**Trabajo Práctico**

**Organizador de Futbol 5**

**Entrega Nº 1**

**Materia: Diseño de Sistemas**

**Profesor: Nicolas Passerini**

**Ayudante: Gisela Dacuzi**

**Alumnos:**

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

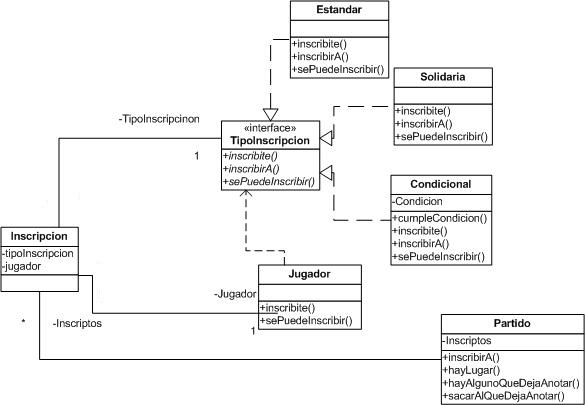
**2014**

**Punto 2**

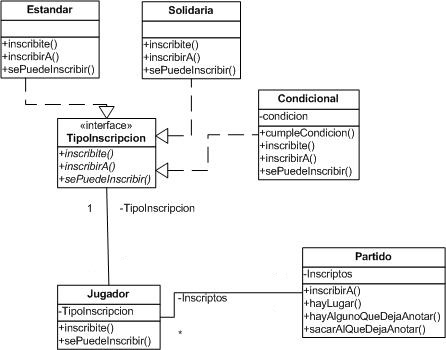
Se plantean 2 soluciones de diseño:

**Soluciones**

*Strategy Stateless*

**

*Strategy Statefull*



**Análisis de Solución y justificación de diseño**

Existe una diferencia sustancial entre la solución de Strategy Statefull y Stateless. El Stateless implementa a la clase "Inscripción", la cual consideramos necesaria para que se conozca la inscripción, ya que este conocimiento brinda la posibilidad de desplazar a un jugador solidario en caso que se inscriba uno estándar. Esto brinda la posibilidad de editar o cambiar a futuro las cosas, como por ejemplo, rechazar una inscripción y que un Jugador pueda tener Inscripciones rechazadas, añadirle mensajes a la Inscripción como podría incluirse un precio o distintas cosas que la caractericen, es decir nos brinda mayor flexibilidad que la solución Stafull

El Strategy Statefull implementa en el jugador el hecho de conocer a su Tipo de Inscripción. Ésta solución, nos parece un poco más simple estructuralmente, ya que incluye una clase menos y eso “colabora” a la simplicidad del modelo, sin embargo quita flexibilidad ya que si bien el atributo del jugador, que indica el modo de inscripción es modificable, define que un jugador solo se puede inscribir de una modalidad sola quintando la posibilidad de que un mismo jugador se inscriba a distintos partidos con modalidades diferentes. Además si se permite modificar ese atributo setter y a un partido se inscribe de una forma, luego se modifica el atributo, y se inscribe a otro partido, se generara una inconsistencia ya que al primer partido se inscribió de una forma y luego esa forma fue modificada.

Si bien en ambas soluciones se conoce la inscripción (De distintas formas), la Stateless es más flexible como se menciono anteriormente, ya que permite que el jugador se inscriba a un partido de una modalidad y a otro partido de otra modalidad así como también, permite un amplio abanico de modificaciones a futuro.

**Punto 4**

La solución que elegimos tiene 6 clases y 1 Intefaz (TipoInscripcion) que permite que 3 de las clases mencionadas puedan ser tratadas polimórficamente, las clases en cuestión son: Estandar, Solidaria y Condicional que representan los 3 tipos de inscripciones que se presentan en el enunciado. El flujo de ejecución sería el siguiente: un jugador desea inscribirse de un determinado modo por lo tanto se invoca jugador.Inscribite() donde se le pasa por parámetro la modalidad de inscripción y: Solidaria, Estandar, Condicional siendo estas objetos, además se le indica el partido al cual se tiene que inscribir. Las 3 clases de inscripciones, anotan al jugador de un partido de diferentes modos (Polimorfismo) según la especificación de requerimiento. Existen varios métodos que estas clases tienen en común, queremos destacar un método que si bien su implementación es trivial proporciona la flexibilidad para poder tener distintos tipos de inscripciones y conservar el encapsulamiento, este método es el dejaAnotar() donde cada clase de inscripciones define si deja anotar, es decir, si su prioridad es baja, como sucede con inscripción Estandar y Solidaria donde si la lista está llena pero tiene un jugador que se inscribió de modo solidario, este jugador es desplazado, y se inscribe al que lo realiza de modo estándar. Un detalle no menor es que el limite de la lista puede ser modificado, si bien se establece en 10 como se pide el enunciado, este valor no es definitivo.

**Punto 5**

Como se desliza en el análisis de soluciones propuestas, la opción elegida es la Strategy Stateless, tal como se mencionó es una solución mas flexible, permite que los jugadores se puedan anotar a distintos partidos con diferentes tipos de inscripción. De modo contrapuesto la solución Strateg Ambas soluciones se consideran mantenibles si se agrega por ejemplo un nuevo tipo de inscripción, sin embargo la solución elegida posee un factor más que aporta a la mantenibilidad y es que la Inscripción (clase) puede proporcionar un comportamiento muy amplio como pagarse, anularse y mas aún como generarse estadísticas en base a inscripciones siento todos estos puntos requerimientos que puedan solicitarse o necesitarse a futuro y que esta solución permite incorporar de modo sencillo