Vamos a hacer una integración bidireccional con **Trello**. En trello tenemos tableros y cada tablero tiene listas y cada lista tiene tarjetas.

En nuestro caso, tendremos dos colecciones, listas y tarjetas. Las listas contendrán tarjetas y una tarjeta solo podrá pertenecer a una lista.

Cada lista tendrá dos atributos (aparte del identificador de mongo):

- `trelloListId`: id que trello le da a la lista
- `name`: nombre

Cada tarjeta tendrá dos atributos (aparte del identificador de mongo):

- `trelloCardId`: id que trello le da a la tarjeta
- `name`: nombre

Se pide:

A. CRUD PARA GESTIONAR LISTAS Y TARJETAS

Crear rutas para gestionar las listas y tarjetas. En este primer paso, solo usar el atributo `name` (aún no estamos trabajando con **Trello**. Tendremos los siguientes endpoints:

```
Agregar una nueva lista
POST /lists

Modifica el nombre de una lista
PUT /lists/:idList

Obtener todas las listas
GET /lists

Obtener una lista en concreto
GET /lists/:idList

Crear una tarjeta en una lista
POST /lists/:idList/cards

Modifica el nombre de una tarjeta
PUT /lists/:idList/cards/:idCard

Mueve una tarjeta a otra lista
PUT /lists/:idList/cards/:idCard/to/:idDestionationList

Eliminar una tarea
DELETE /lists/:idList/cards/:idCard

Eliminar una lista y todas las tarjetas que contenga
DELETE /lists/:idList

Obtener todas las listas y todas las tarjetas jerarquicamente
GET /board
```

B. SINCRONIZAR TRELLO

Atacar la API de **Trello** para enviar los cambios a un tablero de trello en las escrituras anteriores. Cuando se creen elementos, los ids generados en trello se actualizaran en nuestras colecciones de modo que posteriormente puedan ser localizados y eliminados.

En el caso de las listas, trello no permite eliminarlas. La archivaremos

C. TWO WAY SYNC

Leer los webhooks de **trello** para actualizar las listas/tarjetas en nuestra colección:

- Una tarjeta cambia de lista
- Cambia nombre de una tarjeta
- Eliminar tarjeta
- Cambia nombre de una lista
- Archivar lista

Qué se pide

1. Usar Node + Express + Mongo + Typescript
2. Los apartados A y B son obligatorios
3. Tener planteado / avanzado el apartado C

Qué se valora

1. Código limpio
2. Programar en ingles
3. Posibles mejoras
4. Simplicidad de la solución
5. El apartado C