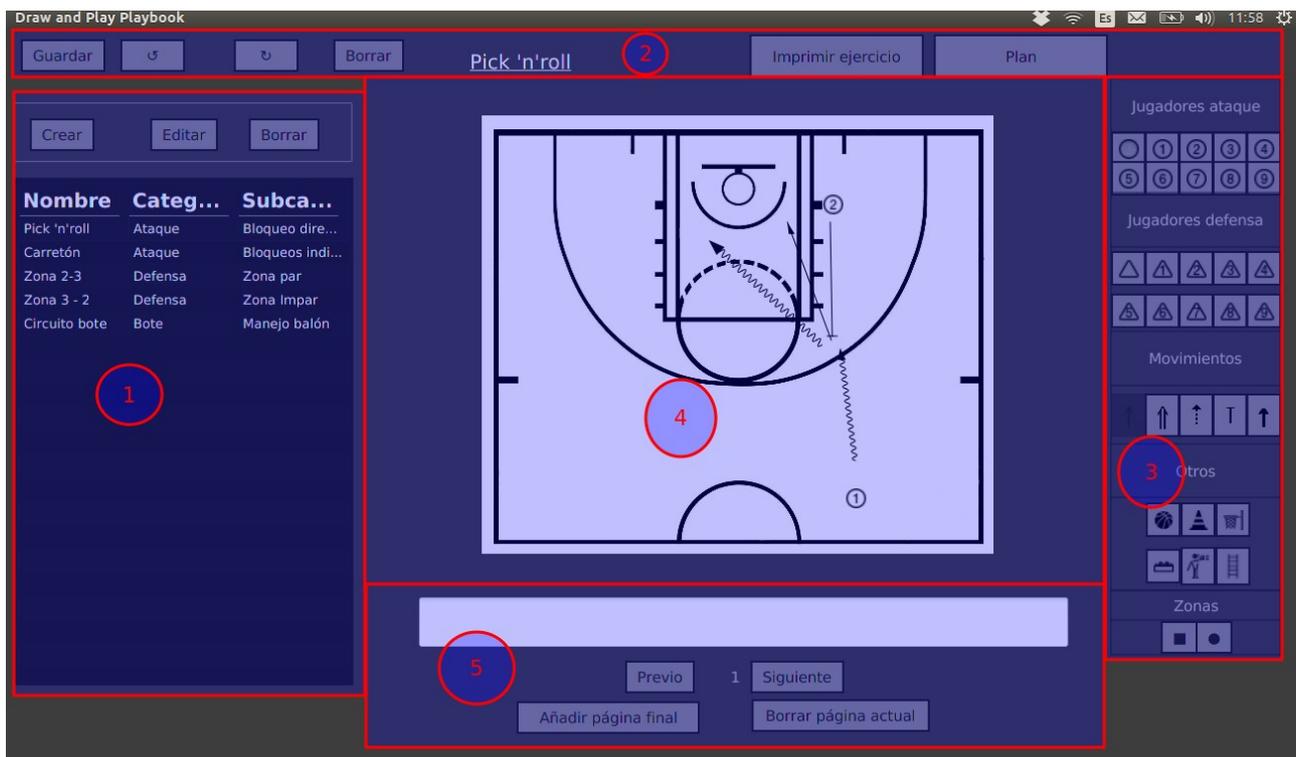


Ilustración 9: Secciones de la vista del Frame

La **sección uno**, contiene una TableView donde se registra todas las jugadas y ejercicios creados hasta el momento. Esta tabla contiene tres columnas, las cuales pueden ser ordenadas alfabéticamente cada una de ellas para realizar una búsqueda más fácil, presionando en el propio nombre de ellas. Esta sección también contiene tres botones, que permiten la creación de una nueva jugada, la modificación de las características de una ya creada, o la eliminación de la seleccionada.

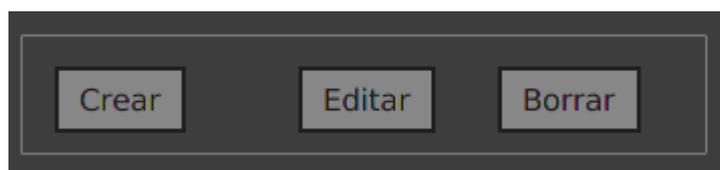


Ilustración 10: Botones de creación, edición y eliminado de jugadas

La **sección dos** agrupa los botones superiores, que tienen las siguientes funciones:

Guardar: permite guardar las modificaciones realizadas a la jugada actual.

Deshacer y Rehacer: estas dos herramientas permiten hacer las funciones que su propio nombre indica, facilitando el editado de las

jugadas.

Borrar: permite borrar un elemento colocado en la zona de dibujo, pulsando sobre él, sin tener así que deshacer todo lo hecho hasta ese momento.

Deshacer y Rehacer: permite un control más fácil de lo realizado en el ejercicio seleccionado.

Imprimir ejercicio: permite la generación de un pdf con la información del ejercicio, todos los dibujos del ejercicio y el texto asociado a cada uno de ellos.

Plan: permite acceder a la ventana para planificar un entrenamiento deportivo.

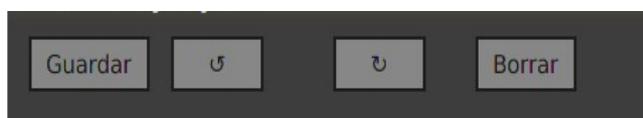


Ilustración 11: Botones de guardar, borrar, deshacer y rehacer



Ilustración 12: Botones de imprimir ejercicio y plan

La **sección tres** tiene todos los elementos posibles para la creación y edición de las jugadas. Se puede ver que están agrupados según la tipología del elemento, facilitando la búsqueda e interacción del usuario. Las agrupaciones son: jugadores de ataque, jugadores de defensa, movimientos, zonas y otros. Cada botón tendrá posteriormente su funcionamiento definido en los controladores.



*Ilustración 13:
Botones de edición de jugadas*

La **sección cuatro** incluye la parte principal de la aplicación, además del título de la jugada seleccionada. Esta zona adaptará un diseño de campo según lo que se elige cuando se crea la jugada. Permitirá la interacción una vez se marque una jugada de la tabla, y un elemento de la sección tres para empezar la edición. Esta sección incluye en su interior la vista Sketch, que se trata de forma independiente al resto de la vista Frame.

A continuación se muestra los tres tipos diferentes de campos a seleccionar:

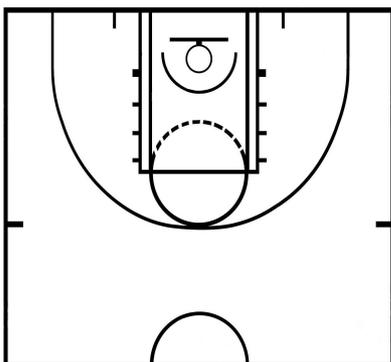


Ilustración 14: Campo de ataque

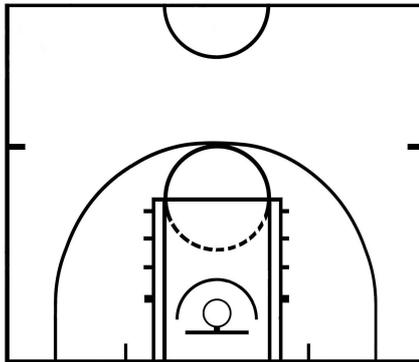


Ilustración 15: Campo de defensa

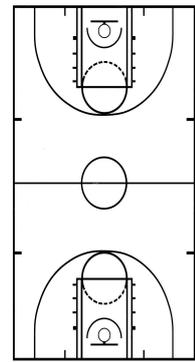


Ilustración 16: Campo entero

La **sección cinco** permite que la jugada tenga más de una página, habilitando

una navegación mediante los botones atrás y delante. También permite añadir una nueva página al final de la jugada y eliminar la página actual. Por último, pero no menos importante, el campo del texto permite asociar una descripción de lo dibujado en esa página.

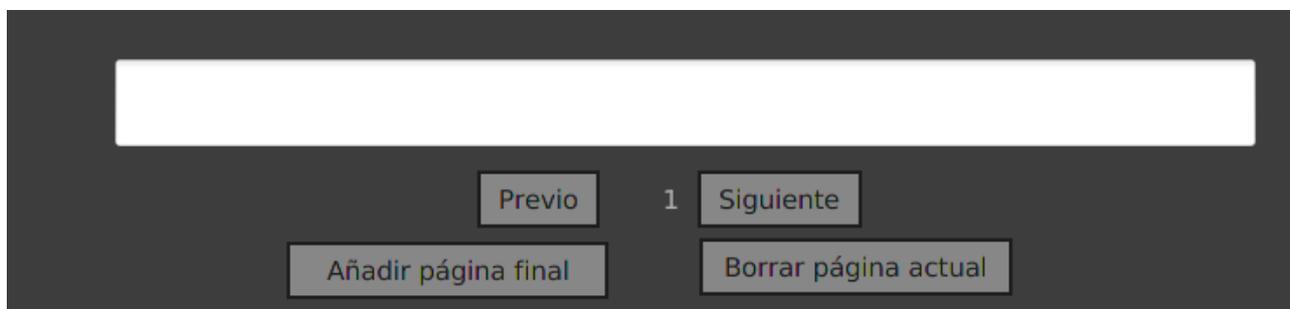


Ilustración 17: Botones de manejo de las páginas de las jugadas, y texto asociado

Vista Plan

La vista plan se activa cuando se presiona el botón de plan de la parte superior. Tras ello, se abre una nueva ventana que permite la creación de un entrenamiento.

Nombre:

Temporada:

Fecha:

Lugar:

Inicio: Fin:

Número de jugadores:

Material:

Añadir nuevo ejercicio
Imprimir plan
 ¿Imprimir ejercicios?

Inicio	Fin	Ejercicio	Notas
16:00	16:10	Circuito bote	Énfasis en uso de la izquierda
16:10	16:30	1c1 ventaja defensa	
16:30	16:40	3c2 uso espacios	Abrir campo al máximo
16:40	17:00	Tiro Australia	Llegar a 120 puntos
17:00	17:10	5c5 individual	
17:10	17:20	Tiro libre	4 seguidos
17:20	17:30	5c5 zona 3-2	

Ilustración 18: Vista del Plan

En esta vista, se pueden incluir los datos necesarios del entrenamiento, como pueden ser: equipo, temporada, fecha, lugar, hora de inicio y fin, número de jugadores y material necesario.

El botón *añadir nuevo ejercicio* permite insertar elementos en la tabla inferior y editarlos posteriormente, y el botón *imprimir plan* genera finalmente el pdf en el sitio indicado, mientras que el checkbox si es presionado detecta de la tabla los ejercicios que están creados y los adjunta al pdf para poder visualizarlos.

Vista PlayEditDialog

Esta vista aparece cuando se quiere crear o editar una jugada, y nos permite ingresar el nombre, la categoría y subcategoría de la jugada, además de elegir el tipo de campo a usar.

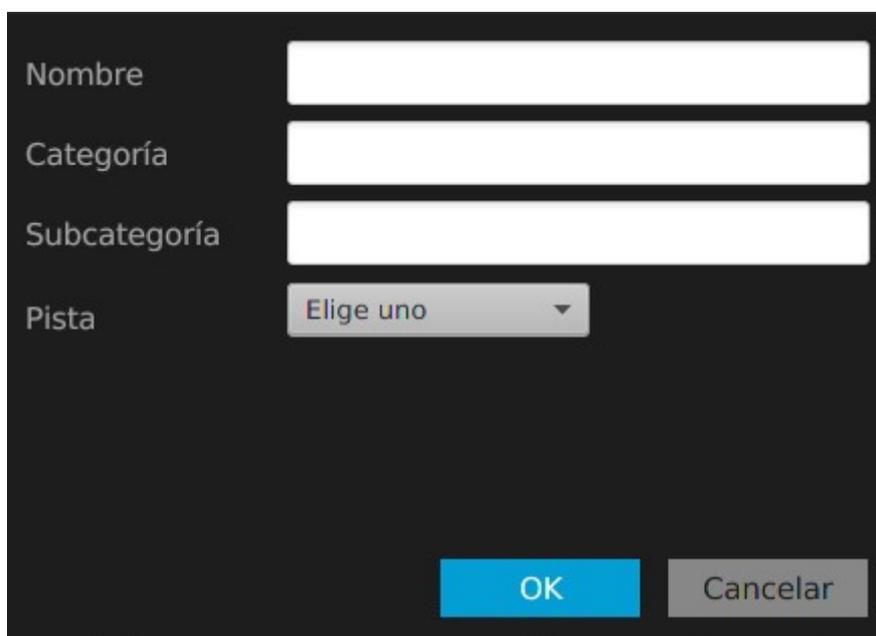
The image shows a dark-themed dialog box titled 'PlayEditDialog'. It contains four input fields stacked vertically. The first three are text boxes labeled 'Nombre', 'Categoría', and 'Subcategoría'. The fourth is a dropdown menu labeled 'Pista' with the text 'Elige uno' and a downward arrow. At the bottom right, there are two buttons: a blue 'OK' button and a grey 'Cancelar' button.

Ilustración 19: Vista del PlayEditDialog

Vista Sketch

Esta vista está incluida dentro de la vista Frame, pero que para una mejor ejecución toda su lógica interna ha sido separado de la otra. Esta separación nos permite cargar correctamente el tipo de pista y los elementos dibujados anteriormente y modificar y registrar los cambios en la jugada.

Desarrollo de los controladores

Teniendo ya creado el modelo de datos y las vistas de usuario, falta implementar las funciones de los botones para crear las jugadas. Para ello se diferencia dos partes:

Tratamiento de los botones, donde desde la petición que se realiza al presionar el botón.

Tratamiento de los datos, donde prepara la información a mostrar por la aplicación.

Para cada Vista se ha creado un controlador asociado que contiene las instrucciones que permiten su funcionamiento, separando así las tres partes de la aplicación.

Tratamiento de los botones

Con el uso de JavaFX y SceneBuilder, cada botón empleado en la vista, ha de ser declarado también en el controlador usando la etiqueta `@FXML` previamente, y vinculándolo a través de un identificador en la vista.

```
//Label del número actual del diagrama
@FXML
private Label playNumberDiagram;

@FXML
private Label labelNombre;

@FXML
private Button delete;

@FXML
private TextArea descripcionArea;
```

Ilustración 20: Ejemplo de declaración con las etiquetas @FXML

Una vez se tiene vinculado todos los botones creados, el siguiente paso es crear las acciones de los botones cuando son presionados, creando una clase para ello y usando también la etiqueta `@FXML` previamente.

```
@FXML
private void coachPressed() {
    setNULL();
    coach.setDisable(true);
    Sketch.getInstance().setShapeType(ShapeType.COACH);
}
```

Ilustración 21: Ejemplo declaración de acción de botón

Los botones incluyen las acciones mínimas, y llaman a otras clases para el tratamiento de los datos, dejándolo en manos de estas el prepararlos para mostrarlos en la vista.

Tratamiento de los datos

Al trabajar con los datos, podemos diferenciar en dos grandes grupos las clases creadas:

Un grupo de clases que preparan los datos para ser mostrados. Estas clases recuperan desde la base de datos los elementos dibujados anteriormente, el tipo de pista y la descripción asociada a los ejercicios, y muestra la sección asociada a la página mostrada.

Un grupo de clases que preparan los datos para ser guardados. Estas clases detectan si se ha realizado alguna modificación desde la carga de la página. En caso de haber un cambio, pregunta si quiere salvar la modificación, y tras tratarlos, los actualiza en la base de datos.

Otro caso aparte son las clases que se dedican a la preparación de los pdf. Estas clases usan la librería iText5, para la preparación de la estructura que tras pasarle los parámetros necesarios generará el documento en el directorio indicado.

Funcionalidades adicionales

Una de las funcionalidades que ya ha sido mencionado anteriormente, es la generación de documentos pdf.

El programa permite la generación de documentos tanto de un ejercicio o jugada de forma individualizada.

También permite la generación de un documento de la planificación de un entrenamiento, con los ejercicios vinculados ya creados anteriormente, lo que permite una gestión más fácil y eficaz, ya que con un único documento tenemos toda la información junta.

Nombre: Zona 3 - 2
 Categoría: Defensa
 Subcategoría: Zona Impar

Diagrama 1

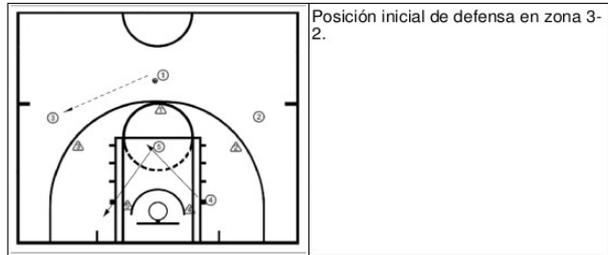


Ilustración 22: Ejemplo de ejercicio imprimido

Equipo:Entrenamiento 22
 Fecha:22 enero
 Inicio:16:00
 Lugar:Centro tecnificación

Temporada:Infantil A fem 16/17
 Jugadores:11
 Fin:17:00
 Material:Conos

Inicio	Fin	Ejercicio	Notas
16:00	16:10	Calentamiento	
16:10	16:30	Rueda entradas	
16:30	16:45	1c1	
16:45	17:00	Zona 3 - 2	

Ejercicio Zona 3 - 2

Diagrama 1

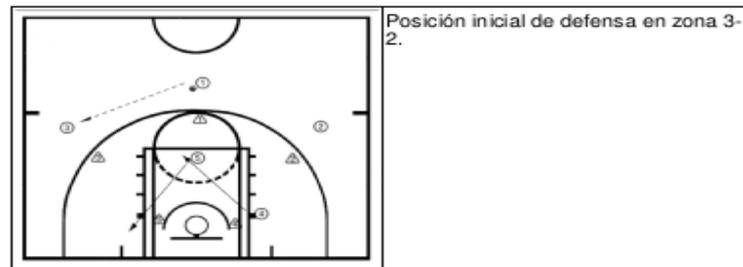


Ilustración 23: Ejemplo de una planificación de entrenamiento

Otra funcionalidad adicional que se ha desarrollado ha sido la internacionalización de la aplicación, preparándola mediante los archivos de propiedades, guardando todos los textos en diferentes idiomas (actualmente

en inglés y castellano), y detectando automáticamente el idioma configurado en el ordenador instalado, para presentarlo en ese idioma. En caso de no tener el idioma disponible, por defecto se usa inglés.

Presupuesto

Para la creación del proyecto, se ha calculado un presupuesto orientativo del coste que tendría realizar esta aplicación.

El coste de hardware sería mínimo o nulo, ya que no se requiere uso de servidores. Sólo se necesitaría el uso de un ordenador para el desarrollo del proyecto.

El coste en licencias también es nulo, ya que durante la toma de decisiones de las tecnologías empleadas se ha decidido evitar este tipo de gastos.

El tiempo empleado para el desarrollo del proyecto se estima en 300 horas, incluido el tiempo de investigación, diseño y programación. Estimamos un precio de 6 euros por hora trabajada, lo que ofrece una cifra de 1800 euros como coste total del desarrollo de este programa.

Conclusiones

El objetivo del trabajo era la creación e implementación de una aplicación de escritorio que permitiera el diseño y planificación de un entrenamiento deportivo basado en el baloncesto.

Tras la finalización del proyecto, se puede desarrollar completamente los objetivos marcados para este, manteniendo los objetivos de sencillez y claridad para usuarios no expertos.

La aplicación también garantiza el acceso y clasificación de los ejercicios de forma eficaz y cómoda.

Así pues, gracias a este proyecto, los entrenadores de baloncesto tienen una opción 100% libre para la generación de estas tareas, algo vital ya que en esta profesión la remuneración muchas veces es mínima o inexistente.

Hay que destacar que se ha podido usar diferentes conocimientos adquiridos en diferentes asignaturas durante el grado en informática, como principios de diseño de interfaces, el uso de java y sus librerías y principios de diseño de software.

También remarcar que gracias al proyecto realizado, me ha permitido reforzar los conocimientos que tenía sobre Java, en la versión 8, y base de datos, o bien aprender otras como es el uso de CSS o javaFX

Mejoras para el futuro

Como proyecto pensado por mí, y como primer interesado en la aplicación,

pretendo que no se quede estancado su desarrollo.

Tras sacar la primera versión disponible, ya se está trabajando en una segunda versión, incluyendo mejoras que se quedaron fuera de esta por cuestiones de tiempo.

Algunas de las mejoras ya decididas para esta siguiente versión serían:

- Posibilidad de incluir líneas de movimiento curvas.

- Creación de plantillas para agilizar el uso de formaciones

- Modificaciones de tipografía

- Mejoras en el campo de descripción: permitir uso de negrita, cursiva, etc.

- Poder visualizar los elementos dibujados en la anterior página sin tener que cambiar.

- Creación de copias de seguridad

- Exportar e importar librerías.

- Guardar registro de las planificaciones de entrenamientos.

Por otra parte, cómo proyecto colgado en github, también se puede colaborar en él o sugerir mejoras.

Sharan, K. (2015). *Learn JavaFX 8* (1st ed.).

SQL Tutorial. *W3schools.com*. Retrieved 16 December 2016, from <https://www.w3schools.com/sql/default.asp>

SQLite Home Page. *Sqlite.org*. Retrieved 15 December 2016, from <https://www.sqlite.org/>

Vos, J., Gao, W., Chin, S., Iverson, D., & Weaver, J. (2014). *Pro JavaFX 8* (1st ed.). New York: Apress.

Basketball Playbook: Software, Plays and Drills. (2017). *Jes-soft.com*. Retrieved 28 February 2017, from <http://www.jes-soft.com/playbook/>

Big Data meets big-time basketball – The Mercury News. (2014). *Mercurynews.com*. Retrieved 28 February 2017, from <http://www.mercurynews.com/2014/05/17/big-data-meets-big-time-basketball/>

FastDraw® Basketball Play Diagramming Software. (2017). *FastModel Sports*. Retrieved 28 February 2017, from <http://fastmodelsports.com/products/fastdraw-play-diagramming-software>

Blog | code.makery.ch. (2017). *Code.makery.ch*. Retrieved 28 February 2017, from <http://code.makery.ch/blog/>

Scene Builder - Gluon. (2017). *Gluon*. Retrieved 28 February 2017, from <http://gluonhq.com/products/scene-builder/>

Anexos

Localización del repositorio en Github

Todo el código del proyecto realizado se puede encontrar subido a la página web de repositorios Github.

<https://github.com/davidgarciajuan/Draw-and-Play-Playbook>

Manual de usuario

Una vez se inicia el programa, aparece la siguiente ventana:

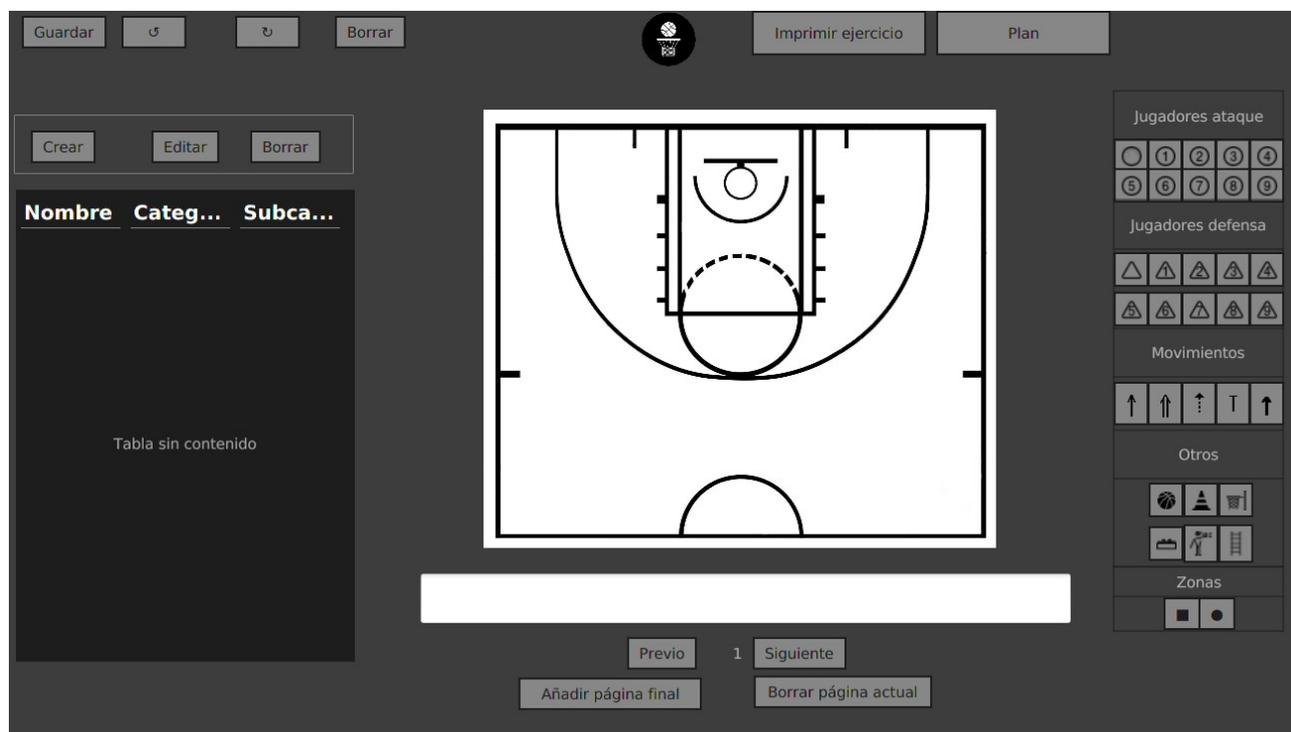


Ilustración 24: Ventana principal una vez iniciado el programa

Esta ventana podemos distinguir 4 zonas:

En la parte superior encontramos una serie de botones. A la izquierda se encuentran los siguientes: guardar, deshacer/rehacer y borrar, mientras que a la derecha tenemos: imprimir ejercicio y plan, que sirve para hacer planificaciones.

Una tabla a la izquierda, donde se almacenarán las jugadas creadas, y sus botones de creación, edición y eliminación, que están inmediatamente arriba.

A la derecha están las diferentes opciones para la edición de los ejercicios/jugadas.

En la zona central se encuentra la zona donde se dibujaría el ejercicio, debajo de esta un campo para añadir texto y por último cuatro botones para pasar de página y crear o borrar alguna página.

Creación de una jugada

Para la creación de una jugada, el proceso es simple:

Presionar el botón de crear de la parte superior de la tabla. Mostrará la siguiente ventana:

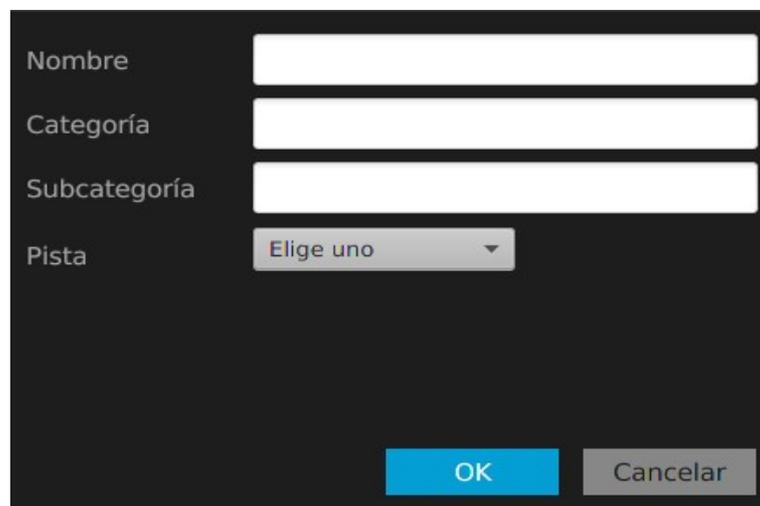
Una ventana de diálogo con fondo oscuro. Contiene cuatro campos de entrada: 'Nombre' (campo de texto), 'Categoría' (campo de texto), 'Subcategoría' (campo de texto) y 'Pista' (menú desplegable con el texto 'Elige uno' y una flecha hacia abajo). En la parte inferior derecha hay dos botones: 'OK' (de color azul) y 'Cancelar' (de color gris).

Ilustración 25: Ventana de creación de jugada

Escribir un nombre, categoría y subcategoría y seleccionar un campo.

Al presionar ok con los campos rellenos, la jugada pasará a estar disponible en la tabla.

Con el botón de editar, se podrá modificar esta información posteriormente, si hiciera falta.

El siguiente paso será seleccionarla en la lista y empezar a colocar los elementos que se quieran.

Edición de una jugada

Para la edición de una jugada tenemos diferentes elementos.

A la derecha se encuentran los elementos que se pueden colocar en el campo de dibujo. Se pueden diferenciar cinco grupos: los jugadores de ataque, los jugadores de defensa, las flechas de movimiento, las zonas y otros.

Para colocarlos hay que presionar al elemento, y tras ello este elemento estará activo y será colocarlo en el campo. En el caso de las flechas, hay que presionar donde se inicia y donde acaba.

En la parte superior tenemos botones de ayuda, como son el deshacer, rehacer y borrar, además de la opción de guardar lo hecho.

Planificar un entrenamiento

Para planificar un entrenamiento, el primer paso es presionar el botón superior derecho, que se llama plan. Mostrará la siguiente pantalla:

Nombre: Número de jugadores:

Temporada:

Fecha: Material:

Lugar:

Inicio: Fin:

¿Imprimir ejercicios?

Inicio	Fin	Ejercicio	Notas
Tabla sin contenido			

Ilustración 26: Pantalla de creación de una planificación

La primera parte añade información que puede ser necesaria para el entrenamiento. Solo se mostrarán después los campos rellenos

Para añadir un ejercicio hay que presionar el botón, e incluirá una fila en la tabla con los campos para editar.

Para editarlos, simplemente hay que presionar sobre él, escribir lo que se quiere y pulsar la tecla intro.

Si se quiere imprimir los ejercicios hay que marcar la caja al lado del botón de imprimir. Para este caso, hay que haber creado primero el ejercicio, y escribir el nombre de forma exacta a la almacenada en la tabla.