



Introducción

Tras leer el apartado PLANTEAMIENTO DEL PROYECTO ABP de la GUÍA DEL ALUMNADO habréis visto que vuestro trabajo en Sinking Soft va a consistir en desarrollar una aplicación Cliente/Servidor para la gestión de apuestas deportivas. No es una aplicación Web, ya que hay otras asignaturas donde se trabaja en la especialidad Ingeniería del Software del plan de estudios del Grado en Ingeniería Informática.

En el apartado de METODOLOGÍA Y DESARROLLO DE LA ASIGNATURA de la misma guía se indica que vais a recibir el enunciado inicial del Proyecto, junto con varios artefactos (programa y documentación) correspondientes a una iteración ya desarrollada del proyecto (la iteración 0). Este es el propósito de este documento.

Antes de comenzar con el desarrollo de la iteración 1 (en grupos), se os propone una actividad individual que consiste en que extraigáis nuevos requisitos o funcionalidades de la aplicación. En concreto, este es el párrafo de la guía del estudiante donde se indica:

Pero antes de formar el grupo queremos comprobar que sabes en qué te estás metiendo. Tu primera actividad individual consistirá en una reflexión de los requisitos del proyecto, teniendo en cuenta no sólo las especificaciones y artefactos recibidos sino también algunos ejemplos reales como los comentados anteriormente. Si lo haces bien te meteremos en un grupo del proyecto y tendrás ocasión de hacer más méritos.

En este documento se presenta la información del proyecto desarrollado hasta el momento, durante la primera iteración del proyecto.

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 anterior 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.
- 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 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 de las apuestas (p.ej. 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.

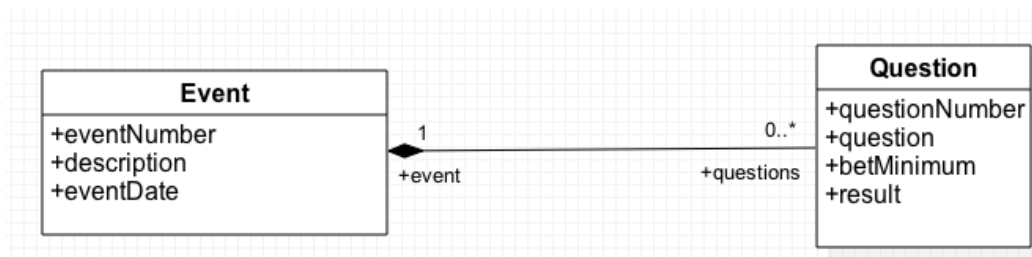
En todo momento el usuario puede consultar todas las apuestas que ha realizado, su estado (finalizada o no), y en caso de finalización su rendimiento.

NOTAS:

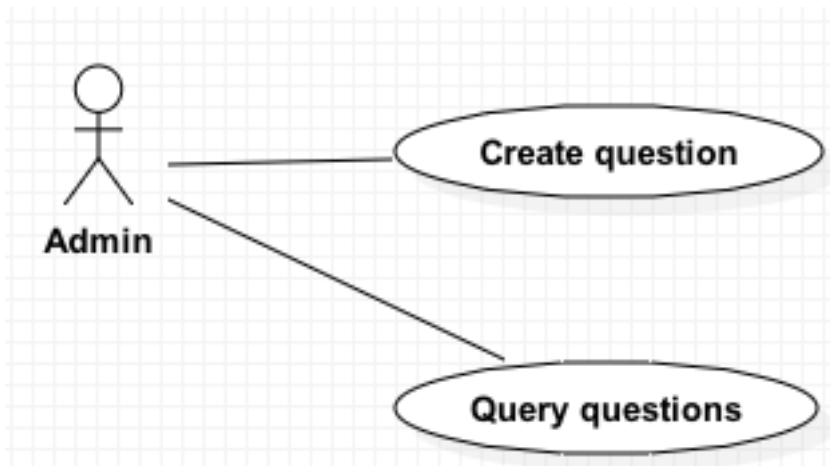
En las siguientes secciones se presentan los artefactos correspondientes al resultado de la iteración ya realizada: el modelo del dominio, el modelo de casos de uso, los de flujos de eventos de los 2 casos de uso desarrollados y el diagrama de secuencia para uno de esos casos de uso.

En esta primera iteración no se han realizado todos los casos de uso que corresponderían a la anterior descripción, 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

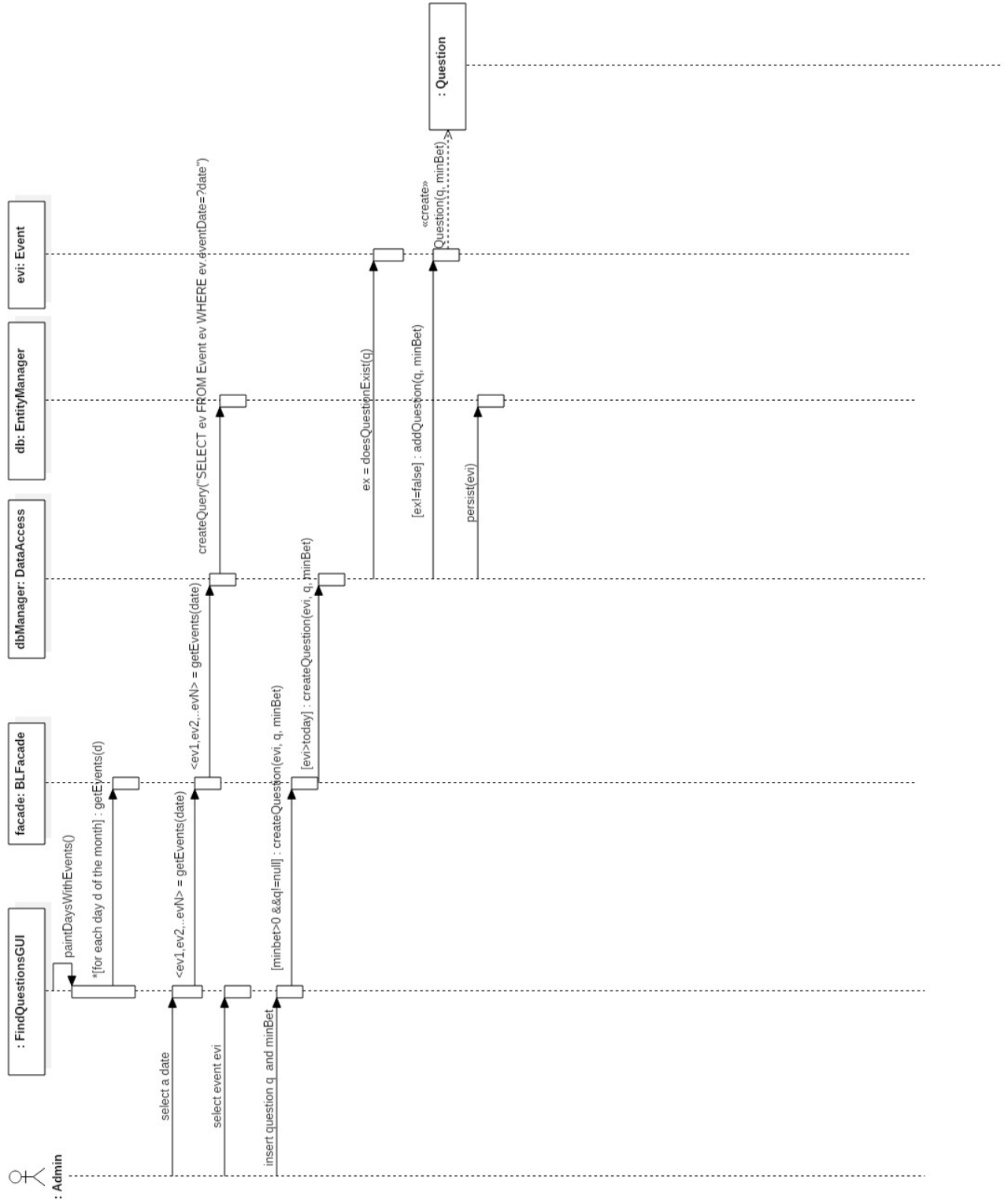
Basic Flow

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 casas rurales, resultado de la iteración 0 se encuentra disponible en el eGela de la asignatura (fichero Bets.zip)

Para usar el proyecto en el entorno de desarrollo Eclipse hay que hacer lo siguiente:

- 1) Descomprimir el fichero en el disco duro. Se creará una carpeta llamada Bets que contiene el código fuente Java, las librerías que trabajan con calendarios JCalendar y con bases de datos objectDB, y algunos ficheros que se encuentran en el directorio resources del proyecto. Se trata del 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).

- 2) Importar el proyecto desde Eclipse-

File=>Import => General => "Existing projects into workspace" y seleccionar el directorio Bets que se ha descomprimido

7. Documentación de las clases de implementación

Una vez descomprimido el fichero Bets.zip, en el directorio doc se encuentra el fichero index.html, a partir del cual se puede navegar por la documentación HTML asociada a los paquetes y clases de implementación.

Esta documentación se ha generado automáticamente en Eclipse:

Project => Generate Javadoc => Seleccionar el proyecto Bets =>
Seleccionar directorio donde guardar los ficheros HTML con la documentación

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`)

8. Creación del fichero .jar utilizado en el primer laboratorio de la asignatura y el fichero del despliegue inicial en un único nivel.

Se explica en un documento aparte que se encuentra en el eGela (cómo crear el jar).