

## Tarea 2

En los modelos ágiles de desarrollo se acostumbra hacer entregas pequeñas frecuentes con funcionalidad útil para el cliente.

En quince días (viernes 29 de noviembre) cada pareja de programadores debe entregar el prototipo de un sistema orientado por objetos para CLEI escrito en Python capaz de:

- Crear artículos;
- Crear los miembros del comité de programas y su presidente;
- Asociar las puntuaciones de los árbitros a los artículos y calcular los promedios correspondientes;
- Para considerarse aceptable un artículo debe tener al menos dos evaluaciones y su promedio no puede ser menor de 3,00.
- Dado el número de artículos ‘n’ que pueden escogerse, el sistema debe crear dos listas:
  - Una lista de ACEPTADOS de a lo sumo los ‘n’ artículos aceptables con los promedios más altos. Ningún artículo de esta lista puede tener el mismo promedio que un artículo aceptable que no esté en la lista.
  - Una lista de EMPATADOS correspondientes a aquellos artículos aceptables empatados en el mayor promedio fuera de ACEPTADOS.
    - Por ejemplo si se aceptaran 40 artículos y 30 de ellos tienen promedios mayores o iguales a 3,50 y 20 de ellos están empatados en el promedio más alto después de 3,50, digamos 3,33, ACEPTADOS lista los primeros 30 artículos y EMPATADOS los otros 20.

### FUNCIONALIDADES PARA ENTREGAS POSTERIORES

- El Presidente del Comité de Programa escoge, de entre los EMPATADOS los artículos a aceptarse.
- Se registra el estado final de la selección: aceptado, aceptadoEspecial, rechazadoFaltaCupo, rechazadoPorPromedio.
- Histogramas de los artículos aceptados: por autor, por tópicos de artículos aceptados, por instituciones de afiliación y por país.

Nota: La interacción con el sistema debe poder realizarse por consola corriendo un archivo de nombre “main.py” (p.ej.: `$> python main.py`).

Cada pareja debe entregar, adicional al prototipo:

- El diagrama de clases actualizados, justificando cualquier cambio hecho respecto al modelo de la Tarea 1.
- Un reporte del tiempo total que tardó la tarea y el número de veces que la pareja

rotó.

- Un breve reporte destacando las bondades y las dificultades al usar Eclipse (si no se usó, justificar muy bien la razón).
- Los casos de prueba usadas en PyUnit, destacando el orden en que fueron diseñadas. ¿Fallaron las pruebas de regresión?
- Un breve reporte sobre el uso de refactorización (el tema de refactorización será cubierto en la semana 4)