Nativo Explore

Prueba Técnica

TEÓRICO

Cada punto de la prueba deberá contener su código específico. El siguiente es un ejemplo de un como debe estar codificado cada punto de la prueba.

```
// Este ejemplo toma una lista de números y da como resultado la suma de todos ellos.
let numeros = "1,2,3,4,7,2,3,7,2,8,9";

// Convierte el listado de numeros de un arreglo de caracteres a una lista
numeros = numeros.split(',')
let sumatoria = 0

// Itera sobre todos los numeros de la lista y los suma a la variable sumatoria
for (let numero in numeros) {
  sumatoria = sumatoria + numero
}

// Imprime el resultado
console.log(sumatoria)
```

1. TEÓRICO
    1.1. Explique que es un puntero y cómo funciona.
    1.2. ¿Cuál es la diferencia entre un lenguaje interpretado y uno compilado? Pueden ser igual de rápidos en cuanto a su ejecución?
    1.3. Dada una lista de números enteros, imprima cuál es el menor de los números de la lista.
    1.4. Dada una lista de números enteros, imprima cuál es la suma de los números al cuadrado.
    1.5. Dada una lista de números enteros, ordene e imprima la lista de menor a mayor.
    1.6. Dada una lista de números enteros, imprima un diccionario con el número de repeticiones para cada número en la lista.
    1.7. Dada una palabra, imprima el conjunto de de anagramas, que se pueden realizar con la palabra.
    1.8. Usando javascript o typescript, codifique y explique (con comentarios) 3 patrones de diseño de Software (1 creacional, 1 estructural y 1 comportamental). Más información: https://es.wikipedia.org/wiki/Patr%C3%B3n_de_dise%C3%B1o

PRÁCTICO

Se requiere que se realice una pequeña aplicación de ToDo List. Los siguientes son los prerrequisitos técnicos del proyecto:

- Se debe crear una aplicación que permita el registro de tareas ToDo por hacer.
- La aplicación debe contar con las siguientes funcionalidades:
    - **Listar las tareas registradas:** Se debe poder ver el listado de tareas (pendientes y completadas) que se han registrado en el sistema.
    - **Crear una nueva tarea:** Se debe permitir crear una nueva tarea dentro de la aplicación.

    - **Modificar una tarea:** Se debe permitir modificar una tarea que se encuentre registrada dentro del sistema.
    - **Cambiar estado de una tarea:** Se debe poder cambiar el estado de una tarea registrada entre Completada y Pendiente.
- Las tareas deben contar con la siguiente información:
    - Titulo
    - Descripción
    - Creador (Se podrá agregar el nombre manualmente con un Input)
    - Estado (Completada/Pendiente)
- Se requiere que la aplicación cuente con:
    - Almacenamiento: Se podrá usar un método de almacenamiento que permita la persistencia de datos
        - Puede ser SQL o NoSQL
        - Se recomienda usar una base de datos Sqlite dado que no se requiere de mucha configuración ni despliegue para su uso.
    - API: Se encargará de la interacción entre el frontend y el almacenamiento.
        - Se recomienda que esté hecha con algún framework de NodeJS.
        - Sin embargo se permite que se use cualquier otro framework o lenguaje en el que el aplicante se sienta cómodo.
    - Frontend: Se encargará de la interacción del usuario con el API.
        - Se requiere que sea desarrollada Angular >= 8.0.
        - Debe contar con enrutamiento para la interacción entre las distintas vistas.
- El código fuente debe ser subido a un repositorio público de Github con dos carpetas, una para el API y otra para el Frontend.
- Se tendrá un tiempo máximo de 2 días para completar la prueba, por tanto no se prestará tanta atención a los elementos gráficos.

Observaciones:

- No se requiere que se realicen todos los puntos. Si no puede resolver alguno pase al siguiente.
- Cada punto deberá ser guardado en un archivo en formato .js o .ts (según haya escogido el lenguaje) y tendrá que contener comentarios de cada paso que se está realizando.
- El código fuente debe ser subido a un repositorio público de Github, para ello deberá usar git como sistema de control de versiones.
- Se tendrá un tiempo máximo de 2 días para completar la prueba.

Cualquier inquietud o duda será respondida lo antes posible.

Elaboró:
Edward Camilo Carrillo
Software Engineer