Trabajo Práctico
Organizador de Futbol 5

Entrega Nº 2

Materia: Diseño de Sistemas
Profesor: Nicolas Passerini
Ayudante: Gisela Dacuzi
Alumnos:

- **Acosta Naiara**
- **Brandoni Agustin**
- **Coiro Tomas**
- **Leder Brian**
- **Luis Ostiglia**

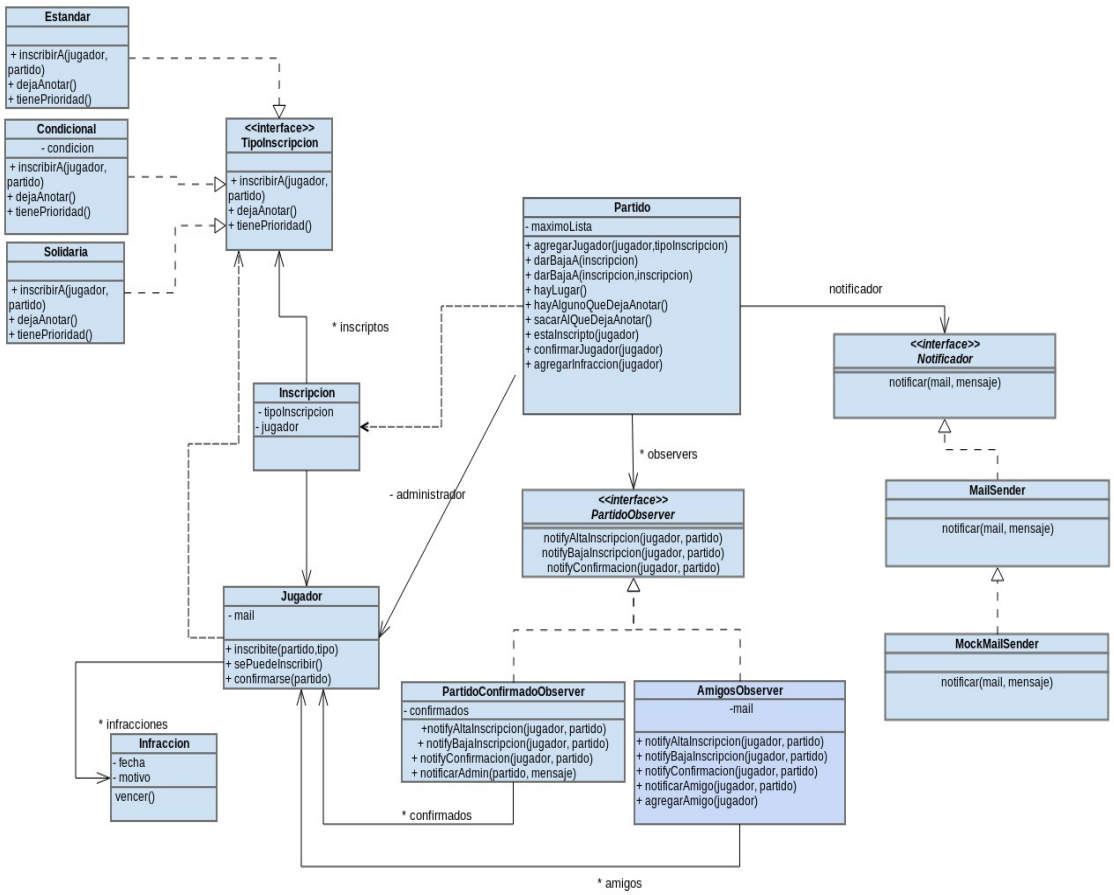
2014

Punto 1

Se plantean 2 soluciones de diseño:

Soluciones

Observer



Decorator

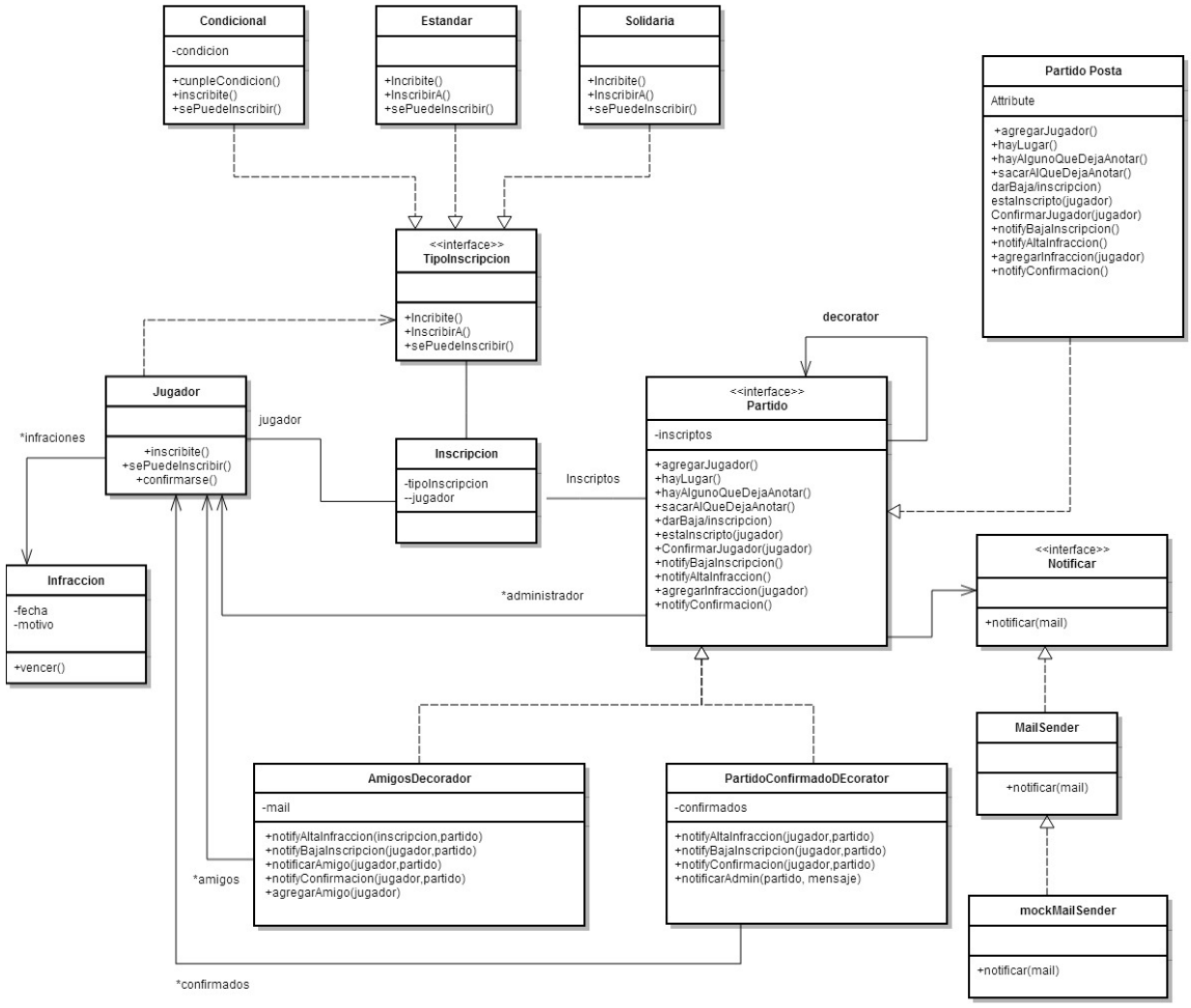
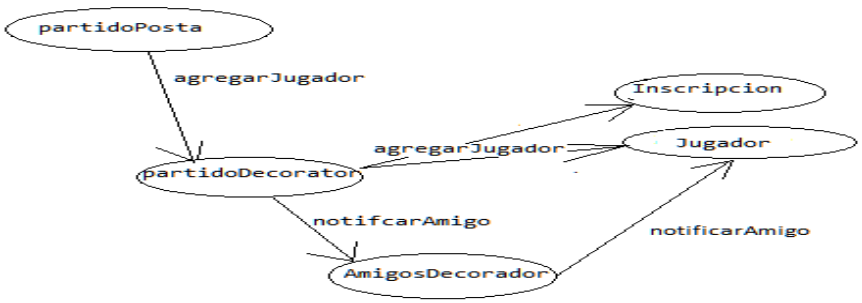
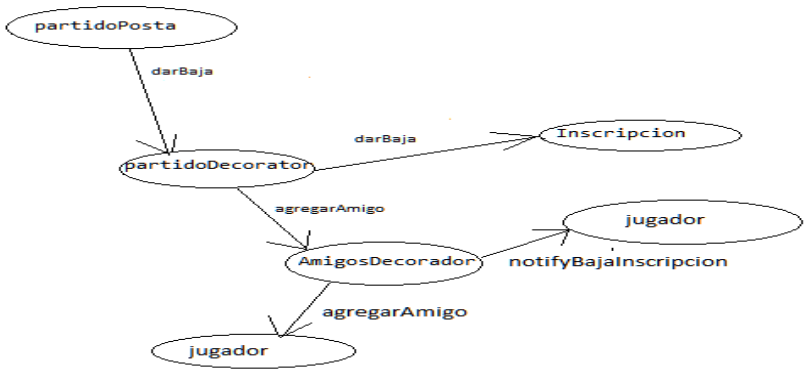


Diagrama de objetos Decorator

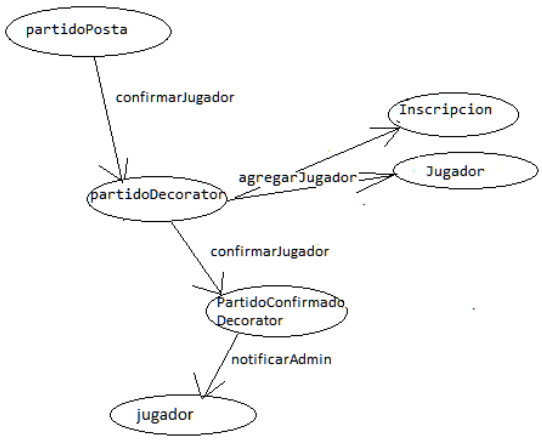
Cada vez que un jugador se inscribe al partido, partidoPosta manda el método a agregarJugador() al partidoDecorator para que este lo decore entonces llama al método agregarJugador con los parámetros inscripción y jugador. Tambien amigosDecorador llama al métodoo notificarAmigo que le **notifica a los amigos** del jugador.



Cuando un jugador se da de **baja** en un partido partidoPosta manda el método a darBaja() al partidoDecorator para que este haga todas las funcionalidad que tiene que hacer. Primero si el jugador indica quien lo reemplaza se le da de baja la inscripción de él y AmigosDecorador le notifica la baja al jugador y agrega al amigo que lo reemplazará con el método AgregarAmigo().



Cuando el jugador número 10 se **confirma** se agrega el jugador y el partidoDecorator confirma el jugador y delega al PartidoConfirmadoDecorato para que le avise al administrador, éste lo hace utilizando el método notificarAdmin().



Punto 3

La utilización del Mock facilitó la implementación de los Test ya que se requería modelar la tarea de notificar al administrador y a los amigos de los jugadores, por lo tanto mediante la utilización del Mock se "simulo" el envío de un mail (como medio de notificación), sin tener que implementar todas las cuestiones que realmente intervienen en un envío de mail, pudo correrse los test y poder evaluar que se notifiquen tanto los amigos, como administradores, y

en caso que mas adelante se requiera algún componente mas va a poder hacer uso del Mock para la notificación por medio de mail.

Punto 4

Cómo se incorpora cada funcionalidad:

- En el observer: se implementa el PartidoObserver (como interfaz con dos subclases). El aviso o notificación a los amigos se da en este Observer. También se incorpora la baja de la inscripción en la clase Partido y sus observadores son notificados debido al "cambio de estado" del objeto. La figura de la Infracción aparece como una clase aparte, la cual es conocida por el Jugador que además tiene incorporado el mensaje de "confirmarse()" para realizar la confirmación a un partido. El notificador se encarga del envío de notificaciones al Administrador.
- En el Decorator: se implementa el Decorator en el Partido, el cual es conocido por el PartidoPosta. En nuestro caso es para añadir funcionalidades de forma dinámica (añadirselo a un objeto y no a toda la clase). Aquí se denota la notificación de Alta o Baja de inscripción. El partido tiene una colección de inscriptos (inscripciones de ese partido que conocen a su jugador). Los jugadores poseen una colección de amigos a los cuales se les notifica por medio del Partido.

En cuanto a identidad

En el Decorator, el Partido pierde Identidad, ya que cuando se le manda un mensaje a un partido, no se sabe si tratamos con el de la clase Partido o el PartidoPosta. En el Observer, no hay "perdida" de Identidad.

En cuanto a cohesión

Tanto en el diseño Decorator como en el Observer podemos decir que el nivel de cohesion aumenta evaluando los puntos donde se implementaron estos patrones.
En ambos patrones se distribuye funcionalidad en clases más pequeñas pero con un objetivo mas claro y conciso, para el caso del Decorator se tiene como ejemplo de lo mencionado las clases PartidoConfirmadoDecortator y AmigosDecorator donde el objetivo se puede definir que cada una se encarga de “decorar” distintos aspectos, es decir agregar funcionalidad pero esos aspectos son claros: notificaciones.
Para el caso del Observer también se puede observar este punto en las clases observadoras, donde la tarea de notificar queda bien definida para estas clases observadoras y no en el partido como hubiese sido una solución pero que bajaba la cohesion de esta clase.