

Presentación proyecto Casa de apuestas (Bets)



En este documento se presenta la información del proyecto desarrollado hasta el momento:

1. Descripción inicial del sistema de información
2. Modelo del Dominio
3. Modelo de Casos de Uso
4. Flujos de Eventos
5. Diagrama de Secuencia de uno de los casos de uso
6. Implementación: código fuente y documentación, generación del .jar

1. Descripción inicial del sistema de información

Se desea desarrollar una aplicación de gestión de apuestas deportivas.

El administrador del sistema determina los eventos (por ejemplo, partido Real Madrid-Barcelona), las apuestas posibles (por ejemplo, ganador del partido o quién marcará el primer gol) así como los posibles pronósticos para cada una de las apuestas y el valor de ganancia por euro apostado. Por ejemplo, sobre el evento “partido Real Madrid-Barcelona” y la apuesta “ganador del partido” se pueden definir los siguientes pronósticos con porcentajes de ganancia diferentes.

Evento: Real Madrid vs Barcelona
Apuesta: ¿Quién ganará el partido? (1-X-2)
Pronóstico: Real Madrid (1)
Ganancia: 1,20 €

Evento: Real Madrid vs Barcelona
Apuesta: ¿Quién ganará el partido? (1-X-2)
Pronóstico: Barcelona (2)
Ganancia: 2 €

Evento: Real Madrid vs Barcelona
Apuesta: ¿Quién ganará el partido? (1-X-2)
Pronóstico: Empate (x)
Ganancia: 1,5 €

Un *evento* se caracteriza por una *descripción*, así como su *fecha* y *hora* de comienzo. Sobre un evento se pueden *establecer diferentes apuestas*. Una *apuesta* viene descrita por una *pregunta* sobre un evento, así como la cantidad de apuesta mínima a realizar, mientras que un

pronóstico establece un resultado a una apuesta, así como un valor en euros de ganancia por euro apostado.

Los usuarios registrados en el sistema, pueden invertir sobre los pronósticos establecidos siempre y cuando se cumplan los siguientes requisitos:

- a) Todavía no se ha cerrado el evento sobre el cual está basado ese pronóstico. Un evento se cierra en el momento de comenzar dicho evento.
- b) La cantidad a apostar es mayor que la cantidad mínima establecida para esa apuesta.
- c) Tiene suficiente dinero en su monedero para invertir en la apuesta. El dinero invertido no estará disponible para realizar una nueva apuesta, es decir estará congelado hasta que el administrador finalice el evento.

Un ejemplo completo de apuesta sería:

Evento: Real Madrid vs Barcelona
Apuesta: ¿Quién ganará el partido?
Pronóstico: Real Madrid (1)
Inversión: 30 € (si acierta ganará 36, en caso contrario perderá 30 €)

Cuando un evento ha finalizado (por ejemplo, ha acabado el partido), el administrador introduce los resultados referente a las apuestas (por ejemplo, ha habido un empate entre el Real Madrid y el Barcelona), y el sistema actualiza los monederos de los clientes en base las apuestas que hayan realizado. Si el resultado coincide con el pronóstico apostado, su monedero será incrementado con la cantidad apostada multiplicado por el valor de ganancia, en caso contrario, pierde todo lo apostado.

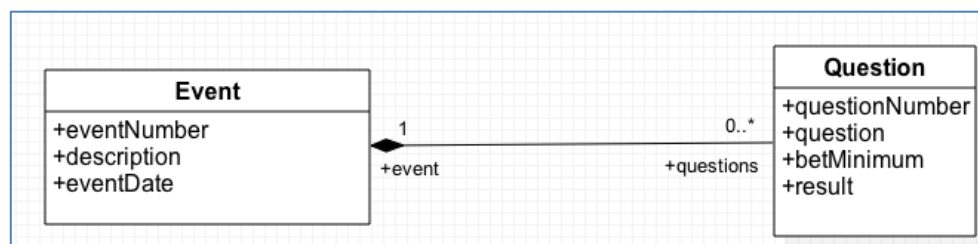
En todo momento el usuario puede consultar todas las apuestas que ha realizado, su estado (finalizada o no), y en caso de finalización su rendimiento.

NOTAS:

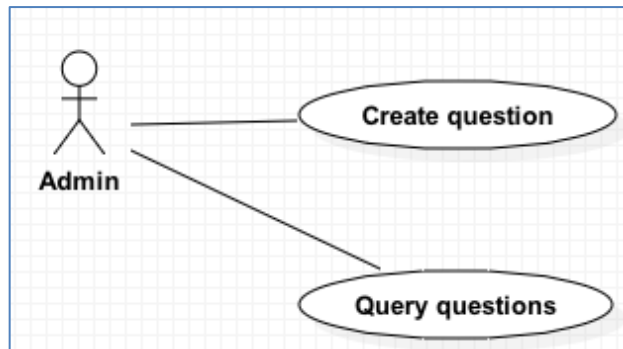
Se presentan los artefactos correspondientes al estado actual del proyecto: el modelo del dominio, el modelo de casos de uso, los flujos de eventos de 2 casos de uso desarrollados y el diagrama de secuencia para uno de esos casos de uso.

Sin embargo, no se han realizado todos los casos de uso que corresponderían a la descripción anterior, ni todos los datos necesarios aparecen reflejados en el modelo del dominio.

2. Modelo del dominio



3. Modelo de Casos de Uso



4. Flujos de Eventos

Query question Flow of events

Basic Flow

1. *System* shows a Calendar where days with events are highlighted
2. *Admin* selects a **Date** in a Calendar
3. *System* displays the **events** of this **date**
4. *Admin* selects an **event**
5. *System* displays the **questions** associated to the selected **event**

Alternative flow

1. There are no events on this date. Events cannot be shown.
2. There are no questions associated to the event. Questions cannot be shown.

Create question Flow of events

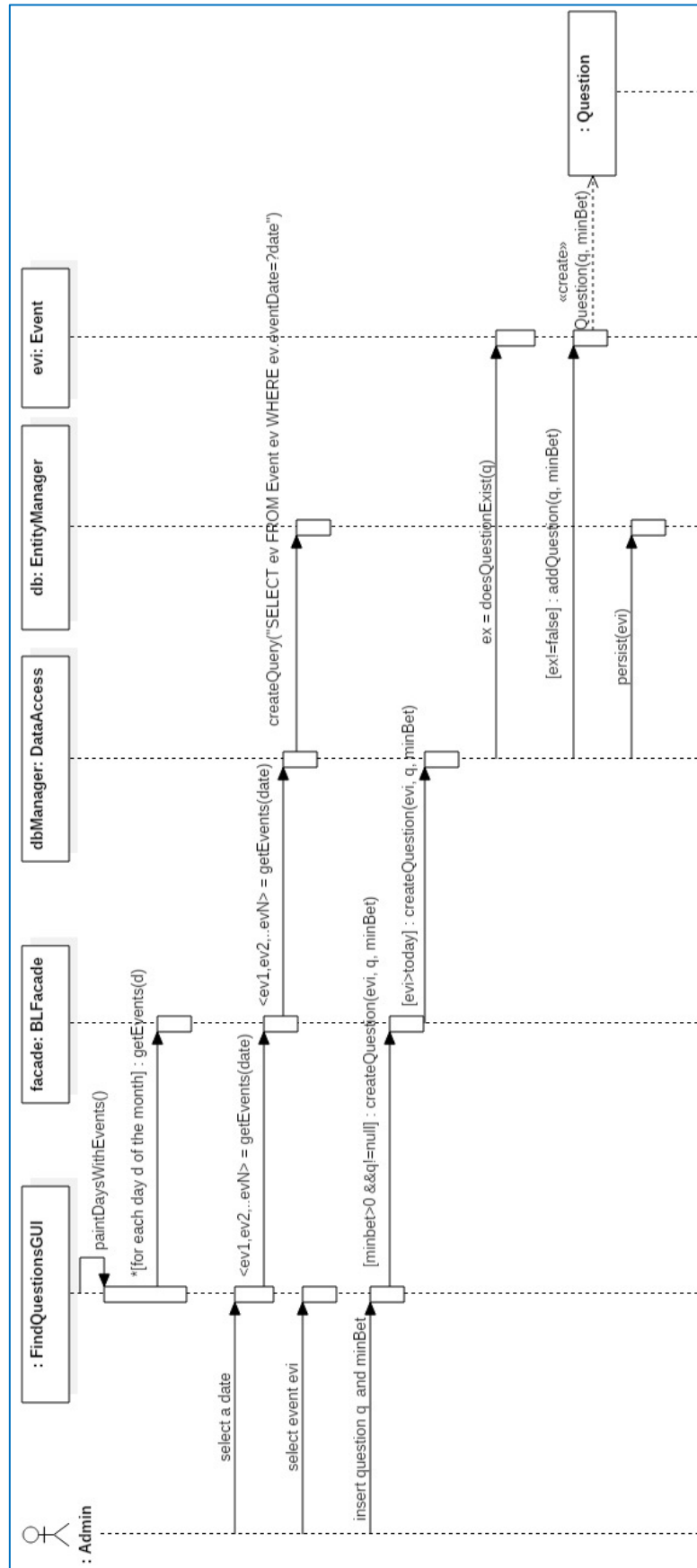
Basic Flow

1. *System* shows a Calendar where days with events are highlighted
2. *Admin* selects a **Date** in a Calendar
3. *System* displays the **events** of this **date**
4. *Admin* selects an **event**
5. *Admin* introduces a **question** and a minimum **betting price**
6. *System* adds the new **question** with a minimum **betting price** to the selected **event**

Alternative flow

1. There are no events on this date. Question cannot be added.
2. The question field is empty. Question cannot be added.
3. The minimum betting price is empty or it is not a number. Question cannot be added.
4. Event date has already finished (event day is before current day). Question cannot be added.

5. Diagrama de secuencia del caso de uso "Create question"



6. Código fuente.

El código fuente de la aplicación de la casa de apuestas se encuentra disponible en eGela pestaña PROYECTO BETS (archivo Bets2021-master.zip). El proyecto contiene entre otros: el código fuente Java (src/main/java), las librerías que trabajan con calendarios JCalendar-jcalendar-1.4.jar y con bases de datos objectDB- objectdb.jar (referenced Libraries), y algunos ficheros que se encuentran en el directorio resources del proyecto (src/main/resources). Aparecen entre otros el fichero donde se almacenará la base de datos (bets.temp) y un fichero XML con datos de configuración para la aplicación de apuestas (config.xml).

Para usar el proyecto en el entorno de desarrollo Eclipse se puede hacer de dos maneras:

- (A) Importando directamente el proyecto comprimido zip. Se realizará una copia y el proyecto se generará en el workspace en el que se haya abierto Eclipse. Sobre Eclipse: crear un proyecto Java y sobre el importar: File=>Import => General =>Existing Projects into Workspace, pulsa la 2ª opción "Select archive file" y con el botón Browse... selecciona el proyecto comprimido.
- (B) Importando el proyecto descomprimido.
 - a. En este caso se crea el workspace del proyecto en la propia carpeta descomprimida y por tanto todos los cambios se realizan directamente en dicha carpeta. (1) Descomprimir el fichero en el disco duro. Se creará una nueva carpeta con el proyecto. (2) Importar el proyecto desde Eclipse: File=>Import => General => "Existing projects into workspace", pulsa la 1ª opción "Select root directory" y con el botón Browse... selecciona la carpeta descomprimida.
 - b. Si queremos que cree una copia y lo deje en el propio workspace en el que está eclipse, entonces además de seguir los pasos anteriores y antes de darle a Finish seleccionar la opción "Copy projects into workspace".

7. Documentación Javadoc del proyecto

Una vez importado el proyecto en el directorio doc aparecerá entre otros el fichero index.html, a partir del cual se puede navegar por toda la documentación HTML de la implementación asociada a los paquetes y clases.

Esta documentación se genera automáticamente en Eclipse:

Project => Generate Javadoc => Seleccionar el proyecto Bets =>

- Seleccionar directorio donde guardar los ficheros HTML con la documentación (crear la carpeta doc en el proyecto Bets).
- La primera vez es necesario indicar dónde está el ejecutable javadoc.exe (que estará en la carpeta lib del jdk. Copia la dirección o búscala con el browse.

En el código fuente se han escrito varios comentarios Javadoc como el siguiente (antes del método getEvents de la clase BLFacadeImplementation):

```
/**
 * This method invokes the data access to retrieve the events of a given date
 *
 * @param date in which events are retrieved
 * @return collection of events
 */
public Vector<Event> getEvents(Date date)...
```

Podéis comprobar en la documentación HTML qué es lo que se ha generado a partir de este comentario Javadoc. (en doc\businessLogic\BLFacadeImplementation.html).