|  |  |  |  |
| --- | --- | --- | --- |
| Fecha: |  | | |
| Nombre: |  | | |
| Hora Inicial: |  | Hora Final: |  |

# Evaluación Técnica – Desarrolladores Frontend - Backend

Dependiendo del proceso de selección al que este aplicando usted deberá desarrollar los puntos de la siguiente manera:

1. FRONT END: Sí aspira a trabajar con nuestra organización como desarrollador de frontend
2. BACKEND: Sí aspira a trabajar con nuestra organización como desarrollador de backend
3. AMBAS PARTES: Sí usted desea que le realicemos asignaciones de backend y frontend

## FRONTEND

Este punto debe ser desarrollado preferiblemente en Angular 2 o superior. Sí Angular no es su fuerte, puede desarrollar el punto en cualquier otro Framework(React, Vue.js, Ember.js, Meteor, Node.js …) tenga en cuenta que sí desarrolla en otro Framework y resulta elegido deberá realizar el curso de Angular entregado por nuestra empresa.

Elaborar una aplicación para la administración de un maestro de productos que cumpla con el caso de uso descrito a continuación:

Funcionalmente se requiere

1. Crear un home básico con
2. Menú
3. Barra superior
4. Sección de perfil (Foto, Nombre, Opción de Log in, Opción de Logout)
5. Crear un formulario de registro de productos con los siguientes campos:
6. ID
7. Nombre del Producto
8. Características (campo tipo texto)
9. Fecha de lanzamiento
10. Correo fabricante
11. País de fabricación
12. Usar el siguiente servicio (<https://restcountries.eu/#api-endpoints-response-example> ) u otro en desde internet

1. Precio en formato de moneda
2. Unidades disponibles
3. Unidades vendidas
4. Imagen del producto
5. El formulario debe tener las siguientes acciones
6. Guardar
7. Cerrar
8. Se debe crear una vista de consulta de productos con los campos del formulario
9. La vista debe permitir las siguientes acciones:
10. Crear un producto (ir al formulario)
11. Eliminar un producto
12. Filtrar por 2 campos del informe
13. Ordenar por 2 columnas del informe
14. Paginar los registros

El modelo de datos puede ser creado en la herramienta que prefiera. Cree por lo menos 20 referencias de productos para verificar la funcionalidad de la vista.

## BACKEND

Este punto debe ser desarrollado preferiblemente en .Net, aunque también puede utilizar cualquier tecnología de backend.

Requerimos que usted realice una API que ejecute un CRUD contra un servicio de almacenamiento NoSQL. Le entregaremos todos los datos de “Azure Storage” sin embargo también le permitimos usar cualquier otro servicio como Google BigQuery ó Amazon S3.

Datos de Azure Storage

 A partir de las siguientes tablas de datos creadas en Table Storage de Azure

 Storage account name

123459876

Key1

X9XmzmV0xX6AG/WYCXSA8BrrrCF0M4j3X4zFDgJZ5g9hRwPafsIjAWk2kgjLZmj6YSxJGIikJDsLmvxgEfGWTQ==

Connection string

DefaultEndpointsProtocol=https;AccountName=123459876;AccountKey=X9XmzmV0xX6AG/WYCXSA8BrrrCF0M4j3X4zFDgJZ5g9hRwPafsIjAWk2kgjLZmj6YSxJGIikJDsLmvxgEfGWTQ==;EndpointSuffix=core.windows.net

Key2

/2JJB7wO6/JLbXmn7kDmSgC8DxZJAZII3sKdy8oR/YrykkBmMvuZ2aJ4HgBfG9hyyIAV4yANXa3o0xWxBksCRw==

Connection string

DefaultEndpointsProtocol=https;AccountName=123459876;AccountKey=/2JJB7wO6/JLbXmn7kDmSgC8DxZJAZII3sKdy8oR/YrykkBmMvuZ2aJ4HgBfG9hyyIAV4yANXa3o0xWxBksCRw==;EndpointSuffix=core.windows.net

*BlobEndpoint=https://123459876.blob.core.windows.net/;QueueEndpoint=https://123459876.queue.core.windows.net/;FileEndpoint=https://123459876.file.core.windows.net/;TableEndpoint=https://123459876.table.core.windows.net/;SharedAccessSignature=sv=2017-11-09&ss=bfqt&srt=sco&sp=rwdlacup&se=2018-06-16T08:08:59Z&st=2018-06-16T00:08:59Z&spr=https&sig=Aejde534rfvWYOR9Gt82XkmmaePSZWYCsf7XADSA5e8%3D*

*SAS token*

*?sv=2017-11-09&ss=bfqt&srt=sco&sp=rwdlacup&se=2018-06-16T08:08:59Z&st=2018-06-16T00:08:59Z&spr=https&sig=Aejde534rfvWYOR9Gt82XkmmaePSZWYCsf7XADSA5e8%3D*

*blob service SAS URL*

*https://123459876.blob.core.windows.net/?sv=2017-11-09&ss=bfqt&srt=sco&sp=rwdlacup&se=2018-06-16T08:08:59Z&st=2018-06-16T00:08:59Z&spr=https&sig=Aejde534rfvWYOR9Gt82XkmmaePSZWYCsf7XADSA5e8%3D*

*Table service SAS URL*

*https://123459876.table.core.windows.net/?sv=2017-11-09&ss=bfqt&srt=sco&sp=rwdlacup&se=2018-06-16T08:08:59Z&st=2018-06-16T00:08:59Z&spr=https&sig=Aejde534rfvWYOR9Gt82XkmmaePSZWYCsf7XADSA5e8%3D*

Requerimiento:

1. Crear una API en Visual Studio .NET que permita realizar un CRUD.
2. La operación de consulta del CRUD deberá permitir paginación
3. Crear un Job que se ejecute cada minuto y actualice la hora de revisión de todos los productos en estado de revisión.

La documentación de Table Storage puede ser encontrada en

<https://docs.microsoft.com/es-es/rest/api/storageservices/table-service-rest-api#rest-api-operations>