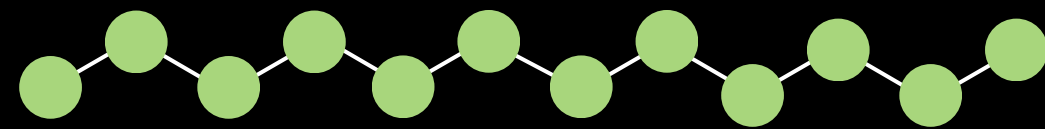




# TAREA 1

Diseño avanzado de aplicaciones web



Grupo 3

# ¿POR QUE PROGRAMACIÓN FUNCIONAL?

- Evitar *datos mutables* y compartir estados entre funciones.
- Enfoque en "qué" estamos haciendo y no en "cómo" se hace.
- Facilidad de pruebas al tener enfoque en *funciones puras*.
- ¿Software mucho más legible y fácil de testear? Pongamoslo a prueba 😊



```
1 // rule 4: each paragraph must have n spaces of indentation
2 const increaseIndentation = ([first, ...rest], n) => [first.padStart(n + first.length), ...rest];
3 const convertParagraphsIndentations = (paragraphs, n) => paragraphs.map(paragraph => increaseIndentation(paragraph, n));
4
```

Se usó **desestructuración** para operar solo sobre la primera frase de cada párrafo.

Nos permite **extensibilidad** a textos de cualquier tamaño.





```
1  const readFile = (path) => {  
2    return _.attempt(() => {  
3      return fs.readFileSync(path, 'utf8');  
4    });  
5  };
```

`attempt` es una función de Lodash para manejar errores evitando que se caiga todo el programa.



```
1  const parseFileData = (data) => {  
2    return _.chain(data)  
3      .split('\n\n')  
4      .map(paragraph => paragraph.split('.')  
5        .filter(sentence => sentence.length > 0)  
6        .map(sentence => sentence.trim() + '.'))  
7      .value();  
8  };
```

**chain** es una función de Lodash para encadenar una secuencia de funciones.







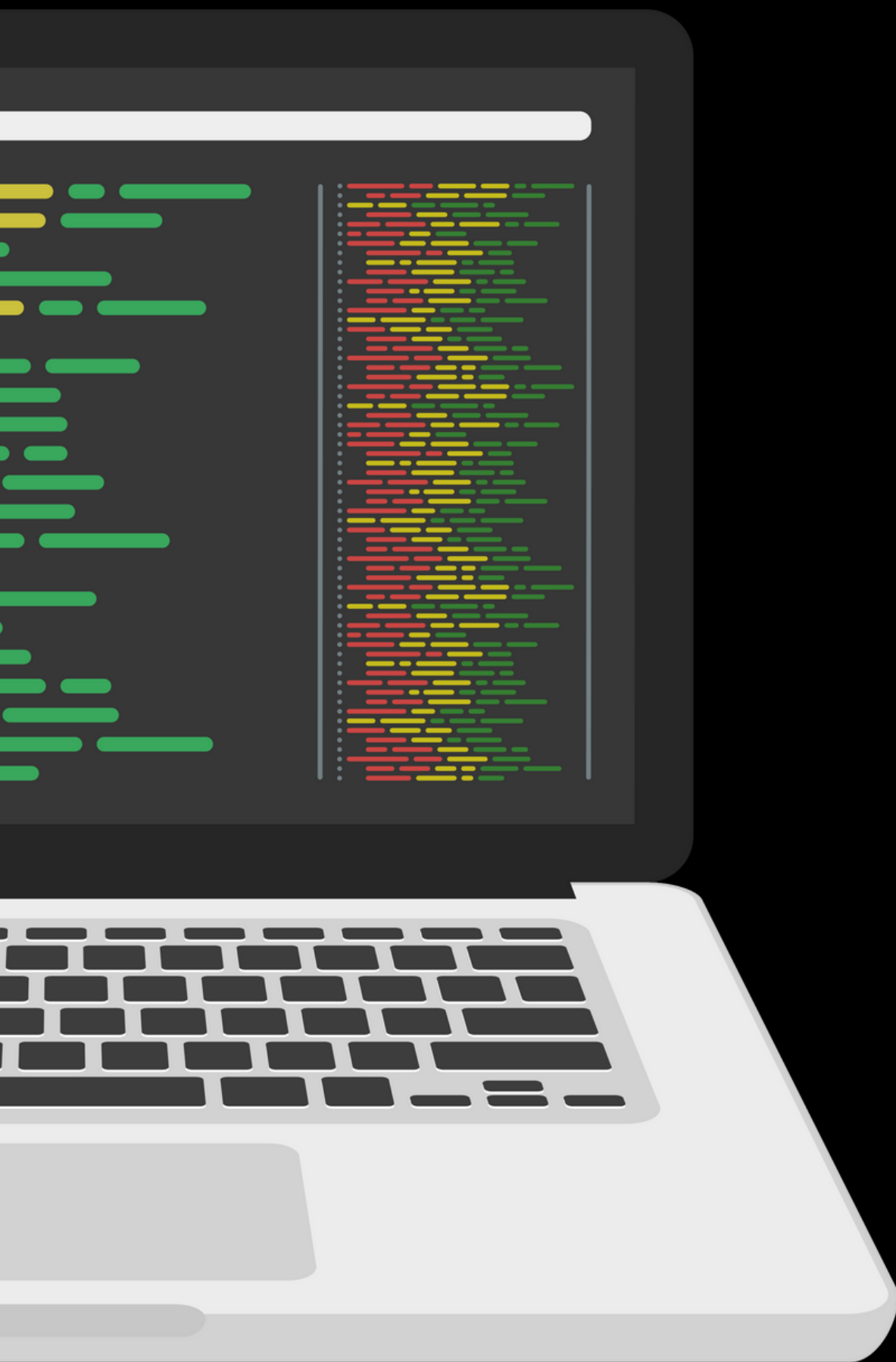
```
1  const example1 = (text, n) => _.flow(  
2    util.convertParagraphsIndentations,  
3    _.partial(util.convertParagraphLineBreaks, _, n),  
4    _.partial(util.filterWithLessSentences, _, n)  
5  )(text, n)  
6
```

`flow` es una función de `Lodash` para generar una nueva función compuesta que retorna el resultado de invocar las funciones proporcionadas. Muy similar a un `pipeline`.

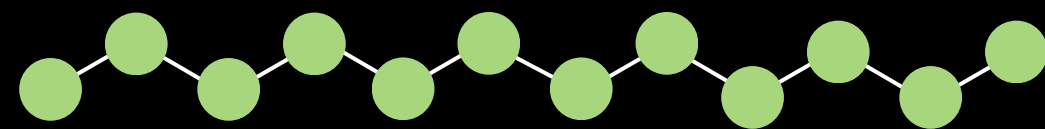


```
1  const example2 = (text, n, m) => _.flow(  
2    util.trimAllParagraphs,  
3    _.partial(util.takeFirstSentences, _, m),  
4    _.partial(util.convertParagraphLineBreaks, _, m),  
5  )(text, n)
```

`partial` nos permite recibir el resultado de la función anterior y fijar nuevos parametros en la llamada.



# DEMO



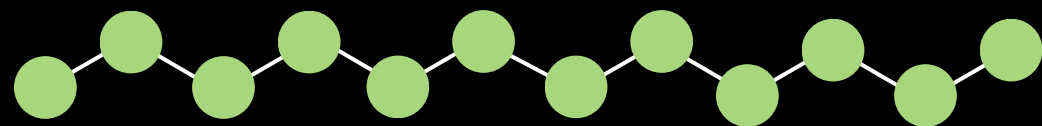
Grupo 3





# APRENDIZAJES

- Código más corto, eficiente y legible, pero requiere mayor curva de aprendizaje
- Lodash tiene muchas funciones optimizadas
- Muchas soluciones pueden ser modeladas utilizando funciones, sin necesidad de usar clases u otras estructuras :)





Grupo 3