Programación Funcional

01/04/2020

Grupo 2
Paul Heinsohn
Sebastian Carreño

Profesor Jaime Navon

Tema 1

- Transformación de archivo de texto
- Combinación de operaciones
-

Input

- Archivo de texto:

```
1   Esta es la primera frase del primer parrafo. Esta es la segunda frase del primer parrafo.
    Esta es la tercera frase del primer parrafo. Esta es la cuarta frase del primer parrafo.
2   Esta es la primera frase del segundo parrafo. Esta es la segunda frase del segundo parrafo.
    Esta es la tercera frase del segundo parrafo.
3   Esta es la primera frase del tercer parrafo. Esta es la segunda del tercer parrafo.
4   Esta es la primera frase del cuarto parrafo.
5
```

Estructura Programa

1. Abrir archivo
2. Preparar para procesar
3. Definir operaciones
4. Ejecutar el pipeline

```
fs.readFile('text.txt', (err, data) => {
  const text = data.toString();
  const paragraphs = getParagraphs(text);

  const operations = [a(2), b(1), h(1), c(200), d(3), e(2), f(2), g];
  console.log(pipe([...operations, joinText])(paragraphs));
});
```

Preparación de Input

1. Texto → Lista de listas

```
const compose = (a, b) => c => a(b(c));

const splitParagraphs = text => text.split("\n");
const splitPhrases = paragraphs => paragraphs.map(p => p.split("."));

const getParagraphs = function(text) {
  const textSplitter = compose(splitPhrases, splitParagraphs);
  const textSplitted = textSplitter(text);
  ...
};
```

Preparación de Input

2. Limpieza y agregación de eliminados

```
const getParagraphs = function(text) {
  ...
  const popEmpty = splitted => _.dropRight(splitted, 1).map(p => _.dropRight(p, 1));
  const addDots = splitted => splitted.map(p => p.map((frase, i) => insertDot(frase, p.length - 1, i)))
  const cleanData = compose(addDots, popEmpty);

  return cleanData(textSplitted);
};
```

```
[ [ 'Esta es la primera frase del primer parrafo.',
    ' Esta es la segunda frase del primer parrafo.',
    ' Esta es la tercera frase del primer parrafo.',
    ' Esta es la cuarta frase del primer parrafo.\n' ],
  [ 'Esta es la primera frase del segundo parrafo.',
    ' Esta es la segunda frase del segundo parrafo.',
    ' Esta es la tercera frase del segundo parrafo.\n' ],
  [ 'Esta es la primera frase del tercer parrafo.',
    ' Esta es la segunda del tercer parrafo.\n' ],
  [ 'Esta es la primera frase del cuarto parrafo.\n' ] ]
```

Definición de Operaciones

- Debemos entregar una lista de las operaciones que queremos realizar
- Pueden estar en cualquier orden:

```
const operations = [a(2), b(1), h(1), c(200), d(3), e(2), f(2), g];
```

Pipeline

- Recibe nuestra lista de listas
- Operaciones se ejecutan
- Retorna un texto

```
const pipe = functions => data => {
  return functions.reduce(
    (value, func) => func(value),
    data
  );
}
const operations = [a(2), b(1), h(1), c(200), d(3), e(2), f(2), g];
console.log(pipe([...operations, joinText])(paragraphs));
```

Ejemplos de Operaciones

a) Cada frase debe empezar con n espacios en blanco (después de un punto seguido)

d) Cada párrafo debe tener n espacios de sangría

```
const addSpace = (text, n, index) => index !== 0
    ? " ".repeat(n) + text
    : text;
const addSpaces = (paragraph, n) => paragraph.map((frases, i) => addSpace(frases, n, i));


const a = n => paragraphs => paragraphs.map(p => addSpaces(p, n));

const d = n => paragraphs => paragraphs.map(p => [addSpace(p[0], n, 1), ...p.slice(1)]);
```

b) Cada Párrafo debe estar separado por n líneas

```
const b = n => paragraphs => paragraphs.map(p => [...p.slice(0, -1), p.slice(-1) + "\n".repeat(n)]);
```

```
[ [ 'Esta es la primera frase del primer parrafo.',
    ' Esta es la segunda frase del primer parrafo.',
    ' Esta es la tercera frase del primer parrafo.',
    ' Esta es la cuarta frase del primer parrafo.\n\n\n\n' ],
  [ 'Esta es la primera frase del segundo parrafo.',
    ' Esta es la segunda frase del segundo parrafo.',
    ' Esta es la tercera frase del segundo parrafo.\n\n\n\n' ],
  [ 'Esta es la primera frase del tercer parrafo.',
    ' Esta es la segunda del tercer parrafo.\n\n\n\n' ],
  [ 'Esta es la primera frase del cuarto parrafo.\n\n\n\n' ] ]
```

Ejemplos de Operaciones

c) El ancho del texto debe ser a lo más n (sin cortar palabras)

```
const cutParagraphs = n => paragraphs => paragraphs.map(p => cutWidth(n)(p));
const cutWidth = n => paragraph => paragraph.length > n
    ? sliceString(n)(paragraph)
    : paragraph;
const sliceString = n => paragraph => paragraph[n] !== ' '
    ? sliceString(n - 1)(paragraph)
    : paragraph.slice(0, n) + '\n';
```

```
const c = n => pipe([joinPhrases, cutParagraphs(n), splitPhrases]);
```

EJEMPLOS DE OPERACIONES

e) Se ignoran los párrafos que tienen menos de n frases

```
const e = n => paragraphs => paragraphs.filter(p => !(p.length < n));
```

f) Se ignoran los párrafos que tienen más de n frases

```
const f = n => paragraphs => paragraphs.filter(p => !(p.length > n));
```

EJEMPLOS DE OPERACIONES

g) Cada frase debe aparecer en un párrafo aparte

```
const g = paragraphs => _.flatten(paragraphs)
                          .map(p => p.match(/\n/g) ? [p] : [p + "\n"]);
```

```
[ [ 'Esta es la primera frase del primer parrafo.\n' ],
  [ ' Esta es la segunda frase del primer parrafo.\n' ],
  [ ' Esta es la tercera frase del primer parrafo.\n' ],
  [ ' Esta es la cuarta frase del primer parrafo.\n' ],
  [ 'Esta es la primera frase del segundo parrafo.\n' ],
  [ ' Esta es la segunda frase del segundo parrafo.\n' ],
  [ ' Esta es la tercera frase del segundo parrafo.\n' ],
  [ 'Esta es la primera frase del tercer parrafo.\n' ],
  [ ' Esta es la segunda del tercer parrafo.\n' ],
  [ 'Esta es la primera frase del cuarto parrafo.\n' ] ]
```

EJEMPLOS DE OPERACIONES

h) Solo las primeras n frases de cada párrafo

```
const h = n => paragraphs => paragraphs.map(p => n < p.length
    ? [...p.slice(0, n - 1), p[n - 1] + p[p.length - 1].match(/\n/g).join("")]
    : p.slice(0, n));
```

```
[ [ 'Esta es la primera frase del primer parrafo.',
    ' Esta es la segunda frase del primer parrafo.\n' ],
  [ 'Esta es la primera frase del segundo parrafo.',
    ' Esta es la segunda frase del segundo parrafo.\n' ],
  [ 'Esta es la primera frase del tercer parrafo.',
    ' Esta es la segunda del tercer parrafo.\n' ],
  [ 'Esta es la primera frase del cuarto parrafo.\n' ] ]
```