

Manual Técnico Proyecto Fiscalías Ministerio Público

César Eduardo Guerra Gil

Contenido

1. Objetivo	3
2. Finalidad	3
3. Herramientas utilizadas para el desarrollo	3
4. Funcionalidad del Sistema.....	3
5. Modelamiento de Base de Datos	4
5.1 Diagrama.....	4
5.2 Diccionario de Datos.....	4
5.3 Base de Datos	5
6. Desarrollo BackEnd	6
6.1 Configuración WebApi.....	6
6.2 Modelos	7
6.2.1 Clase Fiscalías	7
6.2.2 Clase Teléfonos	7
6.3 Controladores	8
6.3.1 Controlador Fiscalías	8
6.3.1 Controlador Teléfonos	10
6.3.2 Controlador Números	11
6.4 Datos de Conexión.....	12
6.5 Ejecutar API-REST	17
7. Desarrollo de Front-End	17
7.1 Diseño	17
7.2 Desarrollo	18

1. Objetivo

Se crea este manual con el fin de mostrar cómo fue elaborado el sistema, referencias de como interactuar con el programa, código, requisitos, clases, entre otros, para que el mismo pueda ser actualizado de ser requerirlo y así mismo se pueda dar el mantenimiento adecuado en caso de algún fallo.

2. Finalidad

La finalidad de este manual es orientar a la persona que va a administrar, editar o configurar el sistema desarrollado, usando las herramientas necesarias.

3. Herramientas utilizadas para el desarrollo

- Navegador Web: (Google Chrome) para el manejo de la aplicación y pruebas de esta.
- Gestor de base de datos: (SQL SERVER 2022 y SQL Server Management Studio Management Studio 19) para la administración de los registros, procedimientos y vistas.
- Aplicación de entorno de desarrollo compatible con lenguaje C#: (Visual Studio 2019) para la modificación y actualización del BackEnd (API-REST).
- Aplicación de desarrollo de páginas web: (Visual Code) para la actualización o modificación del código fuente del FrontEnd y el diseño.

4. Funcionalidad del Sistema

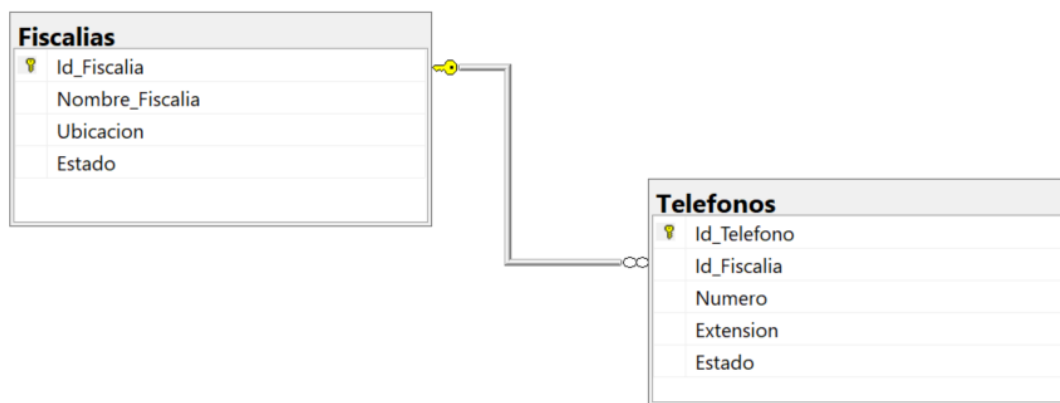
La funcionalidad del sistema es poder mostrar, modificar y eliminar la información de nombre, ubicación y teléfonos de las diversas fiscalías con las que cuenta el Ministerio Público en todo el país.

5. Modelamiento de Base de Datos

5.1 Diagrama

Se detalla el diagrama, el cual está compuesto por la entidades y atributos creados para el almacenamiento de los datos del sistema; cada una de las entidades realiza las siguientes funciones:

- **Fiscalías:** Almacena los datos de la fiscalía que se registran (Nombre, Ubicación y Estado).
- **Teléfonos:** Almacena los datos de números y extensiones que se registran de cada una de las fiscalías agregadas, teniendo una llave foránea por la relación de uno a muchos de Fiscalías a Teléfonos.



5.2 Diccionario de Datos

➤ Modelo Fiscalías

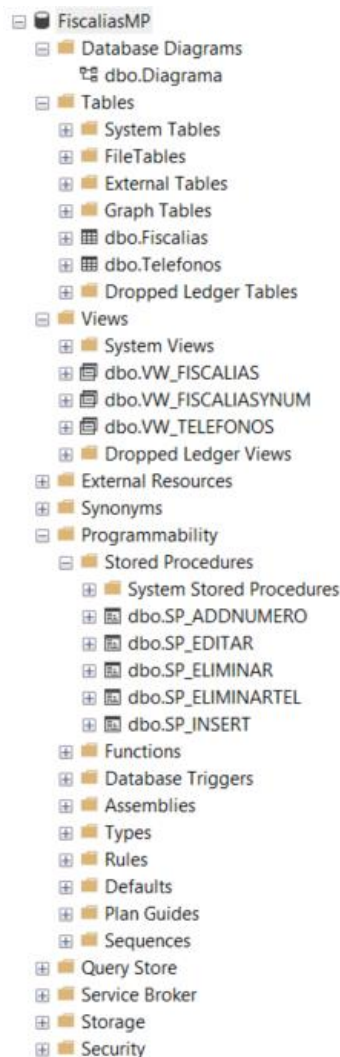
FISCALIAS		
Nombre del Campo	Tipo	Descripción
Id_Fiscalia	int	Llave Primaria (PK)
Nombre_Fiscalia	varchar(255)	
Ubicacion	varchar(255)	
Estado	bit	Campo utilizado para eliminación lógica.

➤ **Modelo Teléfonos**

TELEFONOS		
Nombre del Campo	Tipo	Descripción
Id_Telefono	int	Llave Primaria (PK)
Id_Fiscalia	int	Llave Foránea (FK)
Numero	decimal(8, 0)	
Extension	decimal(5, 0)	
Estado	bit	Campo utilizado para eliminación lógica.

5.3 Base de Datos

La base de datos utilizada tiene por nombre **“FiscaliasMP”**, en la cual, al conectarse, se tendrá acceso al diagrama, tablas, vistas, procedimientos almacenados y demás.



6. Desarrollo BackEnd

En esta sección se realizará una descripción de los aspectos técnicos del Back-End del aplicativo, para la modificación y mejoras de este; se creó una API-REST desarrollada en lenguaje C# utilizando Visual Studio 2019.

6.1 Configuración WebApi

Se configura el archivo WebApiConfig.cs, para que la salida de información de la API sea en formato Json.

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web.Http;
5  using System.Web.Http.Cors;
6
7  namespace ApiFiscalias
8  {
9      1 referencia
10     public static class WebApiConfig
11     {
12         1 referencia
13         public static void Register(HttpConfiguration config)
14         {
15             var cors = new EnableCorsAttribute("*", "*", "*");
16             config.EnableCors(cors);
17
18             // Rutas de API web
19             config.MapHttpAttributeRoutes();
20
21             config.Routes.MapHttpRoute(
22                 name: "DefaultApi",
23                 routeTemplate: "api/{controller}/{id}",
24                 defaults: new { id = RouteParameter.Optional }
25             );
26         }
27     }
```

6.2 Modelos

Dentro de la carpeta modelos se tienen las clases de Fiscalías y Teléfonos, mismas que son necesarias para los controladores y la conexión a base de datos, el nombre de los atributos debe ser igual a nuestra base de datos, en caso de crear una nueva entidad en nuestra base de datos y se quiere agregar a la API, se deben crear las nuevas clases o atributos a las existentes, según corresponda.

6.2.1 Clase Fiscalías

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5
6  namespace ApiFiscalias.Models
7  {
8      14 referencias
9      public class Fiscalias
10     {
11         3 referencias
12         public int Id_Fiscalia { get; set; }
13         0 referencias
14         public int Id_Telefonos { get; set; }
15         4 referencias
16         public string Nombre_Fiscalia { get; set; }
17         4 referencias
18         public string Ubicacion { get; set; }
19         2 referencias
20         public decimal Numero { get; set; }
21         2 referencias
22         public decimal Extension { get; set; }
23     }
24 }
```

6.2.2 Clase Teléfonos

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5
6  namespace ApiFiscalias.Models
7  {
8      13 referencias
9      public class Telefonos
10     {
11         3 referencias
12         public int Id_Fiscalia { get; set; }
13         2 referencias
14         public int Id_Telefonos { get; set; }
15         2 referencias
16         public string Nombre_Fiscalia { get; set; }
17         2 referencias
18         public string Ubicacion { get; set; }
19         3 referencias
20         public decimal Numero { get; set; }
21         3 referencias
22         public decimal Extension { get; set; }
23     }
24 }
```

6.3 Controladores

Las clases controladores servirán para poder hacer el CRUD que ocupara la API-REST, las mismas estarán enlazadas a la clase Conexion.cs.

6.3.1 Controlador Fiscalías

Está acción crea un nuevo registro dentro de la tabla Fiscalía, para lo cual se utiliza el método **POST**.

```
public IActionResult Post(Fiscalias fiscalia) {  
    try  
    {  
        Conexion conFiscalias = new Conexion();  
        bool res = conFiscalias.InsertFiscalias(fiscalia);  
        if (res)  
            return Content(HttpStatusCode.OK, "Guardado correctamente");  
        else  
            return Content(HttpStatusCode.BadRequest, "Ocurrio un error");  
    }  
    catch (Exception ex)  
    {  
        return Content(HttpStatusCode.InternalServerError, "Error Interno en servidor" + ex.Message);  
    }  
}
```

Está acción muestra valores filtrados por Id_Fiscalia de los registros de la tabla Fiscalía, para lo cual se utiliza el método **GET**.

```
public IActionResult Get(int id)  
{  
    try  
    {  
        Conexion con = new Conexion();  
        Fiscalias res = con.GetFiscalias(id);  
        if (res != null)  
            return Content(HttpStatusCode.OK, res);  
        else  
            return Content(HttpStatusCode.BadRequest, "No existen datos");  
    }  
    catch (Exception ex)  
    {  
        return Content(HttpStatusCode.InternalServerError, "Error Interno -->" + ex.Message);  
    }  
}
```


Está acción muestra todos los registros de la tabla Fiscalía, para lo cual se utiliza el método **GET**.

```
public IActionResult Get()
{
    try
    {
        Conexion con = new Conexion();
        List<Fiscalias> resLis = con.GetAllFiscalias();
        if (resLis != null)
            return Content(HttpStatusCode.OK, resLis);
        else
            return Content(HttpStatusCode.BadRequest, "No existen datos");
    }
    catch (Exception ex)
    {
        return Content(HttpStatusCode.InternalServerError, "Error Interno -->" + ex.Message);
    }
}
```

Está acción actualiza o modifica un registro de la tabla Fiscalía, para lo cual se utiliza el método **PUT**.

```
public IActionResult Put(Fiscalias fiscalia)
{
    try
    {
        Conexion conCli = new Conexion();
        bool res = conCli.UpdateFiscalias(fiscalia);
        if (res)
            return Content(HttpStatusCode.OK, "Datos guardados correctamente");
        else
            return Content(HttpStatusCode.BadRequest, "Ocurrio un error");
    }
    catch (Exception ex)
    {
        return Content(HttpStatusCode.InternalServerError, "Error Interno en servidor" + ex.Message);
    }
}
```

Está acción elimina un registro de la tabla Fiscalía, para lo cual se utiliza el método **DELETE**.

```
public IActionResult Delete(int id)
{
    try
    {
        Conexion conCli = new Conexion();
        bool res = conCli.DeleteFiscalias(id);
        if (res)
            return Content(HttpStatusCode.OK, "Datos eliminados correctamente");
        else
            return Content(HttpStatusCode.BadRequest, "Ocurrio un error");
    }
    catch (Exception ex)
    {
        return Content(HttpStatusCode.InternalServerError, "Error Interno en servidor" + ex.Message);
    }
}
```

6.3.2 Controlador Teléfonos

Esta acción crea un nuevo registro dentro de la tabla Teléfonos, para lo cual se utiliza el método **POST**.

```
public class TelefonosController : ApiController
{
    0 referencias
    public IHttpActionResult Post(Telefonos fiscalia)
    {
        try
        {
            Conexion conFiscalias = new Conexion();
            bool res = conFiscalias.InsertTelefonos(fiscalia);
            if (res)
                return Content(HttpStatusCode.OK, "Guardado correctamente");
            else
                return Content(HttpStatusCode.BadRequest, "Ocurrio un error");
        }
        catch (Exception ex)
        {
            return Content(HttpStatusCode.InternalServerError, "Error Interno en servidor" + ex.Message);
        }
    }
}
```

Esta acción muestra valores filtrados por Id_Fiscalia de los registros de la tabla Teléfonos, para lo cual se utiliza el método **GET**.

```
public IHttpActionResult Get(int id)
{
    try
    {
        Conexion con = new Conexion();
        Telefonos res = con.GetTelefonos(id);
        if (res != null)
            return Content(HttpStatusCode.OK, res);
        else
            return Content(HttpStatusCode.BadRequest, "No existen datos");
    }
    catch (Exception ex)
    {
        return Content(HttpStatusCode.InternalServerError, "Error Interno -->" + ex.Message);
    }
}
```

6.3.3 Controlador Números

Se creo este controlador adicional al de teléfonos, con el fin de poder mostrar los registros filtrados por Id_Telefono.

Está acción muestra valores filtrados por Id_Telefono de los registros de la tabla Teléfonos, para lo cual se utiliza el método **GET**.

```
public IActionResult Get(int id)
{
    try
    {
        Conexion con = new Conexion();
        List<Telefonos> resLis = con.GetTelefonosFis(id);
        if (resLis != null)
            return Content(HttpStatusCode.OK, resLis);
        else
            return Content(HttpStatusCode.BadRequest, "No existen datos");
    }
    catch (Exception ex)
    {
        return Content(HttpStatusCode.InternalServerError, "Error Interno -->" + ex.Message);
    }
}
```

Está acción elimina un registro de la tabla Teléfonos, para lo cual se utiliza el método **DELETE**.

```
public IActionResult Delete(int id)
{
    try
    {
        Conexion conCli = new Conexion();
        bool res = conCli.DeleteTelefonos(id);
        if (res)
            return Content(HttpStatusCode.OK, "Datos eliminados correctamente");
        else
            return Content(HttpStatusCode.BadRequest, "Ocurrio un error");
    }
    catch (Exception ex)
    {
        return Content(HttpStatusCode.InternalServerError, "Error Interno en servidor" + ex.Message);
    }
}
```

6.4 Datos de Conexión

En la clase Conexión, se encuentra la conexión y requerimientos a la base de datos, la ruta de base de datos local es:

Data Source= GUERRA; Initial Catalog= FiscalíasMP; User ID=sa; Password= Guerra99"

En la misma, se fueron a llamar los procedimientos almacenados y las vistas de la base de datos FiscalíasMP.

- Insertar Registros en tabla Fiscalías.

```
public bool InsertFiscalias(Fiscalias fiscalia) {
    string conString = "Data Source= GUERRA; Initial Catalog= FiscaliasMP; User ID=sa; Password= Guerra99";
    try
    {
        using (SqlConnection con = new SqlConnection(conString))
        {
            string comando = "EXECUTE SP_INSERT @Nombre_Fiscalia, @Ubicacion, @Numero, @Extension";
            using (SqlCommand cmd = new SqlCommand(comando, con))
            {
                cmd.CommandType = CommandType.Text;
                cmd.Parameters.AddWithValue("@Nombre_Fiscalia", fiscalia.Nombre_Fiscalia);
                cmd.Parameters.AddWithValue("@Ubicacion", fiscalia.Ubicacion);
                cmd.Parameters.AddWithValue("@Numero", fiscalia.Numero);
                cmd.Parameters.AddWithValue("@Extension", fiscalia.Extension);
                con.Open();
                int rowsAffected = cmd.ExecuteNonQuery();
                if (rowsAffected > 0)
                    return true;
                else
                    return false;
            }
        }
    }
    catch (Exception ex)
    {
        return false;
    }
}
```

- Actualizar o modificar Registros en tabla Fiscalías.

```
public bool UpdateFiscalias(Fiscalias fiscalia)
{
    string conString = "Data Source= GUERRA; Initial Catalog= FiscaliasMP; User ID=sa; Password= Guerra99";
    try
    {
        using (SqlConnection con = new SqlConnection(conString))
        {
            string comando = "EXECUTE SP_EDITAR @Id_Fiscalia, @Nombre_Fiscalia, @Ubicacion";
            using (SqlCommand cmd = new SqlCommand(comando, con))
            {
                cmd.CommandType = CommandType.Text;
                cmd.Parameters.AddWithValue("@Id_Fiscalia", fiscalia.Id_Fiscalia);
                cmd.Parameters.AddWithValue("@Nombre_Fiscalia", fiscalia.Nombre_Fiscalia);
                cmd.Parameters.AddWithValue("@Ubicacion", fiscalia.Ubicacion);
                con.Open();
                int rowsAffected = cmd.ExecuteNonQuery();
                if (rowsAffected > 0)
                    return true;
                else
                    return false;
            }
        }
    }
    catch (Exception ex)
    {
        return false;
    }
}
```

- Eliminar Registros en tabla Fiscalías.

```
public bool DeleteFiscalias(int id)
{
    string conString = "Data Source= GUERRA; Initial Catalog= FiscaliasMP; User ID=sa; Password= Guerra99";
    try
    {
        using (SqlConnection con = new SqlConnection(conString))
        {
            string comando = "EXECUTE SP_ELIMINAR @Id_Fiscalia";
            using (SqlCommand cmd = new SqlCommand(comando, con))
            {
                cmd.CommandType = CommandType.Text;
                cmd.Parameters.AddWithValue("@Id_Fiscalia", id);
                con.Open();
                int rowsAffected = cmd.ExecuteNonQuery();
                if (rowsAffected > 0)
                    return true;
                else
                    return false;
            }
        }
    }
    catch (Exception ex)
    {
        return false;
    }
}
```

- Mostrar Registros filtrados por Id_Fiscalia en tabla Fiscalías.

```
public Fiscalias GetFiscalias(int id)
{
    string constring = "Data Source= GUERRA; Initial Catalog= FiscaliasMP; User ID=sa; Password= Guerra99";
    Fiscalias fiscalia = new Fiscalias();
    try
    {
        SqlDataReader reader = null;

        using (SqlConnection con = new SqlConnection(constring))
        {
            using (SqlCommand cmd = new SqlCommand("SELECT * FROM VW_FISCALIASYNUM where Id_Fiscalia = @Id_Fiscalia", con))
            {
                cmd.CommandType = CommandType.Text;
                cmd.Parameters.AddWithValue("@Id_Fiscalia", id);
                con.Open();
                reader = cmd.ExecuteReader();
                if (reader.HasRows)
                {
                    while (reader.Read())
                    {
                        fiscalia.Id_Fiscalia = Convert.ToInt32(reader.GetValue(0));
                        fiscalia.Nombre_Fiscalia = Convert.ToString(reader.GetValue(1));
                        fiscalia.Ubicacion = Convert.ToString(reader.GetValue(2));
                        fiscalia.Numero = Convert.ToDecimal(reader.GetValue(3));
                        fiscalia.Extension = Convert.ToDecimal(reader.GetValue(4));
                    }
                }
                else
                {
                    return null;
                }
            }
        }
    }
    catch (Exception ex)
    {
        return null;
    }
    return fiscalia;
}
```

- Mostrar Todos los Registros en tabla Fiscalías.

```
internal List<Fiscalias> GetAllFiscalias()
{
    string constring = "Data Source= GUERRA; Initial Catalog= FiscaliasMP; User ID=sa; Password= Guerra99";
    Fiscalias fiscalia;
    List<Fiscalias> LisFiscalias = new List<Fiscalias>();
    try
    {
        SqlDataReader reader = null;

        using (SqlConnection con = new SqlConnection(constring))
        {
            using (SqlCommand cmd = new SqlCommand
                ("SELECT * FROM VW_FISCALIAS", con))
            {
                cmd.CommandType = CommandType.Text;
                con.Open();
                reader = cmd.ExecuteReader();
                if (reader.HasRows)
                {
                    while (reader.Read())
                    {
                        fiscalia = new Fiscalias();
                        fiscalia.Id_Fiscalia = Convert.ToInt32(reader.GetValue(0));
                        fiscalia.Nombre_Fiscalia = Convert.ToString(reader.GetValue(1));
                        fiscalia.Ubicacion = Convert.ToString(reader.GetValue(2));
                        LisFiscalias.Add(fiscalia);
                    }
                }
                else
                {
                    return null;
                }
            }
        }
    }
    catch (Exception ex)
    {
        return null;
    }
    return LisFiscalias;
}
```

- Insertar Registros en tabla Teléfonos.

```
public bool InsertTelefonos(Telefonos fiscalia)
{
    string conString = "Data Source= GUERRA; Initial Catalog= FiscaliasMP; User ID=sa; Password= Guerra99";
    try
    {
        using (SqlConnection con = new SqlConnection(conString))
        {
            string comando = "EXECUTE SP_ADDNUMERO @Id_Fiscalia, @Numero, @Extension";
            using (SqlCommand cmd = new SqlCommand(comando, con))
            {
                cmd.CommandType = CommandType.Text;
                cmd.Parameters.AddWithValue("@Id_Fiscalia", fiscalia.Id_Fiscalia);
                cmd.Parameters.AddWithValue("@Numero", fiscalia.Numero);
                cmd.Parameters.AddWithValue("@Extension", fiscalia.Extension);
                con.Open();
                int rowsAffected = cmd.ExecuteNonQuery();
                if (rowsAffected > 0)
                    return true;
                else
                    return false;
            }
        }
    }
    catch (Exception ex)
    {
        return false;
    }
}
```

- Eliminar Registros en tabla Teléfonos.

```
public bool DeleteTelefonos(int id)
{
    string conString = "Data Source= GUERRA; Initial Catalog= FiscaliasMP; User ID=sa; Password= Guerra99";
    try
    {
        using (SqlConnection con = new SqlConnection(conString))
        {
            string comando = "EXECUTE SP_ELIMINARTEL @Id_Telefono";
            using (SqlCommand cmd = new SqlCommand(comando, con))
            {
                cmd.CommandType = CommandType.Text;
                cmd.Parameters.AddWithValue("@Id_Telefono", id);
                con.Open();
                int rowsAffected = cmd.ExecuteNonQuery();
                if (rowsAffected > 0)
                    return true;
                else
                    return false;
            }
        }
    }
    catch (Exception ex)
    {
        return false;
    }
}
```

- Mostrar Registros filtrados por Id_Telefono en tabla Teléfonos.

```
public Telefonos GetTelefonos(int id)
{
    string constring = "Data Source= GUERRA; Initial Catalog= FiscaliasMP; User ID=sa; Password= Guerra99";
    Telefonos telefono = new Telefonos();
    try
    {
        SqlDataReader reader = null;

        using (SqlConnection con = new SqlConnection(constring))
        {
            using (SqlCommand cmd = new SqlCommand("SELECT * FROM VW_TELEFONOS where Id_Telefono = @Id_Telefono", con))
            {
                cmd.CommandType = CommandType.Text;
                cmd.Parameters.AddWithValue("@Id_Telefono", id);
                con.Open();
                reader = cmd.ExecuteReader();
                if (reader.HasRows)
                {
                    while (reader.Read())
                    {
                        telefono.Id_Fiscalia = Convert.ToInt32(reader.GetValue(0));
                        telefono.Id_Telefonos = Convert.ToInt32(reader.GetValue(1));
                        telefono.Nombre_Fiscalia = Convert.ToString(reader.GetValue(2));
                        telefono.Ubicacion = Convert.ToString(reader.GetValue(3));
                        telefono.Numero = Convert.ToDecimal(reader.GetValue(4));
                        telefono.Extension = Convert.ToDecimal(reader.GetValue(5));
                    }
                }
                else
                {
                    return null;
                }
            }
        }
    }
    catch (Exception ex)
    {
        return null;
    }
    return telefono;
}
```

- Mostrar Registros filtrados por Id_Fiscalia en tabla Teléfonos.

```
internal List<Telefonos> GetTelefonosFis(int id)
{
    string constring = "Data Source= GUERRA; Initial Catalog= FiscaliasMP; User ID=sa; Password= Guerra99";
    Telefonos telefono = new Telefonos();
    List<Telefonos> LisTelefonos = new List<Telefonos>();
    try
    {
        SqlDataReader reader = null;

        using (SqlConnection con = new SqlConnection(constring))
        {
            using (SqlCommand cmd = new SqlCommand("SELECT * FROM VW_TELEFONOS where Id_Fiscalia = @Id_Fiscalia", con))
            {
                cmd.CommandType = CommandType.Text;
                cmd.Parameters.AddWithValue("@Id_Fiscalia", id);
                con.Open();
                reader = cmd.ExecuteReader();
                if (reader.HasRows)
                {
                    while (reader.Read())
                    {
                        telefono = new Telefonos();
                        telefono.Id_Fiscalia = Convert.ToInt32(reader.GetValue(0));
                        telefono.Id_Telefonos = Convert.ToInt32(reader.GetValue(1));
                        telefono.Nombre_Fiscalia = Convert.ToString(reader.GetValue(2));
                        telefono.Ubicacion = Convert.ToString(reader.GetValue(3));
                        telefono.Numero = Convert.ToDecimal(reader.GetValue(4));
                        telefono.Extension = Convert.ToDecimal(reader.GetValue(5));
                        LisTelefonos.Add(telefono);
                    }
                }
                else
                {
                    return null;
                }
            }
        }
    }
    catch (Exception ex)
    {
        return null;
    }
    return LisTelefonos;
}
```


6.5 Ejecutar API-REST

Ejecutamos la API-REST, esta nos dará la ruta para poder hacer la conexión en con el Front-End: <http://localhost:63570/>.

7 Desarrollo de Front-End

En esta sección se realizará una descripción de los aspectos técnicos más importantes del Front-End del aplicativo, para la modificación y mejoras de este; se desarrolló una interfaz web utilizando la técnica Ajax en Visual Code para la conexión con el BackEnd.

7.1 Diseño

Para el diseño de la página Web, se utilizó la biblioteca de Bootstrap.

```
<head>
  <meta charset='utf-8'>
  <meta http-equiv='X-UA-Compatible' content='IE=edge'>
  <title>Registrar</title>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.cs
s" integrity="sha384-
TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2"
crossorigin="anonymous">

  <script src="https://code.jquery.com/jquery-
3.0.0.js"crossorigin="anonymous"></script>

  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.m
in.js" integrity="sha384-
ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx"
crossorigin="anonymous"></script>
</head>
<body>
```

7.2 Desarrollo

Index.html, Registro.html, AgregarTelefonos.html, Telefonos.html, son las interfaces web existentes, para navegar entre las mismas.

La interfaz Index.html, se tiene la vista de código, fiscalía y ubicación, así como los botones para Crear Nueva Fiscalía, Ver Teléfonos, Editar y Eliminar.

➤ Vista:

```
window.onload = function() {
    Obtener();
};

function IrFormularioCrear(){
    window.location = "Registro.html";
}

function Obtener(){
    $(".table tbody").html("");

    $.get("http://localhost:63570/api/Fiscalias")
    .done(function( response ) {
        console.log(response);
        $.each( response, function( id, fila ) {
            $(".<tr>").append(
                $(".<td>").text(fila.Id_Fiscalia),
                $(".<td>").text(fila.Nombre_Fiscalia),
                $(".<td>").text(fila.Ubicacion),
                $(".<td>").append(
                    $(".<button>").data("id",fila.Id_Fiscalia).addClass("btn btn-info btn-sm mr-1 VerTelefono").text("Ver Teléfonos").attr({"type":"button"}),
                    $(".<button>").data("id",fila.Id_Fiscalia).addClass("btn btn-success btn-sm mr-1 editar").text("Editar").attr({"type":"button"}),
                    $(".<button>").data("id",fila.Id_Fiscalia).addClass("btn btn-danger btn-sm eliminar").text("Eliminar").attr({"type":"button"})
                )
            ).appendTo(".table");
        });
    });
}
```

➤ Botones con redirección a otra interfaz, las cuales se llevan el Id seleccionado.

```
$(document).on('click', '.VerTelefono', function () {
    console.log($(this).data("id"));
    window.location = "Telefonos.html?id=" + $(this).data("id");
});

$(document).on('click', '.editar', function () {
    console.log($(this).data("id"));
    window.location = "Registro.html?id=" + $(this).data("id");
});

$(document).on('click', '.eliminar', function () {
    console.log($(this).data("id"));

    $.ajax({
        method: "DELETE",
        url: "http://localhost:63570/api/Fiscalias/" + $(this).data("id")
    })
    .done(function( response ) {
        console.log(response);
        if(response){
            Obtener();
        }else{
            alert("Error al eliminar")
        }
    })
});
```

La interfaz Registro.html, nos muestra un formulario para registrar los datos de fiscalía y es el mismo que se utiliza para la modificación o actualización de datos.

```
80 function GuardarFiscalia(){
81     if(editar){
82
83         var data = {
84             Id_Fiscalia : $("#txtidfiscalia").val(),
85             Nombre_Fiscalia : $("#txtnombre").val(),
86             Ubicacion : $("#txtubicacion").val(),
87             Numero : $("#txtnumero").val(),
88             Extension : $("#txttextension").val()
89         }
90
91         $.ajax({
92             method: "PUT",
93             url: "http://localhost:63570/api/Fiscalias",
94             contentType: 'application/json',
95             data: JSON.stringify(data), // access in body
96         })
97         .done(function( response ) {
98             console.log(response);
99             if(response){
100                 alert("Se guardaron los cambios exitosamente");
101                 window.location = "Index.html";
102             }else{
103                 alert("Error al modificar los datos")
104             }
105         });
106     }else{
107
108         var data = {
109             Nombre_Fiscalia : $("#txtnombre").val(),
110             Ubicacion : $("#txtubicacion").val(),
111             Numero : $("#txtnumero").val(),
112             Extension : $("#txttextension").val()
113         }
114
115         $.post("http://localhost:63570/api/fiscalias", data)
116         .done(function(response) {
117             console.log(response);
118             if(response){
119                 alert("Fiscalia registrada con exito");
120                 window.location = "Index.html";
121             }else{
122                 alert("Error al registrar");
123             }
124         });
125     }
126 }
```

La interfaz Telefonos.html, se tiene la vista de los teléfonos registrados en la fiscalía, con los botones de agregar teléfono y eliminar teléfono.

```
$.get("http://localhost:63570/api/Numeros/" + idFiscalia)
.done(function( response ) {
    console.log(response);
    $.each( response, function( id, fila ) {
        $("<tr>").append(
            $("<td>").text(fila.Numero),
            $("<td>").text(fila.Extension),
            $("<td>").append(
                $("<button>").data("id",fila.Id_Telefonos).addClass("btn btn-info btn-sm mr-1 AgregarTelefono").text("Agregar Teléfono").attr({"type":"button"}),
                $("<button>").data("id",fila.Id_Telefonos).addClass("btn btn-danger btn-sm eliminar").text("Eliminar Teléfono").attr({"type":"button"})
            )
        ).appendTo("#table");
    });
});

$(document).on('click', '.AgregarTelefono', function () {
    console.log($(this).data("id"));
    window.location = "AgregarTelefonos.html?id=" + $(this).data("id");
});

$(document).on('click', '.eliminar', function () {
    console.log($(this).data("id"));

    $.ajax({
        method: "DELETE",
        url: "http://localhost:63570/api/Numeros/" + $(this).data("id")
    })
    .done(function( response ) {
        console.log(response);
        if(response){
            window.location = "Index.html";
            alert("Eliminado con exito")
        }else{
            alert("Error al eliminar")
        }
    });
});
```

La interfaz AgregarTelefonos.html, nos muestra un formulario para registrar un nuevo número de teléfono para la fiscalía seleccionada.

```
}

function PintarFiscalia(idTelefono){

$.get("http://localhost:63570/api/Telefonos/" + idTelefono)
.done(function( response ) {
    console.log(response);
    $("#txtidfiscalia").val(response.Id_Fiscalia),
    $("#txtnombre").val(response.Nombre_Fiscalia),
    $("#txtubicacion").val(response.Ubicacion)
});
}

function GuardarTelefono(){
    var data = {
        Id_Fiscalia : $("#txtidfiscalia").val(),
        Numero : $("#txtnumero").val(),
        Extension : $("#txttextension").val()
    }

    $.post("http://localhost:63570/api/Telefonos", data)
    .done(function(response) {
        console.log(response);
        if(response){
            alert("Número registrado con éxito");
            window.location = "Index.html";
        }else{
            alert("Error al registrar");
        }
    });
}
```