



TEMA: MVC

SUBTEMA: Web API

EJERCICIO – Crear una Web API y Consumirla desde el Servidor y desde el Cliente

OBJETIVO

Crear y Consumir una Web API para la actualización de la tabla EstatusAlumnos, desde el Servidor utilizando httpClient y desde el Cliente utilizando AJAX

DESARROLLO

Crear una Web API con los métodos CRUD para la actualización de la tabla EstatusAlumnos. Asimismo, agregar a la aplicación MVC el controlador y vistas correspondientes para que el usuario pueda realizar las actualizaciones a dicha tabla, quedando la siguiente interfaz.

Index				
Crear				
Clave	Nombre			
PTO	Prospecto	Actualizar	Consultar	Eliminar Eliminar Ajax
PRO	En curso propedéutico	Actualizar	Consultar	Eliminar Eliminar Ajax
CAP	En capacitación	Actualizar	Consultar	Eliminar Eliminar Ajax
INC	En Incurción	Actualizar	Consultar	Eliminar Eliminar Ajax
LAB	Laborando	Actualizar	Consultar	Eliminar Eliminar Ajax
LIB	Liberado	Actualizar	Consultar	Eliminar Eliminar Ajax
NI	No le interesó	Actualizar	Consultar	Eliminar Eliminar Ajax
BA	Baja	Actualizar	Consultar	Eliminar Eliminar Ajax
Sus	Suspendido	Actualizar	Consultar	Eliminar Eliminar Ajax
con	Congelado	Actualizar	Consultar	Eliminar Eliminar Ajax
Jur	En Jurídico	Actualizar	Consultar	Eliminar Eliminar Ajax



1 Crear Web API.

1.1 Crear Aplicación tipo Servicio ASP .Net Core Web API.

Crear una nueva aplicación de tipo “ASP .Net Core Web API” con el nombre de **WebAPIEstatusAlumnos**, para el Framework .Net 5.0

1.2 Agregar EntityFramework

Agregar EntityFramework para la tabla EstatusAlumnos

1.2.1 Instalar los siguientes paquetes NuGet

[Microsoft.EntityFrameworkCore](#)

[Microsoft.EntityFrameworkCore.Design](#)

[Microsoft.EntityFrameworkCore.SqlServer](#)

[Microsoft.EntityFrameworkCore.Tools](#)

1.2.2 Ejecutar en la consola del administrador de paquetes el comando Scaffold-DbContext

Ejecutar en la consola del administrador de paquetes el comando Scaffold-DbContext, con la conexión y proveedor correspondiente a su base de datos, únicamente para la tabla EstatusAlumnos, la clase de contexto se debe llamar EstatusContext y debe quedar en la carpeta Models/Context, asimismo las entidades deben de quedar en la carpeta Models/Entidades

1.2.3 Configurar la cadena de conexión en el archivo appsetting.json

1.2.4 Configurar servicio del DbContext en el archivo Startup.cs

1.3 Agregar el Controlador

Agregar el controlador de API con acciones que usan EntityFramework llamado EstatusAlumnosController

2 Probar el Web API

Probar cada uno de los métodos del WebAPIEstatusAlumnos a través del PostMan recordando que para ello deberán usar los verbos GET, PUT, POST, DELETE

3 Consumir WebAPI

3.1 Consumir el WebAPI desde el Servidor con HttpClient

3.1.1 En el proyecto de Negocio agregar la clase NEstadoAlumnos

Crear la clase NEstadoAlumnos con los métodos de acción CRUD para la tabla de EstadoAlumnos

- a) List<EstadoAlumnos> Consultar()
- b) Void EstadoAlumnos Consultar(int id)
- c) EstadoAlumnos Agregar(EstadoAlumnos)
- d) Void Actualizar(EstadoAlumnos)
- e) Void Eliminar(int id)

Implementandolos accedendo el Web API WebAPIEstadoAlumnos

3.1.2 En el proyecto de Presentacion agregar Controlador EstadoAlumnos

Crear en la carpeta Controllers del Proyecto de Presentación, el controlador EstadoAlumnosController con la plantilla Controlador de MVC 5 con acciones de lectura y escritura

Implementar cada uno de los métodos utilizando la clase NEstadoAlumnos

3.1.3 Crear las vistas correspondientes a cada uno de los métodos de acción agregados al Controlador EstadoAlumnos

Crear las vistas correspondientes a cada uno de los métodos de acción agregados al Controlador EstadoAlumnos, dando la apariencia de la pantalla indicada, sin el botón "Eliminar Ajax", la cual se agregará posteriormente.


3.2 Consumir el WebAPI desde el Cliente con AJAX

3.2.1 Agregar el botón Eliminar AJAX

En la vista Index, agregar el botón **Eliminar AJAX**, mismo que deberá responder desde el cliente a fin de que mediante AJAX consulte el Estado seleccionado, a través del WebAPIEstadoAlumnos y muestre la información en una ventana modal. La ventana modal deberá tener un botón para confirmar la eliminación del registro mostrado, y en su caso, proceder a eliminarlo mediante el llamado a la WebAPIEstadoAlumnos.

3.2.2 Agregar la ventana Modal para eliminar

Agregar la ventana Modal para eliminar, como la que se muestra en la figura



TI-Capital Humano

Estado

id	2
Clave	PRO
Nombre	En curso propedéutico

Close Eliminar

3.2.3 Agregar la función javascript para Consultar

Agregar la función javascript que se ejecutará cuando se dé click al botón “Eliminar AJAX”, misma que deberá llamar al WebAPIEstatusAlumnos por medio de AJAX, para consultar el elemento seleccionado y en la respuesta obtenida llenar la ventana modal y mostrarla.

3.2.4 Agregar la función javascript para Eliminar

Agregar la función javascript que se ejecutará cuando se dé click al botón “Eliminar”, de la Ventana Modal, misma que deberá llamar al WebAPIEstatusAlumnos por medio de AJAX, para Eliminar el elemento mostrado.

3.2.5 Adaptar el Web API WebAPIEstatusAlumnos para permitir Intercambio de recursos de origen cruzado (CORS)

Adaptar el Web API WebAPIEstatusAlumnos para permitir Intercambio de recursos de origen cruzado, para ello:

- Instalar el paquete Nuget **Microsoft.AspNetCore.Cors**
- Agregar los servicios CORS en el método **ConfigureServices** del **startup.cs**,
- Agregar el uso de los CORS en el método **Configure** de su **startup.cs**, indicando la url desde la cual se le permitirá hacerle llamadas.