

## EXAMEN DESARROLLADOR WEB GLOBAL PRIMEX TECNOLOGÍA

### Notas:

- En el examen práctico se tendrá en cuenta la calidad del código (estándar de definición de tipos de datos, métodos y clases), además de la lógica utilizada y la generalización de los métodos.
- La evaluación principal será dirigida al framework de desarrollo Angular, pero se tendrá en cuenta la forma de implementar el código en el lado del servidor con NodeJS.
- La base de datos a usar puede ser cualquiera SQL y puede usar el ORM que desee.
- Si es posible, use GraphQL para la comunicación entre servicios y obtendrá puntos extra.
- Si es posible, use NgRX para manejar el estado de la aplicación y obtendrá puntos extra.
- Se tendrá en cuenta su capacidad de investigar y preguntarnos sobre cualquier inquietud que posea en cuanto al desarrollo de este proyecto.
- Es importante que demuestre el manejo de los estándares de JS ES6 en adelante.
- Es importante que demuestre el manejo de la librería RxJS al ser fundamental en el desarrollo con Angular.
- Cualquier duda puede consultarla conmigo en el chat.

### Prueba práctica:

- Debe ser entregada en un archivo zip o un link a github.

1. Crear una aplicación que permita mostrar una lista de personas, los datos a mostrar de cada persona son los siguientes:

- Nombre(s)
- Apellido Paterno
- Apellido Materno
- Dirección
- Teléfono

2. La lista de personas se mostrará en una tabla, la cual debe contar con un buscador (campo de texto) que permita filtrar las personas por nombre o apellido (paterno o materno).

3. También es necesario que en la vista web se cuente con un botón que muestre un formulario para registrar, actualizar y eliminar una nueva persona, los campos del formulario deben de ser los siguientes:

- Nombre(s)
- Apellido Paterno
- Apellido Materno
- Dirección
- Teléfono

**Prueba teórica:**

1. Está recibiendo dos flujos de datos de dos sensores de temperatura (2 observables que devuelven números enteros), ¿qué operador de RxJS utilizaría para recibir estos dos datos en la misma suscripción?

```
combineLatest(temp1$,temp2$).subscribe([temp1Value,temp2Value] => {  
    Console.log('temp1 is ${ temp1Value } and temp2 is ${ temp2Value }')  
})
```

2. Si tiene dos llamadas al servidor y la segunda llamada depende de la primera, ¿cómo manejaría con RxJS esta secuencia de llamadas?

**R= Con async y await o interval,timer**

3. Tengo en un servidor un archivo de texto que está en minúscula y ocupa 2GB en el disco duro, pero le solicitan que todo el archivo debe ser pasado a mayúsculas, ¿cómo lo haría?

**Recibir datos del archivo en una variable y recorrer cada línea con una iteración que tenga un método uppercase();**

4. Tiene un arreglo de strings los cuales deben ser filtrados por su longitud mayor a dos y a la vez convertidos a un array de enteros con la longitud de cada string, ¿cómo lo haría? Ejemplo de entrada y salida: ["hola", "mundo", "es", "una", "prueba"] => [4, 5, 3, 6]

```
5. <script>
6.     const saludo = ["hola","mundo","es","una","prueba"];
7.     const saludoNum= [];
8.     for (let i = 0; i < saludo.length; i++) {
9.
10.         if(saludo[i].length > 2){
11.             saludoNum.push(saludo[i].length);
12.         }
13.     }
14.
15.     console.log(saludo);
16.     console.log(saludoNum);
17. </script>
```

18. Tiene un arreglo de números, los cuales pueden ser o no repetidos, ¿cómo eliminaría los repetidos? ¿Cómo los ordenaría en forma ascendente? Ejemplo de entrada y salida: [1, 2, 5, 10, 8, 8, 1, 3, 4, 5] => [1, 2, 3, 4, 5, 8, 10]

```
<script>
    //1.crear arreglo
    var numeros = [1, 2, 5, 10, 8, 8, 1, 3, 4, 5] ;
    //2.crear nuevo set arreglo sin repetidos
    var nuevosNumeros = [...new Set(numeros)];
    //3.Ordenar Arreglo
    nuevosNumeros.sort( (a,b)=> {
        if(a==b){
            return 0;
        }
        if(a<b){
            return -1;
        }
        return 1;
    });
</script>
```