INFORME PROYECTO DE PROGRAMACIÓN

Proyecto realizado por:

Mikel Lerena Tapia & Javier Marín Díaz de Guereñu

Nombre del proyecto/aplicación: Basket Live

1º INDUSTRIA DIGITAL
ASIGNATURA: PROGRAMACIÓN II

31/05/2019

ÍNDICE

ASPECTOS TÉCNICOS

CONTEXTO Y MOTIVACIÓN DE LA APLICACIÓN

El nombre de la aplicación es Basket Live, y está dirigida al mundo del deporte, concretamente especializada en el baloncesto.

Actualmente, existen una infinidad de deportes, sin embargo, los más conocidos por la sociedad son los deportes de equipo, como por ejemplo el fútbol, baloncesto, waterpolo,etc.

Esta aplicación se ha creado con el propósito de informar y divertir a personas que tengan un cierto afán por el mundo del baloncesto.

Basket live ofrece la posibilidad de darse de alta a nuevos usuarios, introduciendo una serie de datos, además de su equipo favorito, del cual tendrá la posibilidad de visualizar las noticias más exclusivas, proporcionando al usuario una cierta satisfacción y ganas futuras de volver a entrar y seguir utilizando la aplicación.

Basket Live debe ser una aplicación intuitiva, fácil de usar por cualquier usuario,que ofrezca las últimas noticias/novedades de los jugadores de la liga, que de la posibilidad de que un usuario pueda ver la clasificación de los equipos y las jornadas, es decir, que equipos juegan contra quien, además de otros detalles que se irán comentando a lo largo del informe.

ESTRUCTURA DE LA APLICACIÓN DESARROLLADA

Para la administración de la aplicación se han empleado dos plataformas ligadas que son GitHub y GitKraken.

GitHub es una plataforma online en la cual mediante la creación de un repositorio, se almacena toda la información que se quiera.

GitKraken es una plataforma en la cual se clona el repositorio de GitHub para que cuando se realicen cambios en cualquier archivo de ese repositorio, se guarden y se suban a GitHub. De esta forma todos componentes del proyecto tendrán toda la información actualizada a tiempo real.

Una vez organizada toda la información necesaria se empezará con el proyecto.

Lo primero que se hizo fue pensar en una idea de proyecto y plasmarla en una Base de Datos. Se comenzaron a crear todas las tablas con sus diferentes campos y a realizar las relaciones entre ellas. Además había que tener en cuenta que esa no iba a ser la estructura final de la BD y que iban a surgir muchos cambios a lo largo del proyecto.

Lo siguiente que se hizo fue crear los paquetes en los que iba a estar dividida la aplicación:

-Paquete (comun): En este se encuentran todas las clases que van a ser utilizadas por otras clases de toda la aplicación.

- cls Constantes: En esta clase se encuentran todas las constantes que se van a asignar a las clases que están en el paquete LN.
- itfProperty: Esta es una clase que actúa como interfaz que se implementará en las clases del paquete LN para que se pueda acceder a los atributos desde el paquete LP.
- itfPersistable: Es una interfaz que se usará para clases que se vayan a almacenar en la BD.
- Utilidades: Es una clase que se utilizará para generar el calendario en el Gestor de LN.

-Paquete (Excepciones): En este se van a encontrar todas las clases que lancen o hereden una excepción para poder tratarlas en la misma clase o donde se lancen.

- EquiposInsuficientesException: Esta clase se ha creado para poder lanzar una excepción cuando , en este caso, no haya equipos suficientes para montar la liga. En el mensaje que lanza la excepción se exige que haya un número par de equipos para que a la hora de programar las jornadas/partidos, todos los equipos tengan con quien jugar, y evitamos que haya descanso en cada jornada.
- PropiedadIncorrecta: Excepción de tipo Runtime que indica que la propiedad seleccionada no existe para dicho objeto itfProperty en su clase original, básicamente estaríamos tratando por ejemplo un artículo, como si fuera un alumno

-Paquete (LD): Este es el paquete que se conoce como Lógica de diseño. Es donde residen los datos y es el encargado de acceder a los mismos. Está formado por un gestor de bases de datos que realiza todo el almacenamiento de datos, recibe solicitudes de almacenamiento o recuperación de información desde el paquete LN. Estos son sus componentes:

- CampoBD: Clase que se utilizará para realizar la inserción y selección de los datos de los campos en la BD.
- ClasificacionBD: Clase que se utilizará para seleccionar de la BD los datos necesarios para conformar la tabla de la clasificación de la liga de la aplicación.
- clsDatos: Clase que se va a encargar de gestionar la comunicación entre el paquete LD y el paquete LN
- Conexion: Clase que va a gestionar la comunicación entre el paquete de LD y la base de datos.
- constantesBD: Clase creada para asignar constantes a la clase abstracta Conexion.(Las constantes necesarias para conectarse a la BD).
- EquipoBD: Clase que se empleará para insertar y obtener información sobre los equipos de la BD.
- EstadoBD: Clase que se utilizará para insertar los diferentes estados en los que se pueda encontrar un jugador.

- EventoBD: Clase que va a gestionar la situación de cada jugador de la BD.
- JugadorBD: Clase que se empleará para realizar la inserción y obtener la información de los jugadores de los diferentes equipos de la BD.
- PartidoBD: Esta clase se va a utilizar para realizar multiInserts y obtener información de los partidos de la BD.
- PosicionBD: Clase que se utilizará para gestionar la inserción y la obtención de la información de las diferentes posiciones de la BD.
- TemporadaBD: Esta clase contiene métodos que utilizaremos para insertar y obtener información sobre la temporada en la BD.
- TraspasoBD: Clase que se empleará para obtener toda la información sobre los traspasos de la BD.
- UsuarioBD: Clase que se utilizará para registrar y actualizar usuarios, además de obtener la información del usuario logeado.

-Paquete (LN): Este es el paquete que se conoce como Lógica de negocio. Es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con LP, para recibir las solicitudes y presentar los resultados, y con LD, para solicitar al gestor de base de datos almacenar o recuperar datos de él. Estos son sus componentes:

- Aficionado: Clase que se va a utilizar para gestionar la información de los usuarios no administradores, es decir, un usuario cualquiera que podrá elegir su equipo favorito.
- Campo: Clase que va a gestionar los campos en los que se van a jugar los partidos.
- Clasificacion: Clase que se va a encargar de clasificar los participantes de la liga.
- Equipo: Clase encargada de gestionar los equipos de nuestra liga.
- Estado: Esta clase va a gestionar los posibles estados de los diferentes jugadores.
- Evento: Esta clase va a gestionar los diferentes eventos o noticias referidas a los jugadores, en concreto, en que estado se encuentran tanto previamente como posteriormente.
- GestorLN: Clase que se va a encargar de la gestión de la comunicación entre el paquete LN y LP.
- Jugador: Clase que va a gestionar los jugadores guardando sus datos.
- Noticia: Clase que se encargará de guardar los datos referentes a las noticias de la aplicación.
- Participantes: Clase que se utilizará para asignar participantes a la liga.
- Partido: Clase que se va a encargar de gestionar los datos principales de los partidos.
- Posicion: Clase que se utilizará para determinar las posiciones de los diferentes jugadores.
- Temporada: Esta clase va a gestionar las temporadas que vaya a tener nuestra aplicación.
- Traspaso: Clase encargada de gestionar los traspasos de los jugadores entre distintos equipos de la liga.
- Usuario: Esta clase se va a encargar de gestionar todas las cuentas de los usuarios.

-Paquete (LP): Este es el paquete que se conoce como Lógica de presentación. Presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso. Esta capa se comunica únicamente con LN. Estos son sus componentes:

- Paquete imagenes: En el se encuentran todas las imagenes e iconos que se van a utilizar en la aplicación. Además, cuenta con una clase “clsImagen”, que va a ser la encargada de dar el formato a las imagenes que vayan en el fondo de la ventana.
- Paquete InsertarPorVentanas: En él se encuentran las ventanas que van a ser visibles única y exclusivamente por los administradores. Se han creado para poder insertar los jugadores, equipos, campos, posiciones y estados, de una manera rápida y sencilla, sin tener que usar el MySQL Workbench directamente.
- Paquete Updates: En él se encuentran las ventanas que vayan a suponer un update en la BD. En updateEquipo damos la posibilidad de que el usuario normal pueda cambiar de equipo favorito cuando le plazca de una manera muy sencilla. En updateJugador se pueden actualizar los datos de cada jugador de forma que los administradores tengan una forma rápida de realizar dichos cambios. En ListaJugadoresModel se encuentra el código referente al relleno de la JList de la ventana updateJugadores.
- ClasificacionModel: Clase que utilizaremos para crear la tabla de la clasificación de la liga.
- clsUtilidades: Esta clase permite hacer uso de los dispositivos de entrada de forma muy sencilla. Posee distintos métodos públicos y estáticos (por lo tanto no se necesita un objeto como tal para acceder a ellos) para la lectura de los tipos de datos básicos en Java. Además, todos los métodos tienen un control de errores mediante petición sucesiva de introducción de dato hasta que estos sean correctos.
- Menu: Clase intermediaria que se encargará de comunicar el main con el resto del programa.
- NoticiaComponent: Clase que se va a encargar de mostrar todas las noticias de la aplicación.
- ventanaComienzoApk: Clase encargada de diseñar la ventana del inicio de la aplicación.
- ventanaMenu: Esta clase es la encargada de diseñar la ventana que corresponde al menú principal de la aplicación
- wdwClasificacion: Clase que se empleará para diseñar la ventana que contendrá la tabla con la clasificación de la liga.
- wdwDatosPersonales: Clase encargada de diseñar la ventana en la que aparecerán los datos personales del usuario.
- wdwMarcador: Clase que se utilizará para visualizar el marcador de un partido en concreto.
- wdwNoticias: Clase encargada de diseñar la ventana en la que se visualizarán las noticias de la aplicación.
- wdwRegistrarUsuario: Clase que se va a encargar de diseñar la ventana en la que un usuario podrá registrarse y/o logearse para poder utilizar la aplicación.

DETALLES TÉCNICOS

La aplicación desarrollada pretende ser un punto de acceso a toda la información sobre la evolución de una liga de baloncesto. La aplicación cuenta con un sistema para generar los partidos, de forma que todos los equipos jueguen contra todos.

Nuestra aplicación gestionará los equipos, junto a sus partidos y resultados, los jugadores y llevará un seguimiento de su estado y equipo de forma que cuando estos evolucionen, se notificará a los aficionados de forma automática (este sistema basado en triggers de sql). Además los usuarios, podrán seleccionar su equipo favorito para que las noticias relativas a este se vean destacadas.

Para el desarrollo de esta aplicación hemos segmentado en tres capas diferentes, diferenciando el código según su cometido; diferenciando así la lógica de presentación(Referente a la interfaz gráfica y la interacción con el usuario), la lógica de negocio(referente a cómo debe comportarse la aplicación) y la lógica de datos la cual hace posible la persistencia de los datos y gestión de los mismos.

La aplicación carga los datos en memoria desde la Base de Datos al iniciar y solo vuelve a hacer uso de esta para hacer actualizaciones sobre la misma, de esta forma reducimos la penalización en rendimiento en ambos lados, tanto en la aplicación como en la base de datos, ya que al ser una aplicación pensada para un público tan amplio, es posible que en determinadas franjas horarias la cantidad de usuarios concurrentes usando la aplicación sea elevada afectando al rendimiento de la misma.

Además nos hemos valido de herramientas externas como mysql-connector para la capa de datos y jdatechooser para brindarle a los usuarios una interfaz más amistosa para introducir fechas.

Para la programación del proyecto nos hemos apoyado en varias estructuras que Java trae por defecto:

- Interfaces: hemos usado una interfaz llamada itfProperty con la que comunicamos nuestras capas LP y LN de forma que la LP no se vea afectada cuando se hagan cambios en LN y viceversa. Además hemos usado otra interfaz llamada persistable, que implementaremos en todas las clases que queramos recuperar de BD, esta interfaz nos asegurará el método resultSetLoad, el cual básicamente recupera los datos desde un resultSet y los cargará en un objeto.
- Herencia: hemos utilizado las herencias de java en dos clases destacables, Usuario y Aficionado; relación que usaremos para diferenciar los distintos permisos que tendrá el usuario logeado. Por otro lado hemos usado la superclase noticia para modelar Evento y Traspaso, clases que juntaremos usando la clase padre para mostrar como noticias en una misma lista.
- Algoritmos de ordenamiento: para los ordenamientos por ejemplo en la clasificacion o en las noticias más recientes, hemos usado el método Collections.sort usando las interfaces Comparable y Comparator, con las que asignamos los criterios de ordenación que hemos estimado oportunos en cada caso.
- Acceso a Datos: para el acceso a datos hemos programado una serie de clases y métodos, para facilitarnos el desarrollo, (clase conexion) la que implementa los métodos insert, update, select y

delete. De esta forma generamos menos código ‘boilerplate’ y programaremos más rápido.

PLANIFICACIÓN

La planificación la hemos gestionado a través de una plantilla de Excel

En cuanto a la al desarrollo no ha habido una gran desviación en cuanto a los objetivos iniciales, puesto que nos hemos ceñido bastante bien al plan inicial. Aun así se nos han quedado ideas en el tintero, ideas que enumeramos a continuación:

- Fichas de Jugador: añadir una ventana en la que visualizaremos los datos de un jugador, como sus estadísticas, foto, equipos en los que ha jugado o noticias relacionadas con el.
- Fichas de estadio: poder visualizar los datos de un estadio como su dirección o próximos partidos que se jugarán ahi.
- Fichas de equipo: poder mostrar la plantilla de jugadores que juegan para ese equipo, su clasificación o las clasificaciones de sus últimos años.