

TARJETAS

Nos han pedido un desarrollo para generar de forma dinámica tarjetas con distintas celdas de tamaño regular.

1. Las tarjetas estarán compuestas de tantas líneas como se desee.
2. Las líneas tendrán tantas celdas como se desee
3. Cada celda tendrá un tamaño definido en pixeles (px) (alto y ancho).
4. Vamos a crear tres clases:
 - a. Tarjeta
 - b. Linea
 - c. Celda
5. Utilizaremos un layout HTML para cada objeto construido con las clases, serán constantes en el desarrollo, en este caso la altura (height) y anchura (width) de los elementos serán calculados por nuestro desarrollo.

```
const layoutHTMLTarjeta = {
  tipo: 'section',
  atributos: {
    name: "tarjeta",
  },
  estilos: {
    backgroundColor: 'lightgrey',
    display: 'flex',
    flexDirection: 'column',
    alignItems: 'center',
    justifyContent: 'center',
  }
};

const layoutHTMLLinea = {
  tipo: 'div',
  atributos: {
    name: "linea"
  },
  estilos: {
    backgroundColor: 'dodgerblue',
    display: 'flex',
    flexDirection: 'row',
    justifyContent: 'space-around',
    alignItems: 'center'
  }
}

const layoutHTMLCelda = {
  tipo: 'div',
  atributos: {
    name: 'celda'
  },
  estilos: {
    backgroundColor: 'whitesmoke',
    display: 'flex',
    justifyContent: 'center',
    alignItems: 'center',
    placeContent: 'center',
    fontSize: '2em',
    fontFamily: 'Arial',
    fontWeight: 'bold',
    margin: 'auto'
  }
}
```

Este desarrollo consta de 2 SPRINTS

Utilizaremos el siguiente html, con el src = "tarjetas.js"

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>TARJETA</title>
</head>
<body>
</body>
<script src="tarjetas.js"></script>
</html>
```

SPRINT 1. – Maquetación de la Tarjeta	5 puntos
Función generarElementoHTML	2 puntos
<p>Esta función, similar al método estático que utilizamos en el blog, tomará como parámetros:</p> <ul style="list-style-type: none">• Un layout• La altura del elemento a generar• La anchura del elemento a generar <p>Estos parámetros serán pasados por las distintas funciones constructoras de cada una de las clases a utilizar: Tarjeta, Linea, Celda.</p> <p>Un layout se refiere a uno de los establecidos como constantes y consta de las siguientes propiedades:</p> <ol style="list-style-type: none">1. tipo, Recoge el tipo de elemento a crear2. atributos, es un objeto con todos los atributos del elemento3. estilos, es un objeto con todos los estilos a aplicar al elemento <p>Los parámetros anchura y altura se pasarán como numéricos (enteros) y se referirán a pixeles (px)</p> <p>La función devolverá un elemento HTML construido con la especificación dada, con la altura y anchura requerida.</p>	

CLASES Tarjeta/Linea/Celda		3 puntos
Para abreviar, en este desarrollo no utilizaremos setters o getters		
CLASE Tarjeta		
Propiedades		
Constructor (parámetros entrada que pasan a propiedades)	numeroDeLineas	Parámetro, por defecto 5
	numeroDeColumnas	Parámetro, por defecto 5
	alto	Se trata del alto de una celda. Parámetro, por defecto 50
	ancho	Se trata del ancho de una celda. Parámetro, por defecto 50
Constructor (otras propiedades)	anchoTarjeta	Se calculará el ancho de la tarjeta en función del ancho de la celda y el número de columnas
	altoTarjeta	Se calculará el alto de la tarjeta en función del alto de la celda y el número de columnas
	ElementoHTML	Llamará a la función generarElementoHTML con el layout correspondiente y el alto y ancho de la tarjeta (propiedades calculadas anteriormente)
	lineas	Array vacío que recogerá un array con cada línea que generemos para el cartón
Constructor - Proceso		
<p>El constructor de la clase Tarjeta, generará todas las líneas requeridas en el parámetro de entrada.</p> <p>Se llamará al método generarLineas(alto, ancho), los parámetros alto y ancho se refieren a los establecidos para la celda, no los calculados para la tarjeta.</p> <p>Desplegaremos el layout del cartón en el document.body, en la posición 'beforeend'</p>		
Método	generarLineas(alto,ancho)	<p>En un bucle, generará para cada línea un objeto de la clase Linea, al que le pasará 3 parámetros:</p> <ul style="list-style-type: none"> La propia tarjeta El primer número de la línea. Se calculará en función del índice del bucle, por ejemplo, para la primera línea, en el caso de que tenga 9 columnas, irá del 1 al 9, la segunda del 10 al 18, y la tercera del 19 al 27 El último número de la línea, tal como se explica en el punto anterior. El alto y el ancho de la celda <p>Una vez generada el objeto línea, este se añadirá al array líneas. De esta forma tendremos referenciadas todas las líneas de la tarjeta.</p>

CLASE Linea		
Propiedades		
Constructor (parámetros entrada que pasan a propiedades)	tarjeta	Parámetro, recoge el objeto tarjeta
	desde	Número inicial de la línea
	hasta	Número final de la línea
	alto	Alto de la celda
	ancho	Ancho de la celda
	anchoLinea	Se calculará el ancho de la linea en función del ancho de la celda y el número de columnas
	altoLinea	El alto de una línea es equivalente al alto de una celda
Constructor (otras propiedades)	ElementoHTML	Lamará a la función generarElementoHTML con el layout correspondiente y el alto y ancho de la linea (propiedades calculadas anteriormente)
	celdas	Array vacío que recogerá un array con cada celda que generaremos para la línea
Constructor - Proceso		
<p>Se invocará al método generarCeldas(alto, ancho) descrito a continuación para generar todas las celdas.</p> <p>Desplegaremos el layout de la línea en el elementoHTML del cartón en la posición 'beforeend'.</p>		
Método	generarCeldas(alto,ancho)	<p>En un bucle se generarán todas las celdas correspondientes de la línea, el bucle se iniciará en el valor desde y finalizará en el hasta.</p> <p>El bucle generará un objeto de la clase Celda, al que se le pasarán los siguientes parámetros:</p> <ul style="list-style-type: none"> • El propio objeto línea • El numero a generar. • El alto y el ancho de la celda <p>Una vez generado el objeto celda, este se añadirá al array de celdas de la línea.</p>

CLASE Celda																																																																																																																																																														
Propiedades																																																																																																																																																														
Constructor (parámetros entrada que pasan a propiedades)	linea	Parámetro, recoge el objeto linea																																																																																																																																																												
	numero	Numero de la celda																																																																																																																																																												
	alto	Alto de la celda																																																																																																																																																												
	ancho	Ancho de la celda																																																																																																																																																												
Constructor (otras propiedades)	ElementoHTML	Llamará a la función generarElementoHTML con el layout correspondiente a la celda y el alto y ancho de la celda y devolverá el elemento que quedará asignado a la propiedad.																																																																																																																																																												
Constructor - Proceso																																																																																																																																																														
Al innerHTML del elemento HTML generado se le asignará el número																																																																																																																																																														
Desplegaremos el layout de la celda en el de la línea, en la posición ‘beforeend’.																																																																																																																																																														
SALIDA PREVISTA																																																																																																																																																														
Construyendo 3 tarjetas		<table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td></tr><tr><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td></tr><tr><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td></tr><tr><td>37</td><td>38</td><td>39</td><td>40</td><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td><td>47</td><td>48</td></tr><tr><td>49</td><td>50</td><td>51</td><td>52</td><td>53</td><td>54</td><td>55</td><td>56</td><td>57</td><td>58</td><td>59</td><td>60</td></tr><tr><td colspan="12"></td></tr><tr><td colspan="2">1</td><td colspan="2">2</td><td colspan="8"></td></tr><tr><td colspan="12"></td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td colspan="7"></td></tr><tr><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td colspan="7"></td></tr><tr><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td colspan="7"></td></tr><tr><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td colspan="7"></td></tr><tr><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td colspan="7"></td></tr></table>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60													1		2																						1	2	3	4	5								6	7	8	9	10								11	12	13	14	15								16	17	18	19	20								21	22	23	24	25							
1	2	3	4	5	6	7	8	9	10	11	12																																																																																																																																																			
13	14	15	16	17	18	19	20	21	22	23	24																																																																																																																																																			
25	26	27	28	29	30	31	32	33	34	35	36																																																																																																																																																			
37	38	39	40	41	42	43	44	45	46	47	48																																																																																																																																																			
49	50	51	52	53	54	55	56	57	58	59	60																																																																																																																																																			
1		2																																																																																																																																																												
1	2	3	4	5																																																																																																																																																										
6	7	8	9	10																																																																																																																																																										
11	12	13	14	15																																																																																																																																																										
16	17	18	19	20																																																																																																																																																										
21	22	23	24	25																																																																																																																																																										
<pre>let miTarjeta1 = new Tarjeta(5,12,60,80); let miTarjeta2 = new Tarjeta(1,2,150,150); let miTarjeta3 = new Tarjeta();</pre>																																																																																																																																																														

SPRINT 2. – Dando contenido

5 puntos

1. En este sprint, vamos a modificar el contenido de nuestras tarjetas con un juego de palabras.
2. Partiremos de un array de múltiples palabras. Para las pruebas, tenéis disponible un array con los municipios de Asturias.
3. De este array tomaremos de forma aleatoria, tantas palabras como columnas tenga la tarjeta
4. De esta forma, compondremos las columnas con las todas las letras que podamos de cada palabra. Por lo tanto, se leerán en vertical

EJEMPLOS (2 TARJETAS GENERADAS)

S	V	R	C	C								
a	e	i	a	o								
n	g	b	r	a								
t	a	a	a	ñ								
a	d	d	v	a								
	e	e	i									
E	o	s	a									
u		e										
l		l										
a		l										
l		a										
i												
a												
d												
A	C	A	C	P	B	G	C	S	P	R	P	
l	a	v	o	i	o	o	a	a	e	i	e	
l	b	i	a	l	a	z	s	n	ñ	b	ñ	
e	r	l	ñ	o	l	ó	o		a	a	a	
r	a	é	a	ñ		n		M	m	d	m	
	l	s		a				a	e	e	e	
	e							r	l	d	l	
	s							t	l	e	l	

```
let miTarjeta3 = new Tarjeta(15,5,50,50, municipios);  
let miTarjeta1 = new Tarjeta(8,12,60,80, municipios);
```

Función numeroAleatorio		0,5
Esta función tomará como parámetros, dos números (desde, hasta) entre los que deberá devolver un entero aleatorio cuyo valor podrán estar incluidos estos dos valores (desde, hasta) Así, la llamada a numeroAleatorio(10,19), devolverá un número entre 10 y 19, pudiendo incluir estos 2.		
Modificaciones CLASE Tarjeta		3
Propiedades		
Constructor (parámetros entrada que pasan a propiedades)	contenido	Recoge el array con las palabras que utilizaremos para crear la tarjeta.
Constructor (otras propiedades)	palabras	Array, en principio vacío donde recogeremos las palabras que seleccionaremos para las tarjetas.
Métodos	seleccionarContenido()	Este método seleccionará, del array contenido y de forma aleatoria, tantas palabras como columnas tenga la tarjeta, cada palabra seleccionada se añadirá al array palabras. Las palabras no pueden repetirse.
	generarLineas()	Modificaremos este método para poder construir la línea con las letras que correspondan de cada una de las palabras. Crearemos un array datosLinea, con cada una de las letras que correspondan. Es decir, por ejemplo, la primera letra de cada palabra en la primera línea. Ese array datosLinea se lo pasaremos a la llamada a la clase Linea para que esta genere las celdas con cada letra.

Modificaciones CLASE Linea		1,5
Propiedades		
Constructor (nuevos parámetros entrada que pasan a propiedades)	datos	Recoge el array con las letras que utilizaremos para componer las celdas de la línea
Método	generarCeldas	Modificamos el método que mediante un bucle procesará el array datos y pasará cada letra a la clase Celda la letra a incluir en la celda. Si la posición del array de datos no contiene datos (undefined), la letra será una cadena vacía.

ARRAY MUNICIPIOS

```
const municipios =
[
  'Allande',
  'Aller',
  'Amieva',
  'Avilés',
  'Belmonte de Miranda',
  'Bimenes',
  'Boal',
  'Cabrales',
  'Cabranes',
  'Candamo',
  'Cangas de Onís',
  'Cangas del Narcea',
  'Caravia',
  'Carreño',
  'Caso',
  'Castrillón',
  'Castropol',
  'Coaña',
  'Colunga',
  'Corvera de Asturias',
  'Cudillero',
  'Degaña',
  'Franco, El',
  'Gijón',
  'Gozón',
  'Grado',
  'Grandas de Salime',
  'Ibias',
  'Illano',
  'Illas',
  'Langreo',
  'Laviana',
  'Lena',
  'Llanera',
  'Llanes',
  'Mieres',
  'Morcín',
  'Muros de Nalón',
  'Nava',
  'Navía',
  'Noreña',
  'Onís',
  'Oviedo',
  'Parres',
  'Peñamellera Alta',
  'Peñamellera Baja',
  'Pesoz',
  'Piloña',
  'Ponga',
  'Pravia',
  'Proaza',
  'Quirós',
  'Regueras, Las',
  'Ribadedeva',
  'Ribadesella',
  'Ribera de Arriba',
  'Riosa',
  'Salas',
  'San Martín de Oscos',
  'San Martín del Rey Aurelio',
  'San Tirso de Abres',
  'Santa Eulalia de Oscos',
  'Santo Adriano',
  'Sariego',
  'Siero',
  'Sobrescobio',
  'Somiedo',
  'Soto del Barco',
  'Tapia de Casariego',
  'Taramundi',
  'Tevera',
  'Tineo',
  'Valdés',
  'Vegadeo',
  'Villanueva de Oscos',
  'Villaviciosa',
  'Villayón',
  'Yernes y Tameza' ]
```