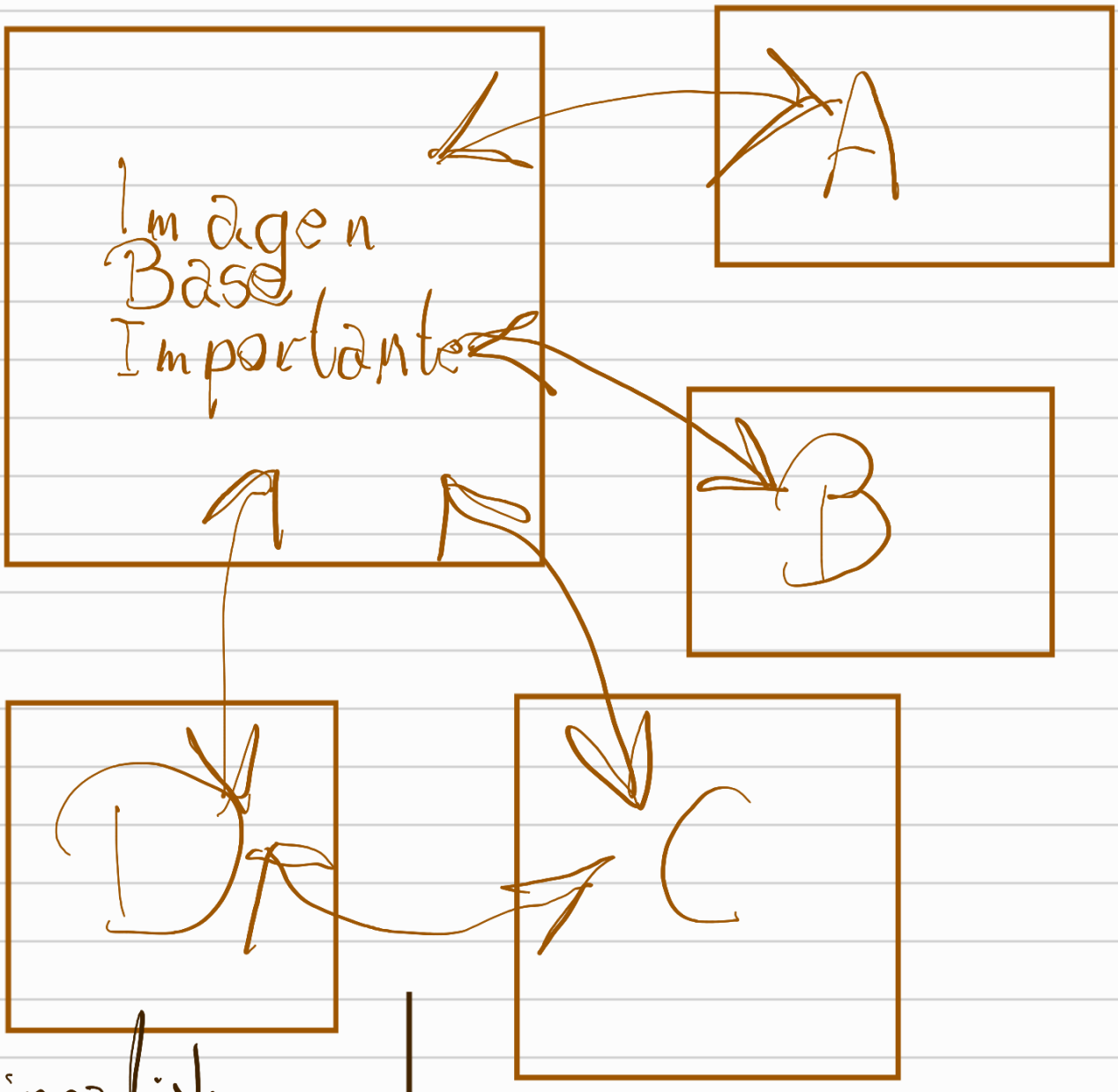
mi problema está siendo que con el transformador logro hacer bien el cambio de imagen pero estando más de una conversión contada para cualquier cosa.



Linea 1: *el cuerpo = imagen base*
2 *llamada a nueva imagen*
3 f irAl *imagen*){ *se remueve, y cambia la imagen* }

Habiendo formado una estructura lineal de esta conexión, necesito establecer parentezco con cualquier imagen que se presente, incluso la básica.

la fórmula para lograr esto de la derecha tiene que venir del lado de un parámetro

establecido la llamada, con todo lujo de detalles así:

que se declare en el mismo onclick

una vez viendo esto, podemos empezar a plantear una estrategia.

```
... <i><a id="*numerico*" onclick=
irA(*nueva imagen*)>Página</a></i>
```

```
function irA (*cambio de imagen*) {
    document.body.removeChild(cuerpo)
//  se limpia la imagen
    *se construye la imagen nueva*
```

Si con esto hacemos el cambio de imagen: ¿no sería hacer lo mismo cada vez que queramos, preservando todas las funciones en una sola y llamar a una variable para que se regenere?...

↳ llamada

```
function dirección (llamada) {
    let valor = llamada;
    return valor}

let punto = dirección (llamada);
```

En este punto estoy fallando. No puedo llamar a una función por su

Viendo que esto fallaba, por la obvia razón marcada en negrita el parámetro se me ocurrió la brillante idea de jugar con el almacenamiento local de

parámetro, si no se si galera cuando es este.

esta manera...

↳ llamada.

```
f pendiente (llamada) {
* guardamos en localStorage() llamada.*

f direccion() {
* tomamos el dato, y limpiamos el almacenamiento*
* exportamos el dato, para convocarlo en ir A, cada vez* .}
```