**Descripción sistema TechnicalExamT3**

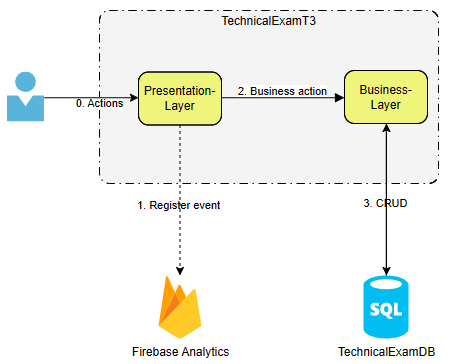
# Objetivo

El siguiente documento busca explicar la funcionalidad del sistema software TechnicalExamT3 desde las características de su arquitectura, hasta su modo de uso y dependencias relacionadas a la necesidad de cumplir con los requisitos funcionales solicitados en el documento “020523-prueba tecnica”.

# Arquitectura

El sistema software se desarrolla como una aplicación web hecha con la librería de .net 6 y lenguage C#. Cumple con arquitectura de n-capas y hace uso de MVC con Razor, y javascript en minima medida. De forma externa hace uso de base de datos tipo SQL server y del registro de datos de eventos en Firebase.

A continuación se muestra más a detalle la arquitectura desarrollada:



De forma simplificada, el proceso que cumple la arquitectura es recibir dependiendo la acción del usuario, la información requerida para posterior a la validación hecha por capa de presentación las siguientes pasos:

**Register event:** Se registra de forma asincrona en Firebase analytics la información acerca de la acción solicitada por el usuario en forma de evento.

**Business action:** La capa de presentación invoca una acción correspondiente en la capa de negocio.

**CRUD:** Desde la capa de negocio se accede a los datos de BD (como lo sugiere lo solicitada en documentación “020523-prueba tecnica”) para realizar operaciones de lectura y/o escritura, ya sea haciendo uso de EF (Entity framework) o de ejecución de procedimientos almacenados en la misma BD.

# Dependencias

El servicio hace uso de las siguientes librerías principalmente:

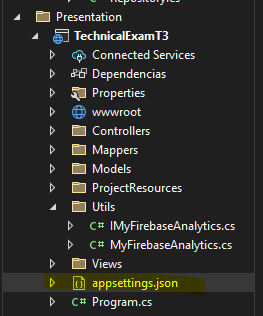
|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **Tipo** | **Version** | **Función** |
| Entity framework | Libreria | 7.0.1 | Facilitar el acceso a la base de datos y operaciones sobre la misma. |
| ClosedXML | Libreria | 0.102.1 | Generación de archivos (excel) |
| X.PagedList.Mvc.Core | Libreria | 9.0.0 | Generación de paginación |
| Sql server | Base de datos |  | Registro de datos y transacciones |
| Firebase | Cloud App |  | Almacenamiento de datos de logs y analitica |

# Configuración

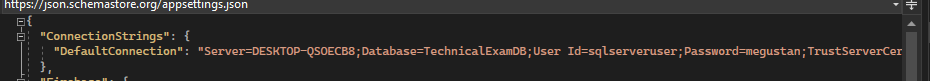
**Aplicación**

Para el correcto funcionamiento de la aplicacion, se debe tener en cuenta los valores configurados en el archivo appsettings.json

Se puede acceder a este para casos de desarrollo en la siguiente ubicación:



Dentro del archivo se halla la configuración a conexion a BD de sqlserver en éste caso:



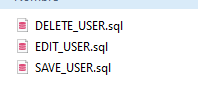
**Base de datos**

Antes de hacer uso de las funcionalidades se debe recalcar la importancia de la ejecutar en base de datos los siguientes scripts en el orden especificado:

Creación de tabla usuario:



Creación de procedimientos almacenados:

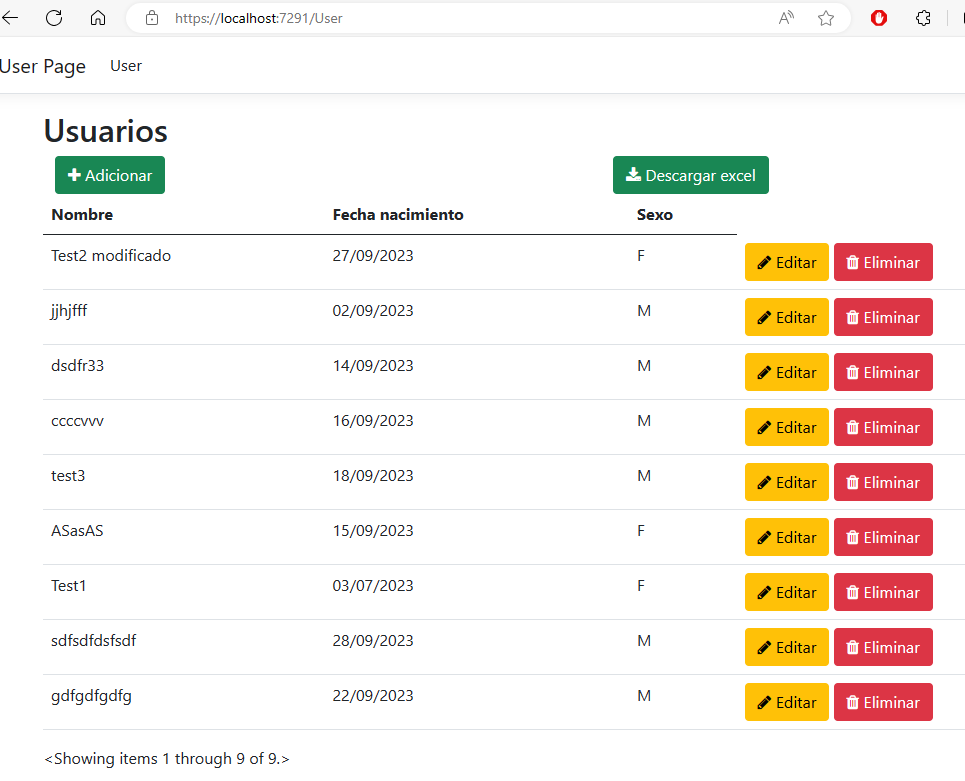


Los archivos pueden hallarse en carpeta “Scripts Base de datos” del repositorio:

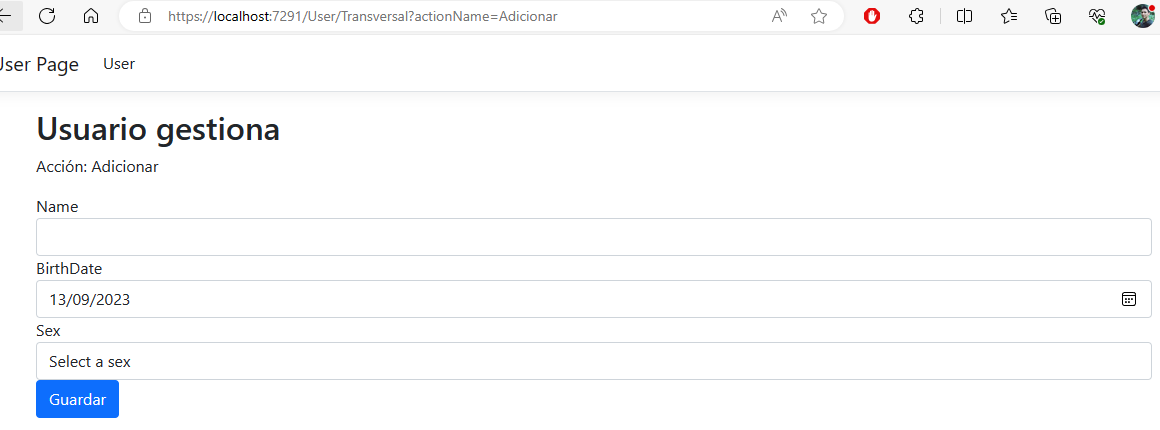
# Demostraciones

**Vistas**

Listado de usuarios:



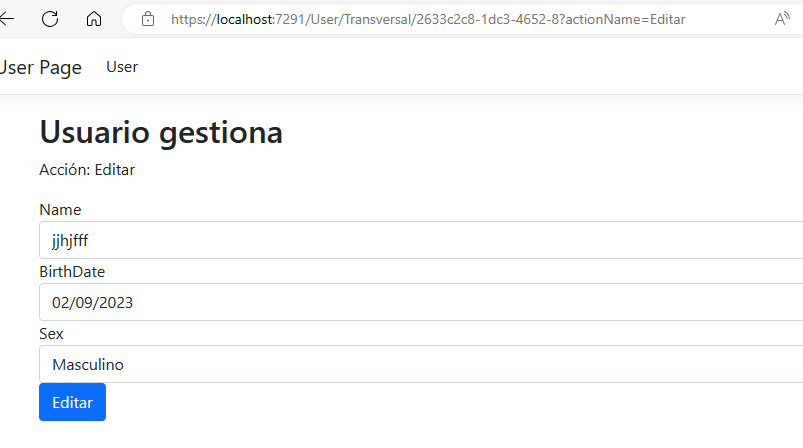
Adicionar usuario:



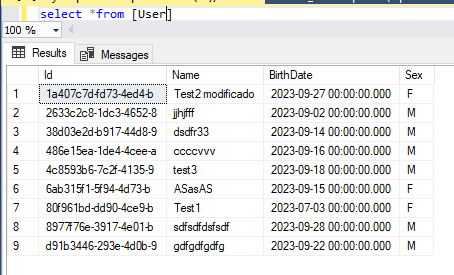
Eliminar usuario:



Editar usuario:



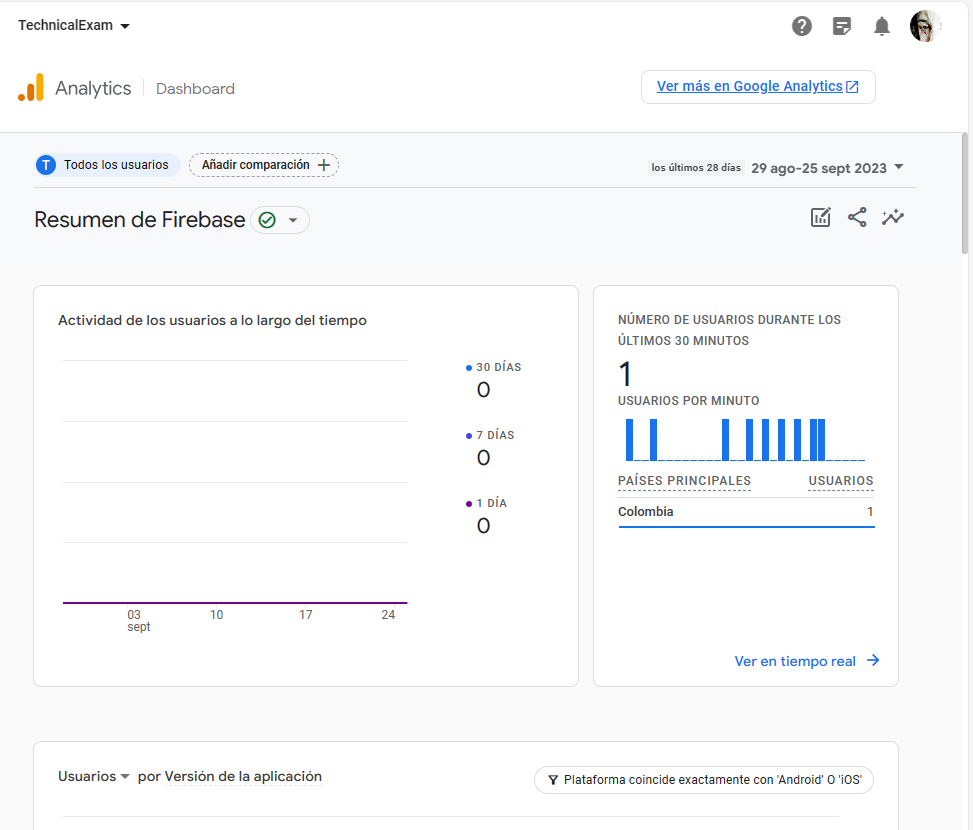
**Tablas**



# Adicional

Cabe recalcar que en el desarrollo de este sistema se buscó la simplicidad y y tratar de abarcar lo solicitado en la documentación “020523-prueba tecnica” aunque la misma no era muy explicita en cuanto a que herramientas, frameworks, versiones y documentación utilizar para el mismo.

Con respecto a Firebase api no se pudo conseguir documentación clara y concisa sobre como consumir la funcionalidad de Api para Analytics directamente desde .net, sin embargo, se consiguió sobre como conectar con javascript donde se pudo conseguirla conexión pero no la generación de eventos.



Por lo anterior, el proyecto se entrega sin dicha funcionalidad hasta el momento.

No se entrega funcionalidad de consulta de datos de Usuarios utilizando SP en BD, ya que no se considera buena practica y además es redundante, sin embargo, aqui se deja planteado como seria un script para tal caso:

