

JavaScript Funcional

Jose Antonio Castro // Tony

Benjamín Vicente // BV

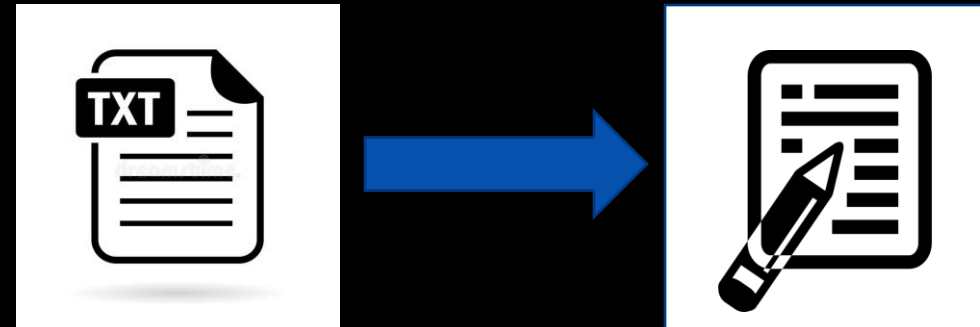
José Madriaza // Cacú



Tareas a realizar

Transformaciones de texto:

1. Indentar todas las frases
2. Cambiar saltos de líneas de los párrafos
3. Limitar los anchos
4. Indentar párrafos (sangría)
- 5&6. Filtrar párrafos entre N y M
7. Transformar frases a párrafos
8. Filtrar primeras frases de cada párrafo



Funciones para cada tarea

Transformaciones de texto:

1. Indentar todas las frases
2. **Cambiar saltos de líneas de los párrafos**
3. Limitar los anchos
4. Indentar párrafos (sangría)
- 5&6. Filtrar párrafos entre N y M
7. **Transformar frases a párrafos**
8. Filtrar primeras frases de cada párrafo

```
// Pregunta 2
/** Cada párrafo debe estar separado por ••• líneas (después de un punto aparte) */
export const lineBreaks = transformFn(
  { n: { name: "Line breaks", default: 2 } }, (text, { n }) =>
    text.replaceAll(/(\n+)/g, "\n".repeat(n))
);

// Pregunta 7
/** Cada frase debe aparecer en párrafo aparte */
export const convertToParagraphs = transformFn(
  {}, (text) =>
    text
      .split("\n")
      .map((line) =>
        line
          .split(".")
          .map((p) => p.trimStart())
          .join(".\n\n")
          .trimEnd()
        )
      .join("\n")
    );
```

Functional JS

Set options (2)

Reset

El ozono es una molécula que se encuentra en la atmósfera terrestre y que tiene una gran importancia para la vida en nuestro planeta. En particular, la capa de ozono es una región de la atmósfera que contiene una alta concentración de ozono y que se encuentra a una altitud de entre 10 y 50 kilómetros sobre la superficie de la Tierra.

La capa de ozono tiene una función muy importante, ya que actúa como un escudo protector que nos protege de la radiación ultravioleta (UV) emitida por el sol. La radiación UV es dañina para la salud de los seres vivos, ya que puede causar quemaduras en la piel, daños en los ojos y, a largo plazo, aumentar el riesgo de desarrollar cáncer de piel.

Sin embargo, en las últimas décadas hemos visto una disminución significativa en la cantidad de ozono en la atmósfera. Esta disminución se debe principalmente a la emisión de sustancias químicas como los clorofluorocarbonos (CFC), que se utilizan en la industria y en los sistemas de refrigeración y aire acondicionado.

La disminución en la capa de ozono ha sido un problema de gran preocupación para la comunidad científica y para la sociedad en general, ya que podría tener graves consecuencias para la salud y el medio ambiente. Por esta razón, se han tomado medidas para reducir la emisión de sustancias que dañan la capa de ozono, como la eliminación gradual de los CFC.

El ozono es una molécula que se encuentra en la atmósfera terrestre y que tiene una gran importancia para la vida en nuestro planeta.

En particular, la capa de ozono es una región de la atmósfera que contiene una alta concentración de ozono y que se encuentra a una altitud de entre 10 y 50 kilómetros sobre la superficie de la Tierra.

La capa de ozono tiene una función muy importante, ya que actúa como un escudo protector que nos protege de la radiación ultravioleta (UV) emitida por el sol.

La radiación UV es dañina para la salud de los seres vivos, ya que puede causar quemaduras en la piel, daños en los ojos y, a largo plazo, aumentar el riesgo de desarrollar cáncer de piel.

Sin embargo, en las últimas décadas hemos visto una disminución significativa en la cantidad de ozono en la atmósfera.

Nuestra implementación

```
import { array as arr, function as fn } from "fp-ts";

export type Params = Record<string, { default: number; name: string }>;
type FilterFn<P> = (text: string, params: { [key in keyof P]: number }) => string;
const transformFn = <P extends Params, Fn extends FilterFn<P>>(p: P, fn: Fn) => ({ params: p, fn });
```

```
// Pregunta 2
/** Cada párrafo debe estar separado por •n• líneas (después de un punto aparte) */
export const lineBreaks = transformFn(
  { n: { name: "Line breaks", default: 2 } }, (text, { n }) =>
    text.replaceAll(/(\n+)/g, "\n".repeat(n))
);

// Pregunta 7
/** Cada frase debe aparecer en párrafo aparte */
export const convertToParagraphs = transformFn(
  {}, (text) =>
    text
      .split("\n")
      .map((line) =>
        line
          .split(".")
          .map((p) => p.trimStart())
          .join(".\n\n")
          .trimEnd()
        )
      .join("\n")
);
```



```
map([🐱, 🍌, 🐔, 🌽], cook)
=> [🍷, 🍷, 🍷, 🍷]

filter([🍷, 🍷, 🍷, 🍷], isVegetarian)
=> [🍷, 🍷]

reduce([🍷, 🍷, 🍷, 🍷], eat)
=> 🍷
```

Nuestro "sello personal"



Set options

CLOSE

limitWidth

Width

Up Down Remove

80

filterParagraphs

Min sentences

Max sentences

Up Down Remove

4

10

addIndentationToPhrase

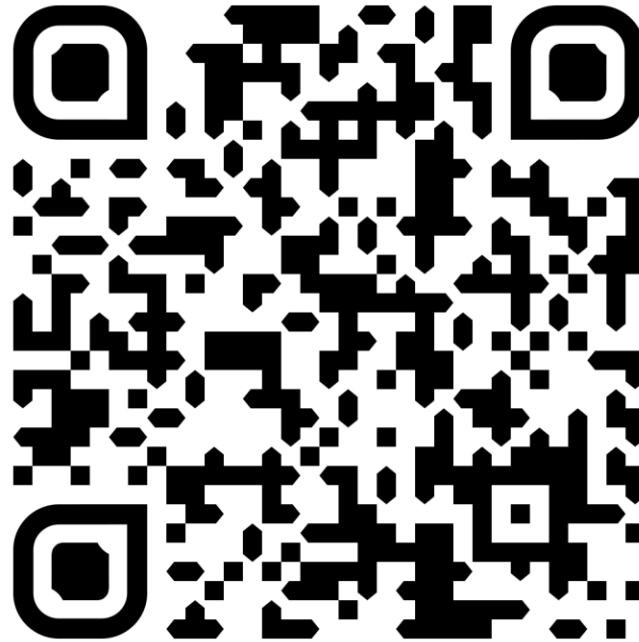
ADD FILTER

Local Storage



DEMO

- QR (try it yourself)
- Live demo



<https://iic3585-2023.github.io/functional-js-grupo-01/>

|> ipe Operator

Pipes

- Permite realizar una cadena de procesos sin variables intermediarias
- No necesariamente es un operador

```
// without pipe
```

```
assert.strictEqual(double(len('aaa')), 6)
```

```
// with pipe
```

```
assert.strictEqual(pipe('aaa', len, double), 6)
```

```
foo(bar(baz(new_function(other_function()))))
```

```
other_function() |> new_function() |> baz() |> bar() |> foo()
```



```
julia> 1:10 |> sum |> sqrt  
7.416198487095663
```

```
julia> 1:3 .|> (x -> x^2) |> sum |> sqrt  
3.7416573867739413
```

```
iex> "Elixir rocks" |> String.upcase() |> String.split()  
["ELIXIR", "ROCKS"]
```

TC39 Pipe Operator Proposal

```
const json = pkgs[0]  
|> npa(^).escapedName  
|> await npmFetch.json(^, opts);
```

github.com/tc39/proposal-pipeline-operator

¿y cuando?

- Aceptada en stage 1 el 2017-07 y stage 2 en 2021-08
- Definir el syntaxis final es complejo
- Probablemente en varios años :(

github.com/tc39/proposal-pipeline-operator/blob/main/HISTORY.md

Cortar párafos luego de n caracteres sin cortar palabras

```

// Pregunta 3
const reduceFlatWithStackLastItem = <In, Out>(f: (last: Out[], current: In) => Out[][]) =>
  arr.reduce<In, Out[][]>([[ ]], (acc, value) => [
    ...acc.slice(0, acc.length - 1),
    ...f(acc[acc.length - 1], value),
  ]);

/** El ancho del texto debe ser a lo más •n• (sin cortar palabras) */

export const limitWidth = transformFn({ n: { name: "Width", default: 80 } }, (text, { n }) =>
  fn.pipe(
    // Separar por lineas
    text.split("\n"),
    arr.chain(
      fn.flow(
        // Separar cada linea por espacio, quedamos string[]
        (line) => line.split(" "),
        // Separamos las lineas, quedando string[][]
        reduceFlatWithStackLastItem((last: string[], word) =>
          last.join(" ").length + word.length > n ? [last, [word]] : [...last, word]
        ),
        // Unimos cada linea que quedó
        arr.map((line) => line.join(" "))
      )
    ),
    // Unimos las lineas
    (lines) => lines.join("\n")
  );
```