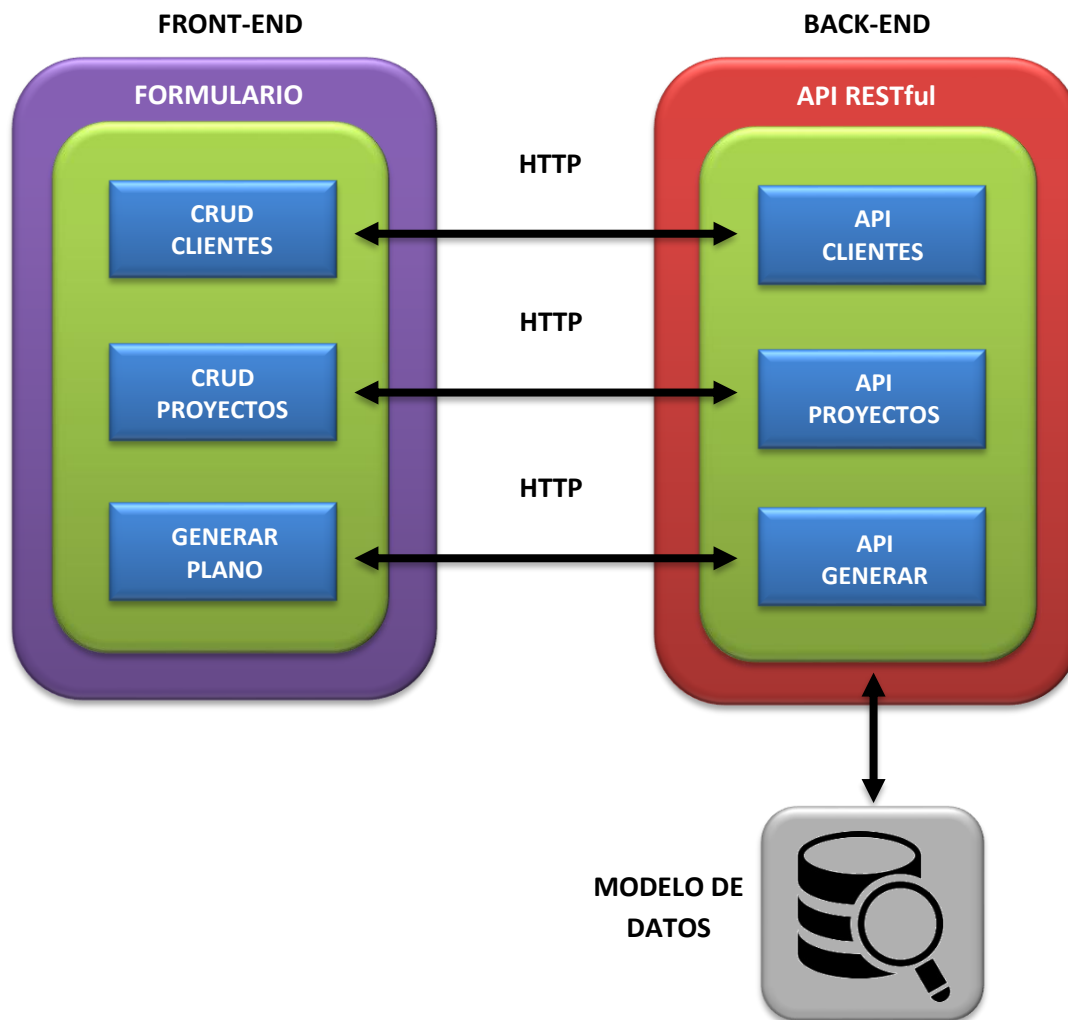


<b>OLSoftware</b> Soluciones Tecnológicas Efectivas	<b>PRUEBA PRACTICA .NET</b>		
	CODIGO: FM-GH-71	VERSIÓN: 01	PAGINA: 1 de 4
	F. EMISION: 02-02-2018	F. ACTUALIZACION: N/A	



#### Consideraciones:

- La arquitectura se encuentra dividida en tres componentes: Front-End (Mvc o Razor Pages), Back-End (Api RestFul) y Modelo de Datos (SQL Server).
- La Base de Datos a utilizar debe ser **SQL Server 2019 Express Edition**.
- La construcción del modelo de datos es libre. Proponer el modelo de datos que considere apropiado para los objetivos del taller.
- Utilizar **Entity Framework Core y Expresiones Lambda**.
- Utilizar **T-SQL** para la construcción de los procedimientos almacenados encargados de realizar las sentencias DML en la Base de datos.
- Se debe utilizar **.NET Core**
- Las peticiones al servidor deben realizarse por el método **HTTP** correspondiente **GET, POST, PUT o DELETE** de acuerdo a la funcionalidad del **CRUD** que se desea implementar.

Se requiere realizar un aplicativo web para la gestión de los proyectos la empresa **OLSoftware S.A.S**, de tal manera que logremos construir un banco de información de los proyectos que se tienen por cada cliente. Por lo tanto, se requiere que el aplicativo web cumpla con las siguientes funcionalidades y condiciones:

### MODELO DE DATOS

Se requiere construir un modelo de datos que permita soportar y persistir las siguientes necesidades:



- Información del cliente: datos básicos, datos de contacto, proyectos con su respectivo estado (En Negociación, En Proceso, Terminado o Anulado).
- Información del proyecto: Nombre del proyecto, cliente, fecha de inicio, fecha de finalización (No puede ser menor a la fecha de inicio), precio, cantidad de horas y los diferentes lenguajes de programación que requiere con su respectivo nivel (Alto, Medio y Bajo).
- Construir un procedimiento almacenado (Stored Procedure) para obtener la información de los proyectos (Nombre del cliente, Telefono, Nombre del proyecto, fecha de inicio, fecha de finalización, precio, cantidad de horas y estado).
- Crear datos semilla: 10 Clientes y 20 proyectos con los lenguajes de programación que requieren.

### BACK-END

#### BACKEND>

- Construir API's RESTful que implementen los servicios CRUD de la aplicación (Clientes y Proyectos).
- Construir un Microservice para la generación y exportación de un archivo plano con la información de los proyectos por cada cliente, utilizando la información que retorna el procedimiento almacenado creado en el punto anterior. El Archivo debe quedar alojado en una carpeta dentro de la aplicación.

### FRONT-END



- Utilizar ASP.Net Core Identity para gestionar los usuarios de la aplicación.
- Construir una interfaz que permita gestionar los clientes consumiendo las API RESTful (CRUD) respectivas.
- Construir una interfaz que permita gestionar los proyectos por cada cliente consumiendo las API RESTful (CRUD) respectivas.
- Construir una opción o funcionalidad que permita invocar (Consumir) el API RESTful para la generación del archivo plano.