F1 Fan. Modelo de la aplicación

Interfaces con el Usuario

Curso 09/10

Resumen

Este documento describe el desarrollo del componente *modelo* para la aplicación *F1 Fan*.

Diseño

El componente *modelo* de la aplicación está formado por una serie de clases que conforman el *core* del mismo, más otras clases e interfaces que permiten la iteración con otros componentes, en concreto con la interface de usuario.

Core

En el *core* encontramos cuatro clases: Equipo, Piloto, GP y Temporada, tal y como se muestra en el diagrama de clases de la figura 1. Cada temporada consta de varios GPs, en los cuales participan equipos y pilotos. Por cada GP tendremos dos relaciones que establecen el orden en que quedaron los pilotos en la parrilla de salida y al finalizar la carrera.

La relación más compleja en el core del modelo es la que existe entre pilotos y equipos, ya que ésta es de carácter temporal. En una fecha dada, cada piloto corre únicamente para un equipo, pero en otra fecha puede estar vinculado a un equipo distinto. Para implementar esta relación, en los métodos realcionados, p.e. `obtener_equipo`, es necesario indicar la fecha.

Las figuras 2 y 3 muestran en detalle las clases del core.

Interface

Los demás componentes del sistema deben interactuar con el modelo a través de los elementos que se muestran en la figura 4.

El acceso a las funcionalidades del modelo se realiza a partir de la clase Facade que provee métodos para el acceso a la BB.DD. y la creación de nuevas temporadas, GPs,

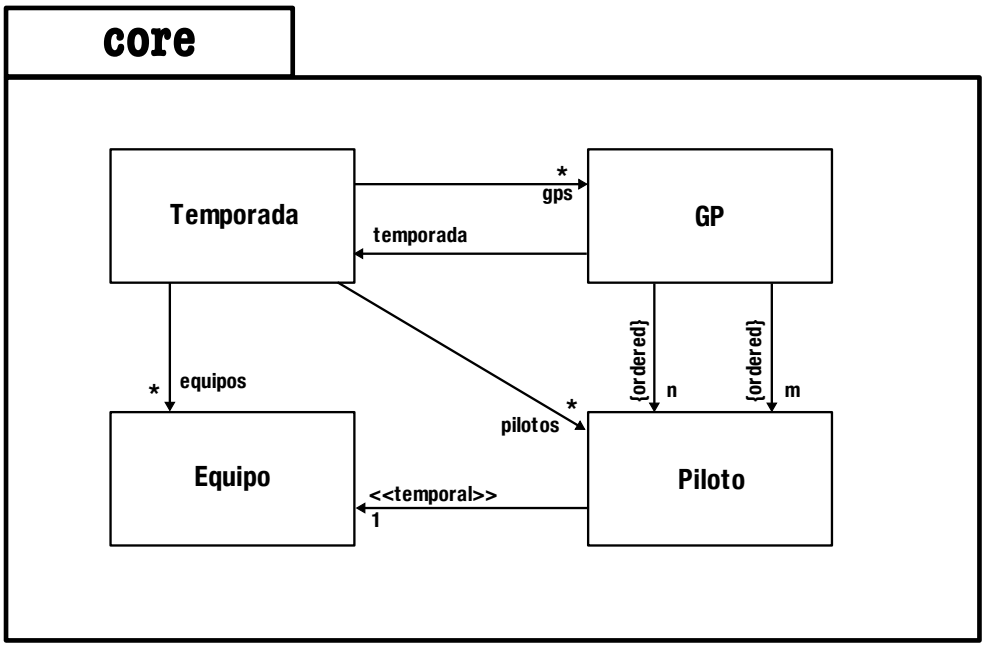

Figura 1: Diagrama de clases del core del modelo

pilotos y equipos. La fachada también mantiene una cache de temporadas para optimizar el acceso a la BB.DD.

Todos los objetos del core del modelo realizan la interface UI_VO, que define los métodos `obtener_datos` y `establecer_datos` y les permite comportarse de forma análoga a un *Data Transfer Object* o *Value Object*. La manipulación de estos objetos se debe hacer usando dichos métodos.

En caso de error, los objetos del modelo lanzarán la excepción correspondiente. A mayores de las excepciones propias del sistema, el modelo define otras más mostradas en el diagrama de la figura 4.

Por último, en caso de querer implementar una arquitectura Model-View-Controller, el modelo incorpora un mecanismo de *observadores*, tal y como se muestra en la figura 4.

La Base de Datos

La BB.DD. de la aplicación consiste en una colección de ficheros con formato YAML. Dichos ficheros se almacenan en un único directorio y cada fichero contiene un objeto

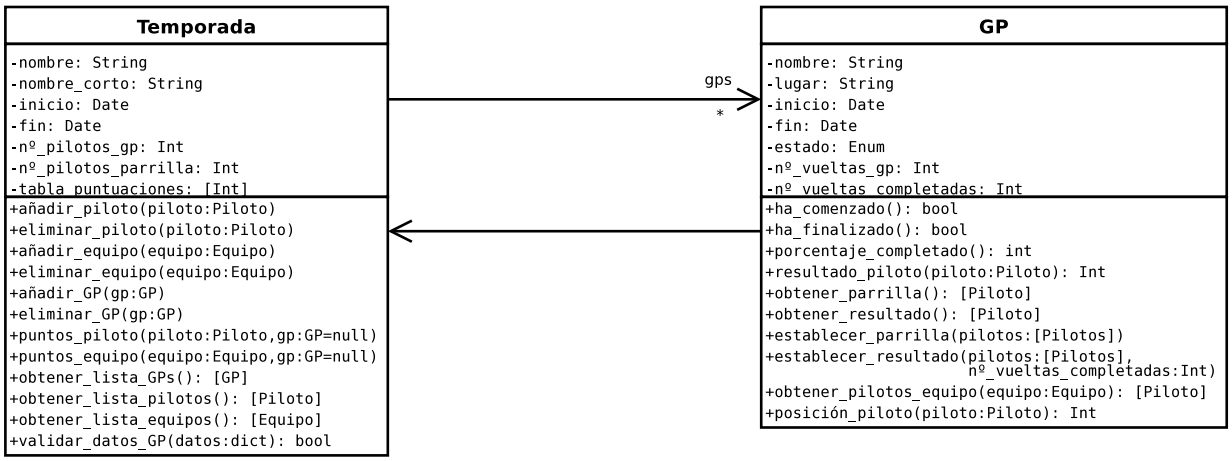

Figura 2: Diagrama detallado de las clases Temporada y GP

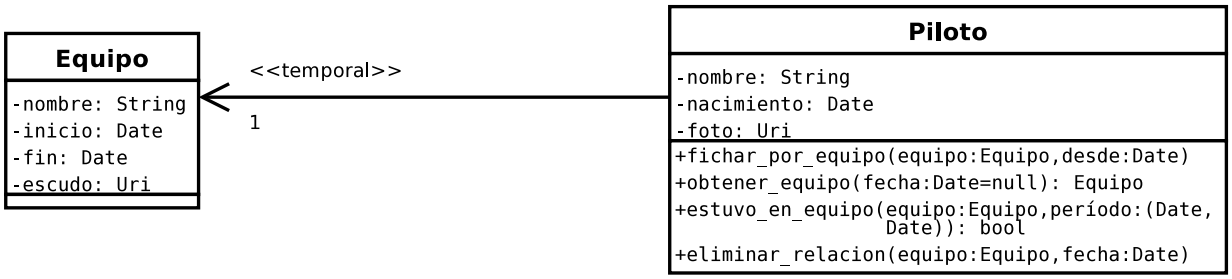

Figura 3: Diagrama detallado de las clases Piloto y Equipo

temporada y todos los objetos GP, Piloto y Equipo relacionados.

El sistema permite el uso de varias BB.DD. distintas, simplemente indicando el directorio en que se encuentra.

A la hora de almacenar o recuperar la información de la BB.DD. sólo es necesario indicar la temporada correspondiente. El sistema almacenará o recuperará la temporada y todos los GPs, pilotos y equipos relacionados.

El mantenimiento de la BB.DD. se puede realizar de forma sencilla manipulando los ficheros `.yaml`. Estos ficheros tienen formato de texto estructurado por lo que se pueden modificar directamente con cualquier editor de texto.

Elementos multimedia

El modelo no da soporte a la manipulación de elementos multimedia. Si se desea, por ejemplo, manejar desde la interface de usuario imágenes con los escudos de los equipos o las fotos de los pilotos, se debe implementar los mecanismos necesarios teniendo en cuenta que los objetos Piloto y Equipo del modelo únicamente registran un identificador (*URI*) del medio, pero no el medio en si mismo.

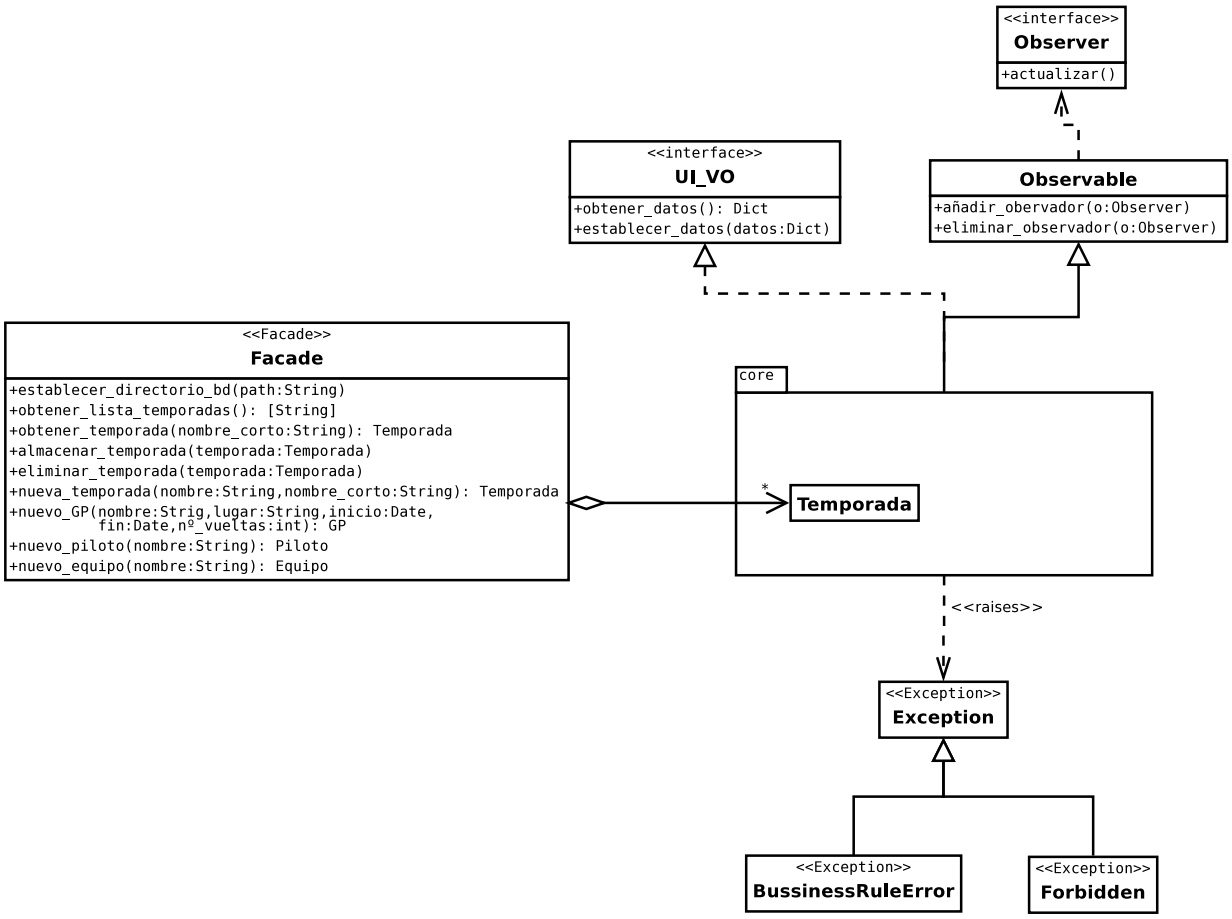

Figura 4: Elementos de interface con el modelo

BB.DD. para pruebas

Junto con el modelo se proporciona a modo de ejemplo un script para la creación de una BB.DD. útil para realizar pruebas sobre la aplicación. Dicho script también se puede utilizar como base para la creación de otras BB.DD. o para el estudio del funcionamiento del modelo.

Código fuente

El código fuente del modelo únicamente puede estar accesible a efectos de consulta y/o comprensión de su funcionamiento. Cualquier cambio realizado en el código fuente del modelo se considera un error de programación en la realización de la práctica.

En caso de existir bugs en el código del modelo, éstos se deben notificar en el foro de prácticas de la asignatura. Una vez corregidos los bugs, se liberará en la web de la asignatura una nueva versión del modelo.

API

La API detallada de los objetos del modelo, sus métodos, ... se puede consultar en la documentación que acompaña al código.

Consideraciones de implementación

A efectos de la realización de la práctica se considerarán errores de programación las siguientes acciones:

- Modificar el código fuente del modelo.
- Romper la encapsulación de los objetos del modelo accediendo directamente a sus atributos.