



Impuestos Internos

Dirección General de Impuestos Internos

Nombre del Sistema

Sistema de Contribuyentes

Responsable

Flavio José Robles

Tipo de manual

Técnico

Fecha de Elaboración

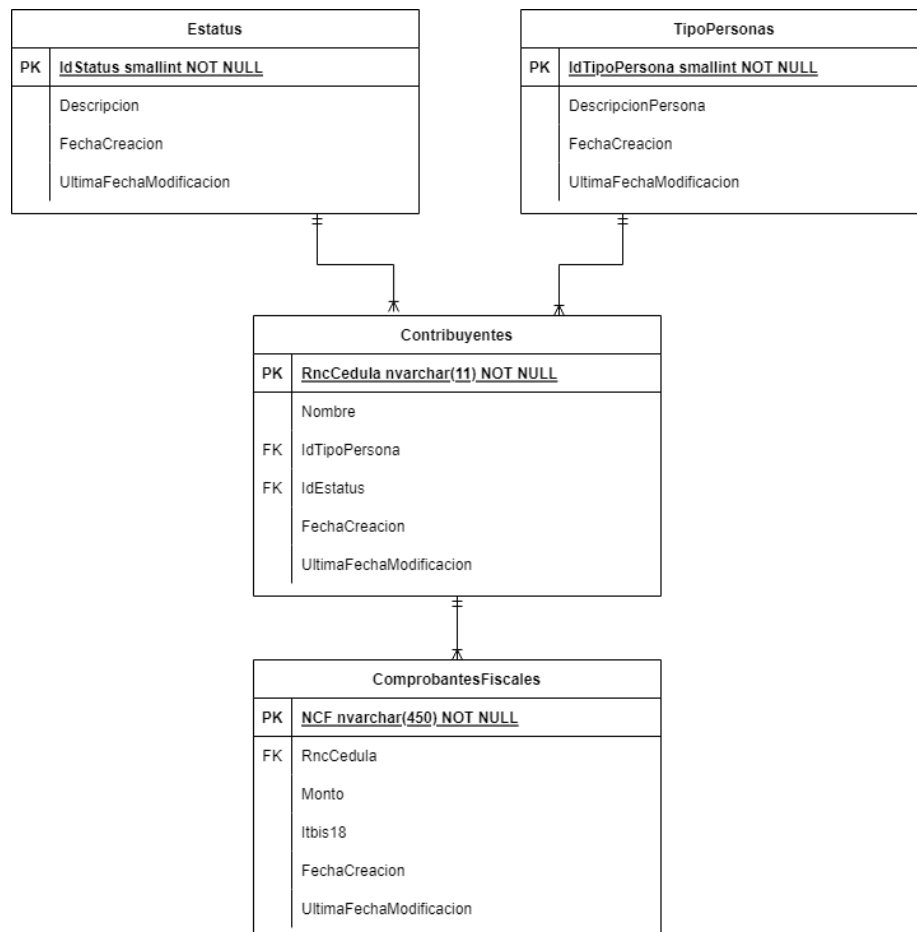
18 de mayo del año 2023

Introducción

El objetivo de esta documentación es proporcionar detalles técnicos de la API creada para que la persona que desee usarla tenga una idea clara de la estructura completa de la misma. Los principales métodos de esta API son dos: uno desde el cual se pueda obtener el listado de todos los contribuyentes y otro método con el que se pueda obtener el listado de todos los comprobantes fiscales. Además de esto, esta devuelve los resultados en formato JSON.

A continuación, se muestra un diagrama de entidad relación que utilizó como base para la construcción del API:

Ilustración 1: Diagrama de Entidad Relación



El Diagrama Entidad-Relación ilustra las relaciones entre las entidades del sistema. Muestra cómo las entidades *Contribuyentes* y *Comprobantes* se relacionan con otras entidades de la base de datos, proporcionando una representación visual de la estructura de datos.

Puertos

La aplicación web, que es la otra parte de esta prueba, está configurada para correr en el puerto **7092** (**https://localhost:7092**), si por alguna razón se necesita correr en otro puerto, se debe de cambiar el puerto en el contenedor de servicios del API para que sea posible ejecutar peticiones desde (la aplicación web) el nuevo puerto. A continuación, se muestra el código en donde realizar dicho cambio:

Ilustración 2: Program.cs - Líneas: 28-35

```
//CORS
builder.Services.AddCors(opts =>
{
    opts.AddPolicy("MyCORSPolicy", policy =>
    {
        policy.AllowAnyOrigin();
        policy.WithOrigins("https://localhost:7092");
    });
});
```

Endpoints

Controlador de Contribuyentes

El controlador de Contribuyentes maneja las operaciones relacionadas con los contribuyentes.

Ruta base: /api/Contribuyentes

Métodos:

1. Obtener todos los registros:

- Método: **GET**

- Ruta: /
- Descripción: Retorna todos los contribuyentes.
- Parámetros de entrada: Ninguno.
- Respuesta exitosa: Código 200 OK y una lista de objetos *ContribuyenteDTO*.
- Respuesta de error: Código 404 Not Found si no se encuentran registros.

2. Obtener un registro por ID:

- Método: **GET**
- Ruta: **/id**
- Descripción: Retorna un contribuyente específico según su ID (RNC/CEDULA).
- Parámetros de entrada: ID del contribuyente.
- Respuesta exitosa: Código 200 OK y un objeto Contribuyente.
- Respuesta de error: Código 404 Not Found si no se encuentra el contribuyente.

3. Crear un nuevo registro:

- Método: **POST**
- Ruta: /
- Descripción: Crea un nuevo contribuyente.
- Parámetros de entrada: Objeto Contribuyente en el cuerpo de la solicitud.
- Respuesta exitosa: Código 201 Created y el objeto Contribuyente creado.
- Respuesta de error: Código 400 Bad Request si la solicitud es inválida o si el número de RNC/CEDULA ya existe. Código 500 Internal Server Error si ocurre un error en el servidor.

4. Actualizar un registro existente:

- Método: **PUT**
- Ruta: **/id**
- Descripción: Actualiza los datos de un contribuyente existente.
- Parámetros de entrada: ID del contribuyente y objeto Contribuyente actualizado en el cuerpo de la solicitud.
- Respuesta exitosa: Código 200 OK y el objeto Contribuyente actualizado.
- Respuesta de error: Código 400 Bad Request si la solicitud es inválida, si el número de RNC/CEDULA no puede ser modificado o si el contribuyente no se encuentra. Código 500 Internal Server Error si ocurre un error en el servidor.

5. Eliminar un registro:

- Método: **DELETE**
- Ruta: **/id**
- Descripción: Elimina un contribuyente existente.
- Parámetros de entrada: ID del contribuyente a eliminar.
- Respuesta exitosa: Código 204 No Content si se elimina correctamente.
- Respuesta de error: Código 404 Not Found si el contribuyente no se encuentra.

Controlador de Comprobantes Fiscales

El controlador de Comprobantes Fiscales maneja las operaciones relacionadas con los comprobantes fiscales.

Ruta base: /api/ComprobantesFiscales

Métodos:

1. Obtener todos los registros:

- Método: **GET**
- Ruta: **/**
- Descripción: Retorna todos los comprobantes fiscales.
- Parámetros de entrada: Ninguno.
- Respuesta exitosa: Código 200 OK y una lista de objetos ComprobanteFiscal.
- Respuesta de error: Código 404 Not Found si no se encuentran registros.

2. Obtener un registro por ID:

- Método: **GET**
- Ruta: **/id**
- Descripción: Retorna un comprobante fiscal específico según su ID.
- Parámetros de entrada: ID del comprobante fiscal.
- Respuesta exitosa: Código 200 OK y un objeto ComprobanteFiscal.
- Respuesta de error: Código 404 Not Found si no se encuentra el comprobante fiscal.

3. Crear un nuevo registro:

- Método: **POST**
- Ruta: /
- Descripción: Crea un nuevo comprobante fiscal.
- Parámetros de entrada: Objeto ComprobanteDTO en el cuerpo de la solicitud.
- Respuesta exitosa: Código 201 Created y el objeto ComprobanteFiscal creado.
- Respuesta de error: Código 400 Bad Request si la solicitud es inválida, si el monto es menor o igual a cero, si el RNC/CEDULA no existe o si el NCF ya existe. Código 500 Internal Server Error si ocurre un error en el servidor.

4. Actualizar un registro existente:

- Método: **PUT**
- Ruta: /{id}
- Descripción: Actualiza los datos de un comprobante fiscal existente.
- Parámetros de entrada: ID del comprobante fiscal y objeto ComprobanteDTO actualizado en el cuerpo de la solicitud.
- Respuesta exitosa: Código 200 OK y el objeto ComprobanteFiscal actualizado.
- Respuesta de error: Código 400 Bad Request si la solicitud es inválida, si el NCF no puede ser modificado o si el monto es menor o igual a cero. Código 404 Not Found si el comprobante fiscal no se encuentra. Código 500 Internal Server Error si ocurre un error en el servidor.

5. Eliminar un registro:

- Método: **DELETE**
- Ruta: /{id}

- Descripción: Elimina un comprobante fiscal existente.
- Parámetros de entrada: ID del comprobante fiscal a eliminar.
- Respuesta exitosa: Código 204 No Content si se elimina correctamente.
- Respuesta de error: Código 404 Not Found si el comprobante fiscal no se encuentra.

Ejemplos de payloads:

Método	Endpoint	Petición	Respuesta
GET	/api/contribuyentes		[{ "rncCedula": "string", "nombre": "string", "tipo": "string", "estatus": "string" }]
GET	/api/ComprobantesFiscales		[{ "rncCedula": "string", "ncf": "string", "monto": 0, "itbis18": 0 }]
POST	/api/ComprobantesFiscales	{ "rncCedula": "string", "ncf": "string", "monto": 0 }	{ "rncCedula": "string", "ncf": "string", "monto": 0, "itbis18": 0 }

POST	/api/contribuyentes	<pre>{ "rncCedula": "string", "nombre": "string", "idTipoPersona": 0, "idEstatus": 0 }</pre>	<pre>{ "rncCedula": "string", "nombre": "string", "idTipoPersona": 0, "idEstatus": 0, "tipo": { "idTipoPersona": 0, "descripcionPersona": "string" }, "estatus": { "idEstatus": 0, "descripcion": "string" }, "comprobantes": [{ "rncCedula": "string", "ncf": "string", "monto": 0, "itbis18": 0 }] }</pre>
------	---------------------	--	--

Tabla 1. Ejemplos de peticiones a la API

Se han implementado también **controladores genéricos** en la estructura de la API. Es decir, se ha implementado una clase base de controlador genérico que contiene operaciones CRUD (Create, Read, Update, Delete) comunes y que puede ser heredada por controladores de entidad específicos (en este caso *Estatus* y *TipoPersonas*). Este enfoque **reduce la duplicación de código** y promueve la coherencia entre los controladores.

Ejemplos de peticiones con los controladores genéricos:

Método	Endpoint	Petición	Respuesta
GET	/api/Estatus		[{ "idEstatus": 0, "descripcion": "string" }]
GET	/api/TipoPersonas		[{ "idTipoPersona": 0, "descripcionPersona": "string" }]
POST	/api/Estatus	{ "descripcion": "string" }	{ "idEstatus": 0, "descripcion": "string" }
POST	/api/TipoPersonas	{ "descripcionPersona": "string" }	{ "idTipoPersona": 0, "descripcionPersona": "string" }

Tabla 2. Ejemplos de peticiones de controladores genéricos

Conclusión

En conclusión, este manual técnico ha proporcionado una descripción exhaustiva de la API y su funcionalidad. Hemos explorado los controladores específicos de la API, incluyendo su estructura, métodos y parámetros de entrada y salida. Además, hemos discutido el uso de controladores genéricos para lograr un enfoque más flexible y reutilizable en la implementación de controladores de entidades.

Al utilizar este manual, los desarrolladores pueden comprender rápidamente cómo interactuar con la API y aprovechar sus capacidades para realizar operaciones CRUD en los recursos disponibles. Se ha proporcionado documentación detallada sobre cada uno de los controladores, incluyendo ejemplos de código y explicaciones claras de su funcionamiento.

Además, se ha incluido un diagrama de relaciones de entidad para proporcionar una visión general de las relaciones entre los recursos en la base de datos subyacente. Esto facilitará la comprensión de cómo los diferentes controladores interactúan entre sí y cómo los datos se relacionan dentro del sistema.