

ComfyChair

Trabajo Práctico - TyH 2024

Se quiere desarrollar un sistema llamado ComfyChair para organizar conferencias científicas, particularmente el proceso de envío y revisión de artículos.

El sistema debe permitir crear **conferencias**, cada una con sus **chairs** (organizadores) y su comité de programa (grupo de **revisores**). También estarán los **autores** de los artículos. Todos ellos (chairs, revisores, autores) son en realidad usuarios registrados en ComfyChair con la misma información, solo que pueden ocupar diferentes roles; es decir, un autor también puede ser revisor o chair. De cada usuario se sabe su nombre completo, afiliación (por ejemplo, UNLP, o alguna empresa), email y contraseña.

La conferencia está organizada en sesiones (también conocido como track) en donde se aborda un tema en particular (por ejemplo, en una conferencia de Informática podríamos tener Ingeniería de Software, IA, o BigData). A cada sesión de la conferencia se envían artículos de dos tipos: regulares o posters. Todos tienen un título y un archivo adjunto (del cual se guarda una URL), pero los regulares tienen además un *abstract* (resumen), y los posters un segundo archivo adjunto (de nuevo, una URL) con los fuentes. Para todos los artículos se conoce al grupo de autores, y en particular cuál es el autor encargado de las notificaciones, todos usuarios registrados en ComfyChair.

Toda sesión de la conferencia pasa por un proceso:

- **Recepción:** Durante este estado, se pueden enviar artículos al evento. Luego del envío por parte de los autores se revisa su formato:
 - para los **artículos regulares**, se valida que tengan un abstract de menos de 300 palabras y el título y autores estén cargados (al menos un autor).
 - los **posters** se validan de igual manera pero no se considera el abstract.si no se cumple con estos requisitos, se rechaza - es decir que no pasa a la siguiente etapa. La etapa de recepción de artículos finaliza automáticamente en una fecha determinada (deadline) que puede ser diferente para cada sesión. Los envíos pueden modificarse hasta entonces.
- **Bidding:** los artículos en revisión pasan por un proceso de *bidding*. En este punto, los revisores expresan si están o no interesados en revisarlo. Además de "interesado"/"no interesado" existe también un "quizás". El proceso completo de bidding y asignación de artículos será explicado aparte. Durante esta instancia, no se aceptan más artículos.
- **Asignación y Revisión:** una vez finalizado el proceso de *bidding*, los artículos se asignan a los revisores. Cada revisor emite una revisión con una recomendación que es una calificación entre -3 y +3 (incluyendo el 0). Durante esta instancia, no se aceptan más artículos.

- **Selección:** finalizadas las revisiones, se pasa al proceso de selección: los artículos con mejores recomendaciones pasan a estar aceptados, y el resto se rechaza. El número máximo de artículos a aceptar está definido en cada sesión. El número final aceptado puede ser menor. Más adelante se describen detalles adicionales sobre este proceso. Durante esta instancia, no se aceptan más artículos.

Paper Bidding

Durante el proceso de paper bidding mencionado anteriormente se reciben “bids” que expresan el interés de los revisores. Éstos no tienen la obligación de expresar interés, pero en ese caso se les asignan los artículos aleatoriamente, o según la necesidad de los chairs. Los revisores pueden cambiar de opinión respecto a un *bid*, es decir, pueden pasar de “interesado” a “quizás” o cualquier otro tipo de interés para un mismo paper. Al cerrarse el proceso de bidding, se asignan los revisores a los artículos, y siempre debe haber 3 revisores por artículo. Esto implica que se necesita un número total de revisiones igual al triple del número de artículos - ej. si se enviaron 10 artículos, se necesitan 30 revisiones. Esto impacta en el número de revisiones que se le pide a cada revisor: siguiendo con el ejemplo, si hay 10 artículos (30 revisiones) y 5 revisores, cada uno deberá revisar 6 artículos ($5 \text{ revisores} \times 6 = 30 \text{ revisiones}$). Si el número no es redondo, puede ser que algunos revisores reciban más artículos que otros. Siguiendo con el ejemplo, si en lugar de 5 revisores hubiera 7, 5 de ellos revisarán 4 artículos (20 revisiones) y 2 de ellos 5 (10 revisiones).

Asignación y Revisión

La asignación se realiza de la siguiente manera: para cada artículo, se buscan revisores que no hayan llegado al límite de revisiones. De éstos se asignan primero los que se marcaron “interesado”, si no se llega a 3, se buscan entre los “quizás”. Si aún no se llegó a los 3, se busca entre los que no indicaron interés alguno, y si no hay se recurre finalmente a los “no interesado”. Luego de la asignación, los revisores ingresan sus revisiones para cada artículo. Una revisión tiene un texto y un puntaje que va de -3 a 3 incluyendo el 0. Los artículos no deben admitir más de 3 revisiones.

Selección

Una vez cargadas todas las revisiones, se seleccionan los artículos considerando la calificación de los revisores. Hay dos maneras de hacer esto:

- Corte fijo: la conferencia acepta solo un porcentaje de los envíos, y se aceptan en orden decreciente de puntaje hasta completar ese porcentaje.
- Mejores: la conferencia acepta todos los artículos que tienen puntaje superior a un número a definir.

No se descarta en el futuro contar con otras formas de selección.

Cada sesión puede definir su manera de seleccionar, o incluso cambiarla (por ejemplo, si al utilizar el corte fijo quedan muchos artículos de bajo puntaje).

Extensión (punto extra)

Una sesión puede ser **regular**, **workshop**, o de **posters**. La diferencia radica en la forma de tratar los artículos. Las sesiones regulares admiten solamente artículos regulares, los

workshops admiten regulares y posters, y las sesiones de posters solo reciben posters. En el caso de las sesiones de workshop, hay que considerar que el criterio de selección se define para cada tipo de paper por separado.

Aclaraciones

- Evitar el uso de expresiones lambda o funciones anónimas, salvo para la API de colecciones.
- Si se utilizan patrones de diseño para la solución, documentarlos.

Objetivos

- Realizar un **diagrama de clases** con la solución.
- Un documento que indique cualquier tipo de toma de decisiones sobre el enunciado.
- **Implementar la solución en NodeJS** utilizando el paradigma de orientación a objetos.
- Implementar **tests de unidad** para probar la solución con una cobertura mínima del 80% del código.
- Utilizar GitHub y se evaluará la contribución de cada miembro. Compartir los proyectos a:
 - matiasurbieta