

## Prueba de postulación para Front-end

12/08/2019

Realice un proyecto en angular 7 o superior en el que se tendrá que implementar lo siguiente:

1. Una vista en la cual se deberán implementar los siguientes campos:
  - a. Un botón, cuya acción será el consumir la información del endPoint <http://168.232.165.184/prueba/array> .  
Con la información obtenida se deberán completar los demás campos del formulario.
  - b. Una tabla que deberá contener las siguientes columnas:
    - i. N°: índice n incremental.
    - ii. number: El numero procesado.
    - iii. quantity: cantidad de ocurrencias del número procesado (cada número sólo puede aparecer una vez).
    - iv. first position: índice de primera posición dentro del arreglo.
    - v. last position: índice de última posición dentro del arreglo.
  - c. Complementario a la tabla anterior:
    - i. Pintar en verde la fila que contenga ocurrencias  $\geq 2$ .
    - ii. Pintar en amarillo la fila que contenga ocurrencias = 1.
    - iii. Pintar en gris la fila que contenga ocurrencias  $< 1$ .
  - d. Imprimir en un textbox los valores ordenados de menor a mayor (sin utilizar `array.sort()`)
2. Una vista en la cual se deberán implementar los siguientes campos:
  - a. Un botón cuya acción será el consumir la información del endPoint <http://168.232.165.184/prueba/dict> .
  - b. Una tabla que deberá contener las siguientes columnas en su encabezado:
    - i. N°: índice n incremental.
    - ii. Valores ordenados de la 'a' a las 'z'.
  - c. Para el cuerpo de la tabla, se deberá procesar el arreglo consumido del endpoint de tal manera de poblar la tabla. Las filas serán llenadas con la cantidad de apariciones de las letras del valor de la llave 'paragraph'.
    - i. Incorporar una última columna en la que se deberá (de contener números) sumar los valores numéricos absolutos del valor de la llave 'paragraph'.

N°	a	b	c	d	e	z	Suma numericos contenidos
1	34	5	6	3	4	0	234
2	12	7	43	4	12	0	0
3	67	5	0	0	4	0	124
4	0	0	0	0	0	0	34
5	12	1	8	0	4	0	0

- Se deberá crear un repositorio en bitbucket y enviar el link para su revisión.
- Utilizar bootstrap para los templates del proyecto.
- Quedará de parte del evaluado el cómo se realizará la navegación para moverse por las vistas.
- Para la situación 1 el resultado entregado por el endPoint podría variar en cantidad de valores, tiempos de respuestas e incluso es posible que la respuesta retornada sea nula. Dado lo anterior, opcionalmente, se pide el manejar algún flujo en la vista para estos casos. (Ej. loading)

- Para la identificación de números considerar los siguiente casos:
  - "(texto)... 1,005...(texto)": los números serán 1 y 5 por separado
  - "(texto)... -15...(texto)": el numero será 15
- La prueba consta de un tiempo de 24 horas para su desarrollo y se deberá enviar el link del repositorio a [rrhh@miningtag.cl](mailto:rrhh@miningtag.cl).
- Asegurarse que el repositorio enviado tenga visibilidad pública o de otra manera se considerará como inválida la postulación.
- En ambos casos la interfaz de respuesta será la siguiente:

```
export interface IRemoteResponse {  
  data?: any;  
  error?: any;  
  success: boolean;  
}
```