

Audry Morillo 07-41253

Michael Woo 09-10912

Cambios al diagrama de clases:

- Clase `ComitePrograma`:
 - o Se cambió `EvaluarArticulo(Texto:Articulo):Boolean` por `EvaluarArticulo(Puntaje: Integer, Texto: Articulo): Boolean`. Se agregó como parámetro de entrada `Puntaje` que determina el número de puntos que asigna el árbitro que ejecuta este método.
 - o Se agregó el método `AsignarPresidencia()`. Cambia de valor el atributo booleano `EsPresidente` de falso a verdadero. Mayor comodidad al momento de analizar el código en lugar de realizar un cambio de valor sin procedimiento.
- Clase `Articulo`:
 - o Se cambió el atributo `Arbitro` de tipo `ComitePrograma` por el de `Arbitros` de tipo lista de `ComitePrograma`. Se modificó su visibilidad a privada: Los jurados de los artículos únicamente pueden ser accedidos y modificados por métodos asociados a esta clase por privacidad.
 - o Se agregó el atributo de tipo flotante `Promedio_Puntuacion` que describe el promedio que posee el artículo respecto a los puntajes dados por 1 o más árbitros.
 - o Se agregaron los métodos de clase:
 - `SetArbitraje`: Recibe una dupla (`Arbitro: ComitePrograma`, `Puntaje: Integer`) donde `Arbitro` es el miembro del comité programa que sirve de jurado al artículo y `Puntaje` la puntuación asociada a la evaluación. El método realiza una evaluación del artículo.
 - `GetArbitraje`: Retorna la lista de evaluaciones hechas al artículo en forma de dupla (`evaluador`, `puntaje`).
 - `Promediar`: De acuerdo con las evaluaciones hechas al artículo calcula el promedio de puntuación del mismo.

- GetPromedio: Retorna el promedio de puntuación del artículo.
- Evaluar: Inicialmente calcula el promedio de puntuación del artículo, luego verifica que su promedio sea mayor o igual a 3.00 y que haya sido evaluado por un número mayor o igual a 3 jurados; de ser éste el caso, se agrega el artículo a la lista de artículos posibles a aceptar.

La realización de la tarea nos llevo un aproximado de 3 días, y la cantidad de rotaciones fueron 9 veces.

Al momento de utilizar eclipse se nos presentaron algunos contratiempo, ya que la variedad de versiones complico al momento de querer instalar el plugin de pyDev, eso ocasiono retraso de tiempo al no contar con la herramienta, realmente fueron muchos los intentos por lograr instalar la herramienta con resultado negativo, y decidimos hacer uso de otra herramienta IDE llamada Aptana Studios que esta basada en Eclipse por lo que genera un ambiente de trabajo muy similar al que hubiésemos experimentado con la otra herramienta, entre las ventajas es la gran cantidad de facilidades como auto completado, indentacion, el resaltado, y la integración de la terminal entre otras.

Los casos de prueba fueron diseñados a medida que se presentaron las necesidades, con el método de escribir prueba y luego escribir el código, aunque luego se vio en la necesidad de reescribir el código de pruebas pues existía mucha replicación de código.

Con respecto a la refactorizacion, la mayor refactorizacion fue en el código de las pruebas unitarias, que existía mucho código replicado y se llevo a la menor cantidad de código posible.