**Prueba Técnica Desarrollador Front**

1. **Introducción**

El propósito de la prueba es el evaluar las habilidades técnicas y capacidades del desarrollador para ser capaz de analizar un problema y poder darle una solución efectiva. Para esto, se busca analizar dos componentes, desarrollo front end en Angular y una prueba de lógica.

Tómese el tiempo para leerlo con calma ya que podría omitir algún detalle que afecte su resultado.

1. **Objetivos**

Se deben realizar los siguientes ejercicios, deben ser los dos no solo uno.

* 1. **Desarrollo Angular**

En el comprimido hay un proyecto de Angular como base que contiene lo siguiente:

* Login (No hace nada)
* Sidebar
* Header
* Content area (sección donde aparece el contenido y puede scrollear, para que el sidebar y header siempre sean fijos)
* Un dashboard sencillo
* Un servicio que puebla el dashboard
* Un servicio básico de autenticación

1. El objetivo de la prueba es implementar una API y ejecutar un CRUD sobre la misma, puede seleccionar cualquier API a gusto, si la API seleccionada no expone ningún método POST/PUT, puede hacer una combinación entre la API y localstorage, como considere mejor. En el siguiente link hay unas recomendaciones de API’s que podrían ser útiles (https://github.com/public-apis/public-apis).
2. Se debe crear una nueva sección para realizar las operaciones CRUD de la API que va a implementar, se adjuntan 2 imágenes (Listar.png y Agregar-Editar.png) que son 2 diseños de referencia de otro proyecto para crear la sección. Como se da cuenta, la temática y colores del diseño no coincide exactamente con el proyecto que fue enviado, ya que son proyectos reales diferentes, entonces, el objetivo es que pueda ajustar el diseño enviado (más que nada colores y detalles menores) para que lo pueda desarrollar tanto móvil como escritorio.  
     
   Se espera que la sección tenga lo siguiente:
   1. Tabla donde se pueda ver mediante una MatTable con un paginador funcionando (angular material tiene un paginador)
   2. Modal donde se pueda agregar/editar
   3. Opción para borrar los registros
   4. Estilos acordes a los colores de la página
3. Habilitar el funcionamiento del login y protección de rutas del proyecto, tanto en las rutas que existían como en las nuevas, para esto, basta con tener un usuario quemado en el servicio para verificar que funcione
4. Ajustar el dashboard para que muestre información relacionada al CRUD creado.
   1. **Prueba lógica**

Junto a este documento, va a encontrar un archivo Excel en formato csv llamado “time\_series\_covid19\_deaths\_US.csv” que muestra el acumulado de muertes por COVID-19 por cada ciudad y estado de US agrupados por fechas.

Desarrollar un script que reciba como input el archivo y como output de los resultado de las preguntas:

1. Estado con mayor acumulado a la fecha
2. Estado con menor acumulado a la fecha
3. El porcentaje de muertes vs el total de población por estado
4. Cual fue el estado más afectado (explicar por qué)

Dar una breve explicación de los resultados.

Tener en cuenta lo siguiente:

1. El valor de cada fecha no son las muertes del día, es el acumulado de hasta la fecha, por consiguiente, cada fecha nueva siempre va a ser igual o mayor a la anterior.
2. Tener presente que cada fila representa una ciudad/pueblo, no el estado, entonces se deben agrupar los datos por estado para obtener las respuestas a las preguntas.
3. Revisar el formato de las fechas de las columnas
4. Realizar el ejercicio en Typescript
5. Una vez completo, poner dentro de una carpeta llamada “Ejercicio lógica” el código fuente y una captura o log de los resultados que obtuvo junto con su explicación.
6. **Requisitos**

Se necesita que se cumplan las siguientes condiciones:

1. **No** se puede usar boostrap, materialize o cualquier framework para manejar el responsive/adaptative ya que se espera que la mayor parte del código sea scss/sass o en su defecto, css puro (no tan deseable), se sugiere utilizar Flexbox
2. Utilizar angular material (no es lo mismo que materialze) para utilizar una tabla, un MatTable, no se recomienda utilizar más ya que no permitiría ver las capacidades de estilos.
3. Utilizar Angular 11+
4. Diseño responsive
5. Buenas prácticas y código limpio
6. Utilizar un repositorio público e ir haciendo commits regularmente ya que eso se revisará
7. Código se debe desarrollar en inglés
8. Actualizar el readme del proyecto original
9. Reutilice código en lo posible (No hablando de copiar y pegar, tampoco forzar la reutilización)
10. Mobile first
11. **Finalización**

Una vez finalizado el desarrollo de angular y la prueba de lógica, enviar al correo [sayala@smartsoftlabs.com](mailto:sayala@smartsoftlabs.com) con el asunto “Entrega prueba técnica Convocatoria” indicando que ya finalizó con el link del repositorio utilizado y poner en el cuerpo del correo dando a entender quién es para diferenciar al candidato.

En caso de dudas, se puede comunicar al correo utilizado para enviar la prueba, o al siguiente en caso de que sean diferentes:

Correo: [sayala@smartsoftlabs.com](mailto:sayala@smartsoftlabs.com)

Teléfono: 313 801 2728

**Nota:** En caso de pasar el tiempo definido para la prueba y no estar lista, puede comunicarse para explicar y solicitar más tiempo si es necesario.

El material proporcionado para esta prueba únicamente puede ser usado para el desarrollo de la misma, cualquier otro uso no está autorizado.