**Lógica del Sistema**

Voy a tener una clase denomida ComfyChair, que es la encargada de crear las Conferencias y los Usuarios.

La clase Conferencias es la encargada de crear las Sesiones y llevar un listado de las sesiones.

La clase Sesiones es la encargada de recibir los artículos enviados por los autores.

La Sesión se crea en el estado ‘Recepción’.

El Autor crea un artículo. El artículo se crea en un estado ‘enProceso’, quiere decir que no está aprobado, ni rechazado aún.

La Sesión valida que el artículo sea apto de recibirlo.

Validaciones:

\*Que no se haya definido un Autor para recibir las notificaciones.

\*Si la fecha actual es mayor al deadline de la recepción, se rechaza.

\*Si la Sesión es Regular y el tipo de artículo no es Regular, se rechaza.

\*Si la Sesión es Posters y el tipo de artículo no es Posters, se rechaza.

\*Si la Sesión es Workshop y el tipo de artículo no es ni Posters, ni Regular, se rechaza.

\*Si el tipo de artículo es Regular y no tiene Abstract o es superior a 300 caracteres o si no tiene Título o si no tiene Archivo adjunto, se rechaza.

\*Si el tipo de artículo es Posters y no tiene Título o si no tiene Archivo adjunto, o si no tiene los Fuentes, se rechaza.

\*Si no se definen los autores del artículo, se rechaza.

Si pasa todas las validaciones se crea un archivo por cada sesión y el nombre del archivo es el nombre del tema de la sesión .json donde almaceno todos los datos del artículo. Como no trabajo con js, ni con apis encontré esta manera de almacenar los datos de los artículos. Si bien a los objetos no se les define un id, a modo de que se me haga más sencillo de buscarlo, rescatarlo, realizar bidds; al artículo le asigné un atributo id.

Los artículos que no pasan las validaciones, no se graban en estos archivos.

Cuando un artículo no pasa las validaciones, la sesión manda un mensaje al autor designado para recibir las notificaciones, avisando porque no pasó la revisión.

Una vez pasada el deadline de recepción, la Sesión cambia su estado a ‘Bidding’.

Definí una clase abstracta denominada Usuarios.

La clase Autores hereda de Usuarios y es ella quien crea los artículos, envía a la sesión el mismo y recibe las notificaciones enviadas por la sesión.

Como el artículo que no pasó las validaciones nunca se llegó a grabar en el archivo, el autor deberá volver a pasar por el proceso de crear artículo.

La Sesión mantiene un listado de todos los artículos que pasaron las validaciones.

Los revisores pueden expresar su interés sobre un artículo. Pueden modificar también su interés.

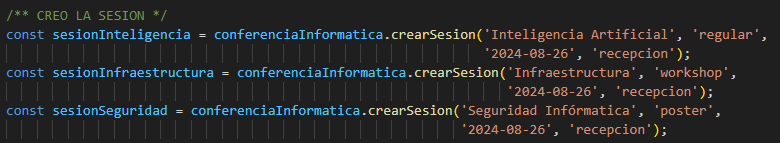
El Chairs tiene acceso al listado de Bidds por artículos.

El encargado de cambiar el estado de la Sesión es el Chairs.

**Que tener en cuenta para testear**

**Playground**

En la etapa de recepción verificar que la fecha actual no sea superior a la fecha del deadline de recepción.



Para que no genere muchas leyendas en la terminal de pruebas, sugiero, en la etapa de recepción de artículos, comentar todo lo que tenga que ver con las etapas posteriores.

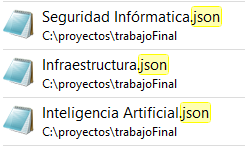


Esta línea debe descomentarse cuando se va a proceder a la etapa de asignación.

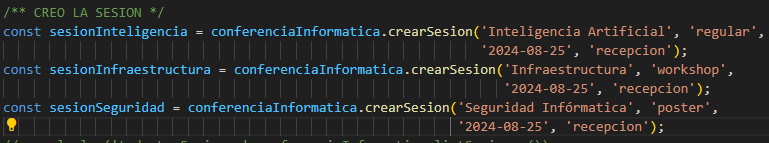
/\*\* EL CHAIR CAMBIA EL ESTADO DE LA SESION DE BIDDING A ASIGNACION \*/

//juan.cambiarEstadoSesion(sesionInteligencia,'asignacion');

Todos los artículos que se envían y pasan las verificaciones, se graban en estos archivos.



Para pasar al bidding, se debe tener en cuenta que ya haya pasado el deadline de recepción (para hacer las pruebas tuve que cambiar la fecha del deadline en la creación de la sesión.



Para que no genere muchas leyendas en la terminal de pruebas, sugiero, en la etapa de bidding, sólo tener descomentado lo que está marcado como etapa de bidding.

**Test**

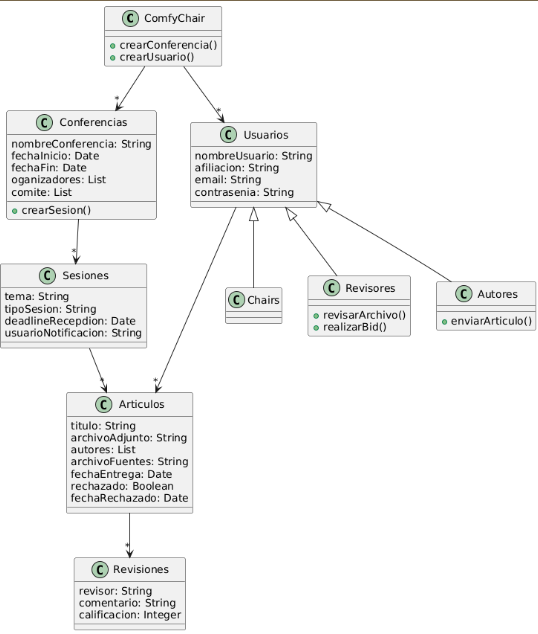
En los siguientes test, crear la sesión con un deadline superior a la fecha actual para que pasen los 8 primeros tests.

test("La sesión recibe el artículo creado"

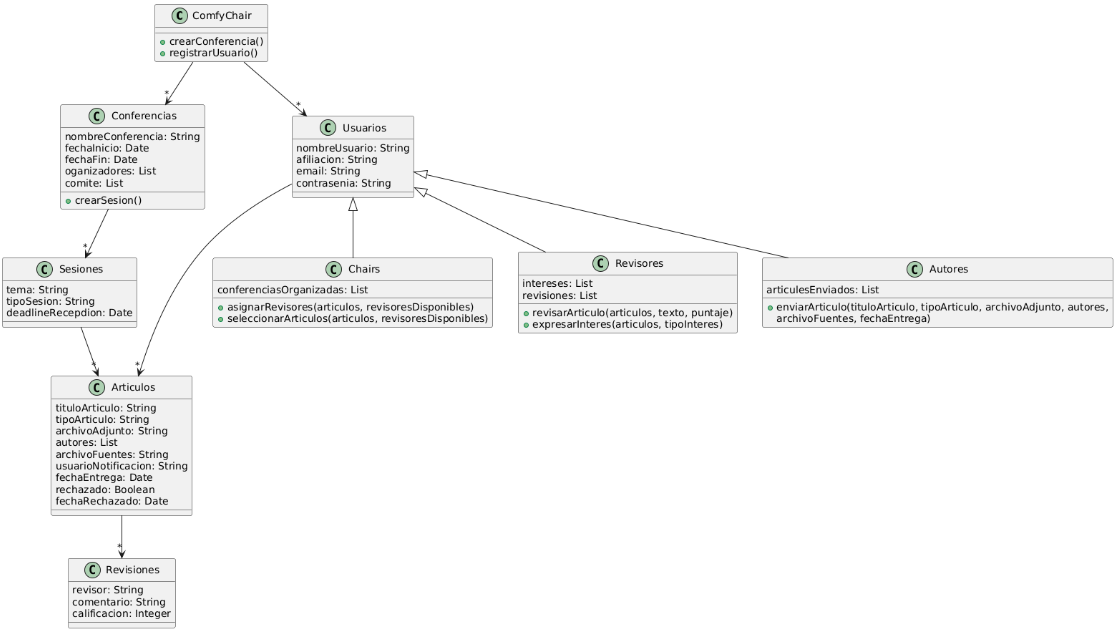
test("La sesión rechaza el artículo y envía un mensaje al autor que recibe las notificaciones"

Para probar el siguiente test definir una fecha de deadline anterior a la actual.

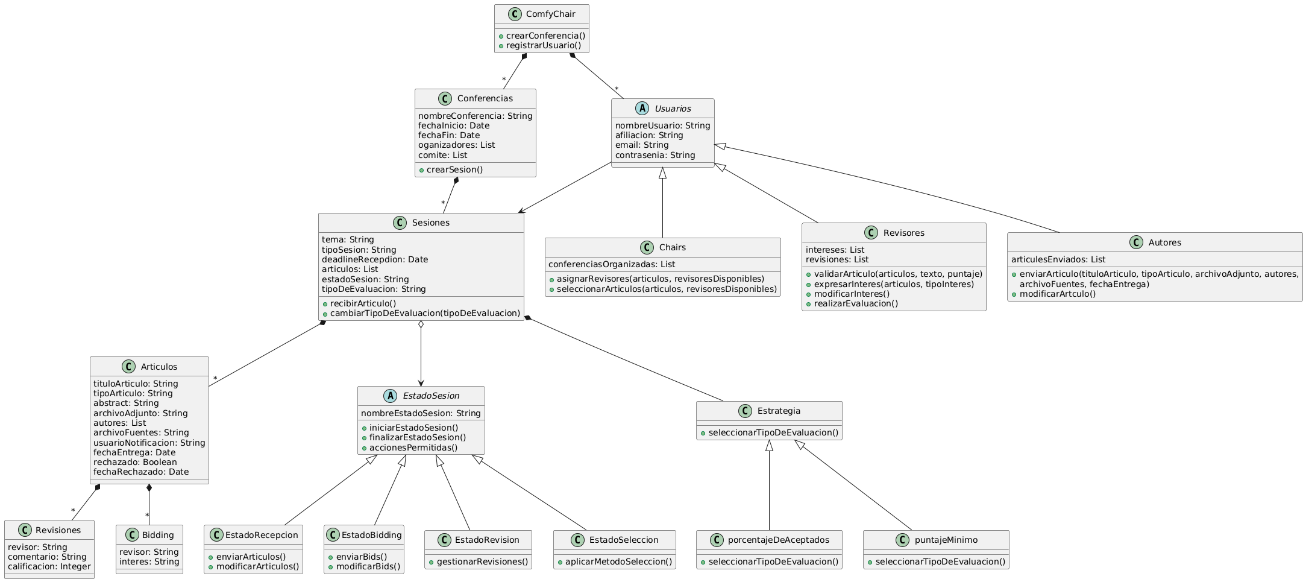
test("La fecha actual es superior al deadline, la Sesión pasa al estado de Bidding"

1era Aproximación a lo q sería el diagrama UML de ComfyChair

2da Aproximación



3era Aproximación



4ta Aproximación

