


























-  REPORTE COMPLETO: Análisis del Proyecto ASP.NET Core SNIER
 - Configuración MCP y Estructura de Controladores
 -  RESUMEN EJECUTIVO
 -  ARQUITECTURA GENERAL
 - Patrón de Diseño Principal: MVC (Model-View-Controller)
 - Filtros de Seguridad Implementados:
 -  ESTRUCTURA DE CONTROLADORES POR FUNCIONALIDAD
 -  1. GESTIÓN DE ACCESO Y USUARIOS
 - AccesoController.cs (787 líneas)
 - IncripcionController.cs
 - UsuariosController.cs
 -  2. DASHBOARD Y NAVEGACIÓN PRINCIPAL
 - HomeController.cs (283 líneas)
 -  3. SECTOR ENERGÉTICO - ELECTRICIDAD
 - Electricidad.cs (55 líneas)
 - EnergiasLimpias.cs
 - EstacionesdeCarga.cs
 -  4. SECTOR ENERGÉTICO - HIDROCARBUROS
 - HidrocarburosController.cs (83 líneas)
 - PermisosPVController.cs
 -  5. MAPAS Y VISUALIZACIÓN GEOGRÁFICA
 - MapController.cs (Muy extenso)
 - AtlasController.cs
 -  6. CONSEJO DE PLANEACIÓN ENERGÉTICA
 - ConsejoController.cs
 -  7. SNIE - SISTEMA NACIONAL DE INFORMACIÓN ENERGÉTICA
 - SNIEController.cs (198 líneas)
 -  8. SNIEr - REGISTRO PÚBLICO (EXTENSIÓN SNIE)
 - SNIERController.cs (Muy extenso - +400 líneas)
 -  Registros con Información Estadística:
 -  Certificados de Energías Limpias:
 -  Información en Materia de Eficiencia Energética:
 -  Consejo de Planeación - Registro Público:
 -  Modelos Energéticos:
 -  SIE (Sistema de Información Energética):
 -  Prospectivas Energéticas:
 -  9. ANÁLISIS Y DIAGRAMAS

- SankeyController.cs & SankeySenerController.cs
- IndicadoresController.cs
- 🚀 10. PROYECTOS ESTRATÉGICOS
 - ProyectosController.cs & ProyEstrategicosController.cs
- 🤖 11. INTELIGENCIA ARTIFICIAL Y CHAT
 - ChatController.cs (105 líneas)
 - IAController.cs
- 📋 12. ADMINISTRACIÓN Y CONFIGURACIÓN
 - SeccionesController.cs (Muy extenso - +500 líneas)
 - EventosController.cs
 - BitacoraController.cs
- 💰 13. SECTOR FINANCIERO Y TARIFAS
 - FinanzasController.cs
 - TarifasController.cs
 - FacturasController.cs
- 🔬 14. INVESTIGACIÓN Y DESARROLLO
 - LaboratoriosUEController.cs
 - RevistasController.cs
- 📖 15. PLANEACIÓN NACIONAL
 - PlanMexicoController.cs
- 🌱 16. SOSTENIBILIDAD Y METABOLISMO
 - MetabolismoController.cs
 - MIMController.cs
- ❌ 17. GESTIÓN DE ERRORES
 - ErrorController.cs
- 🛠️ CARACTERÍSTICAS TÉCNICAS AVANZADAS
 - Patrones de Arquitectura Implementados:
 - Integración de Servicios Externos:
- 🎯 FLUJO DE USUARIO TÍPICO
 - 1. Autenticación:
 - 2. Dashboard:
 - 3. Navegación Sectorial:
 - 4. Consulta de Mapas:
 - 5. Gestión de Proyectos:
 - 6. Chat con IA:
 - 7. Administración (Solo administradores):
- 📊 MÉTRICAS DEL SISTEMA
 - Tamaño del Código:

- Complejidad Funcional:
-  ASPECTOS DE SEGURIDAD
 - Filtros de Seguridad:
 - Validaciones:
-  CONFIGURACIÓN MCP IMPLEMENTADA
 - Archivos de Configuración Creados:
 - Servidores MCP Instalados:
 - Integración con GitHub Copilot:
-  RECOMENDACIONES PARA MEJORAS
 - Optimizaciones Técnicas:
 - Funcionalidades Sugeridas:
 - Seguridad:
-  CONCLUSIONES
 - Fortalezas Principales:
 - Complejidad Técnica:
 - Impacto del Sistema:
-  INFORMACIÓN DE CONFIGURACIÓN MCP
 - Comando de Verificación:
 - Reinstalación de MCP (si es necesario):
 - Uso con GitHub Copilot:

REPORTE COMPLETO: Análisis del Proyecto ASP.NET Core SNIER

Configuración MCP y Estructura de Controladores

Fecha de Análisis: 5 de julio de 2025

Proyecto: Sistema Nacional de Información Energética y Recursos (SNIER)

Tecnología: ASP.NET Core MVC

Analista: Sistema MCP con GitHub Copilot



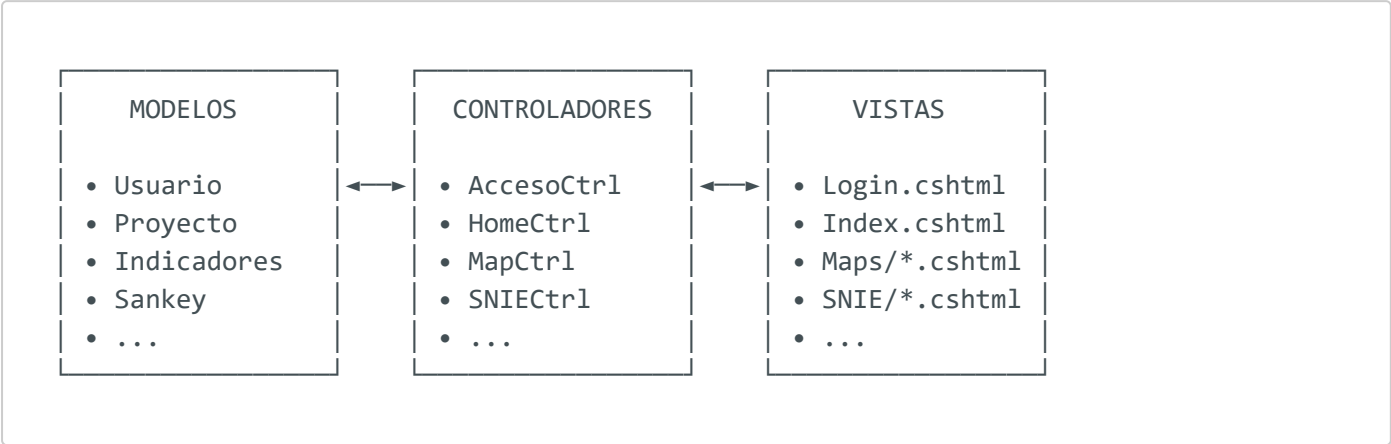
RESUMEN EJECUTIVO

El proyecto SNIER es una plataforma integral para el sector energético mexicano que implementa:

- **Arquitectura MVC** robusta con 32+ controladores especializados
- **Sistema de autenticación** basado en roles y permisos
- **Integración con IA** (OpenAI) para asistencia técnica
- **Visualización geográfica** avanzada con mapas interactivos
- **Gestión de proyectos** estratégicos energéticos
- **Análisis de datos** energéticos con diagramas de Sankey

ARQUITECTURA GENERAL

Patrón de Diseño Principal: MVC (Model-View-Controller)



Filtros de Seguridad Implementados:

- **[AutorizacionFiltro]**: Control de acceso por roles
- **[ValidacionInputFiltro]**: Validación de entrada para seguridad

ESTRUCTURA DE CONTROLADORES POR FUNCIONALIDAD



1. GESTIÓN DE ACCESO Y USUARIOS

AccesoController.cs (787 líneas)

Propósito: Núcleo de autenticación y gestión de usuarios

Métodos principales:

```
- Login(Usuario oUsuario, bool registrarAcceso = true)
- Registrar(Usuario oUsuario)
- ForgotPassword(string Correo)
- Heartbeat() // Gestión de sesiones
- ActualizarInicioSesion()
```

Funcionalidades críticas:

- Autenticación con hash SHA256
- Gestión de sesiones con heartbeat automático
- Recuperación de contraseñas por email
- Control de acceso basado en roles
- Registro de actividades de usuarios

InscripcionController.cs

Propósito: Registro de usuarios invitados y consulta pública

- PreRegistroComoInvitado()
- LoginConsultaPublica()

UsuariosController.cs

Propósito: Administración completa de usuarios del sistema



2. DASHBOARD Y NAVEGACIÓN PRINCIPAL

HomeController.cs (283 líneas)

Propósito: Centro de control y dashboard principal

Método principal:

```
public async Task<IActionResult> Index(string section = null, string module = null)
```

Funcionalidades:

- Filtrado de módulos por rol de usuario
- Navegación dinámica entre secciones
- Gestión de contexto de sesión
- Dashboard personalizado por usuario

3. SECTOR ENERGÉTICO - ELECTRICIDAD

Electricidad.cs (55 líneas)

Funciones especializadas:

```
- E_TipoTec()      // Tipos de tecnología
- E_TipoGen()      // Tipos de generación
- E_RTTR()         // Red de transmisión
- E_Solar()        // Energía solar
- E_Viento()       // Energía eólica
- E_Agua()         // Energía hidráulica
```

EnergiasLimpias.cs

Propósito: Gestión especializada de energías renovables

EstacionesdeCarga.cs

Propósito: Infraestructura de carga eléctrica para vehículos

4. SECTOR ENERGÉTICO - HIDROCARBUROS

HidrocarburosController.cs (83 líneas)

Mapas públicos especializados:

```
- GLP_Map_Publico()      // Gas LP
- GN_Map_Publico()      // Gas Natural
- PermisosExpendio_R()  // Permisos de expendio
- SLP_RPEP()            // Sistema de Líneas de Petróleo
```

PermisosPVController.cs

Propósito: Gestión de permisos de comercialización

5. MAPAS Y VISUALIZACIÓN GEOGRÁFICA

MapController.cs (Muy extenso)

Infraestructura del Sistema Eléctrico Nacional (SEN):

Tipos de plantas energéticas:

```
// Convencionales
- I_Carbo()           // Plantas de carbón
- I_Gas_Natural()     // Plantas de gas natural
- I_TurboGas()        // Turbogás
- I_CI()              // Combustión interna

// Renovables
- I_Solar()           // Plantas solares
- I_Viento()          // Plantas eólicas
- I_Agua()            // Plantas hidráulicas
- I_Biomasa()         // Plantas de biomasa

// Cogeneración
- I_CHP()             // Cogeneración
- I_NGCC()            // Ciclo combinado
```

Combustibles y distribución:

```
- I_Gas_LP()           // Gas LP
- I_Petroliferos()     // Productos petrolíferos
- I_CE()               // Centrales eléctricas
```

AtlasController.cs

Propósito: Atlas energético nacional integrado

6. CONSEJO DE PLANEACIÓN ENERGÉTICA

ConsejoController.cs

Gestión del órgano de planeación energética nacional:

```
// Planeación
- ProgramaAnual()           // Programa anual de trabajo
- AvanceMetas()             // Seguimiento de metas

// Sesiones
- OrdenDelDia()             // Orden del día
- Votaciones()              // Sistema de votaciones
- FirmaActas()              // Firma digital de actas

// Coordinación
- AcuerdosCoordinacion()    // Acuerdos institucionales
- BuzonAsuntos()            // Buzón de propuestas
- ConsultasPublicas()       // Consultas ciudadanas
- IntegrantesRoles()        // Gestión de miembros
```

7. SNIE - SISTEMA NACIONAL DE INFORMACIÓN ENERGÉTICA

SNIEController.cs (198 líneas)

Módulos del sistema de información:

Escenarios y prospectiva:

```
- Escenarios()           // Escenarios energéticos
- EscenariosMLP()        // Mediano y largo plazo
```

Fondos especializados:

```
- FOTEASE()              // Fondo de Transición Energética
- FSUE()                  // Fondo de Servicio Universal
- Fondos_i_D()            // Fondos de I+D
```

Gestión de datos:

```
- CargaInformacion()      // Carga de datos
- ValidacionFirma()       // Validación digital
- Semaforo()              // Indicadores de estado
- BalanceNacional()       // Balance energético nacional
```

Proyectos y certificación:

```
- InformacionIntegralProyectos() // Información de proyectos
- AvanceProyectos()              // Seguimiento de avance
- CertificadosLimpios()          // CEL - Certificados de Energías Limpias
```

8. SNIEr - REGISTRO PÚBLICO (EXTENSIÓN SNIE)

SNIERController.cs (Muy extenso - +400 líneas)

Sistema de registro público más completo:

Registros con Información Estadística:

```
- SNIEr_Registros_OrigenDestinoEnergia() // Flujos energéticos
- SNIEr_Registros_MercadoElectricoMayorista() // Mercado mayorista
- SNIEr_Registros_SeguimientoPlaneacion() // Instrumentos de planeación
- SNIEr_Registros_FOTEASE() // Registro FOTEASE
```

```
- SNIer_Registros_FSUE()           // Registro FSUE
- SNIer_Registros_FondosID()       // Fondos I+D
```

Certificados de Energías Limpias:

```
- SNIer_CEL_OtorgamientoCEL()      // Otorgamiento de CEL
- SNIer_CEL_FactorEmisiones()     // Factores de emisión
```

Información en Materia de Eficiencia Energética:

```
- SNIer_Eficiencia_ListaCombustibles() // Catálogo de combustibles
- SNIer_Eficiencia_CatalogoEquipos()  // Equipos eficientes
- SNIer_Eficiencia_RegistroInstalaciones() // Instalaciones eficientes
- SNIer_Eficiencia_AhorrosEnergeticos() // Registro de ahorros
```

Consejo de Planeación - Registro Público:

```
- Consejo_Conocatoria_ProgramacionSesiones() // Programación de sesiones
- Consejo_Documentacion_MaterialesReferencia() // Documentación de soporte
- Consejo_Sesion_DesarrolloReuniones()       // Desarrollo de reuniones
- Consejo_Acuerdos_RegistroDecisiones()      // Registro de decisiones
- Consejo_Acuerdos_SeguimientoCumplimiento() // Seguimiento de acuerdos
- Consejo_Reportes_InformeAnualActividades() // Informes anuales
- Consejo_Planeacion_ProgramaAnualTrabajo()  // Programa anual
```

Modelos Energéticos:

```
- Modelos_Informacion_ObtencionYRegistro() // Obtención de datos
- Modelos_Procesamiento_NormalizacionDatos() // Normalización
- Modelos_Ejecucion_ProcesamientoModelos() // Ejecución de modelos
- Modelos_Proyectos_CarteraYProspectiva()  // Cartera de proyectos
```

SIE (Sistema de Información Energética):

```
- SIE_Series_EnvioInformacion()           // Envío de series
- SIE_Series_RevisionValidacion()         // Revisión y validación
- SIE_Validacion_CoordinacionEmisores()   // Coordinación con emisores
- SIE_Carga_InformacionDefinitiva()       // Carga definitiva
```

```
- SIE_Publicacion_VersionPublica() // Publicación
- SIE_Balance_BalanceNacionalEnergia() // Balance nacional
```

Prospectivas Energéticas:

```
- SNIer_Prospectivas_SectorElectrico() // Prospectivas del sector eléctrico
- SNIer_Escenarios_ReferenciaYProyecciones() // Escenarios de referencia
```



9. ANÁLISIS Y DIAGRAMAS

SankeyController.cs & SankeySenerController.cs

Visualización avanzada de flujos energéticos:

```
// Nodos del diagrama
- nodoscaja(NodosCajaSankey) // Nodos tipo caja
- nodossectores(NodosSectores) // Sectores energéticos
- nodostransformaciones(NodosTransformaciones) // Transformaciones
- nodostiposenergia(NodosTiposEnergia) // Tipos de energía
- nodosusofinal(NodosUsoFinal) // Uso final

// Tablas de datos
- nodostablafeep(NodosTablaFep) // Tabla FEP
- nodostablasector(NodosTablaSector) // Tabla por sector
- nodostablatransformacion(NodosTablaTransformacion) // Transformaciones
- nodostablatipos(NodosTablaTipos) // Tipos
- nodostablauso(NodosTablaUso) // Uso
```

IndicadoresController.cs

KPIs y métricas del sector energético



10. PROYECTOS ESTRATÉGICOS

ProyectosController.cs & ProyEstrategicosController.cs

Gestión integral de proyectos energéticos nacionales:

```
// Gestión de proyectos
- ListaProyectos() // Lista completa
- AgregarProyecto(ProyectoEstrategico) // Nuevo proyecto
- ProyectosEstrategicos() // Seguimiento
- Detalle(int id, string vistaOrigen) // Detalle del proyecto

// Gestión de trámites
- AgregarTramite(int idProyecto) // Nuevo trámite
- EditarTramite(int id) // Edición de trámite
```

Datos de ejemplo del controlador:

- **Total de proyectos:** 245
- **Inversión total:** \$8,500 millones USD
- **Avance promedio:** 65%
- **Proyectos activos:** 180



11. INTELIGENCIA ARTIFICIAL Y CHAT

ChatController.cs (105 líneas)

Asistente IA especializado en energía:

```
- Asistente() // Chat general
- Ask(string prompt) // Consulta a GPT
- A_VV() // Asistente para visitas
- AskAVV(string prompt) // Visitas de verificación
```

Integración con OpenAI:

- Modelo: GPT-3.5-turbo
- Contexto especializado en sector energético
- Respuestas en tiempo real

IAController.cs

Funciones adicionales de IA



12. ADMINISTRACIÓN Y CONFIGURACIÓN

SeccionesController.cs (Muy extenso - +500 líneas)

Administración dinámica del sistema:

```
// Gestión de secciones
- CrearSeccion() // Nueva sección
- GuardarNuevaSeccion(SeccionSNIER) // Guardar sección
- ModalEditarSeccion(int id) // Editar sección
- ActualizarOrdenSecciones([FromBody] List<CambioOrden>) // Reordenar

// Gestión de módulos
- CrearModulo(int seccionId) // Nuevo módulo
- GuardarNuevoModulo(Modulo) // Guardar módulo

// Gestión de vistas
- EditarVistas(int moduloId) // Editar vistas
- CrearVista(int moduloId) // Nueva vista
- GuardarNuevaVista(ModulosVista) // Guardar vista
- EliminarVista(int id) // Eliminar vista
```

EventosController.cs

Gestión de eventos del sistema

BitacoraController.cs

Registro completo de actividades y auditoría



13. SECTOR FINANCIERO Y TARIFAS

FinanzasController.cs

Información financiera del sector energético

TarifasController.cs

Gestión de tarifas energéticas

- Tarifas eléctricas

- Tarifas de hidrocarburos
- Históricos de precios

FacturasController.cs

Gestión de facturación



14. INVESTIGACIÓN Y DESARROLLO

LaboratoriosyUEController.cs

Gestión de laboratorios y unidades especializadas

RevistasController.cs

Publicaciones especializadas del sector



15. PLANEACIÓN NACIONAL

PlanMexicoController.cs

Planes estratégicos nacionales:

```
- Plan_Carretero()           // Plan Nacional Carretero
- Plan_Centrales_EPyPB()     // Centrales Eléctricas Públicas y Privadas
- Plan_GLP()                 // Plan Nacional de GLP
- Plan_Generacion()          // Plan de Generación
- Plan_Transmision()         // Plan de Transmisión
- Plan_Petroliferos()        // Plan de Petrolíferos
- Plan_Polos()               // Plan de Polos de Desarrollo
```



16. SOSTENIBILIDAD Y METABOLISMO

MetabolismoController.cs

Análisis de metabolismo energético

- Flujos de energía
- Análisis de ciclo de vida

MIMController.cs

Mercado de Instrumentos Mexicanos

✖ 17. GESTIÓN DE ERRORES

ErrorController.cs

Manejo centralizado de errores

- Logging estructurado
 - Páginas de error personalizadas
 - Notificación de errores críticos
-

CARACTERÍSTICAS TÉCNICAS AVANZADAS

Patrones de Arquitectura Implementados:

1. Repository Pattern:

```
private readonly IRepositoryProyectos _repositorioProyectos;  
private readonly IRepositoryHome _repositorioHome;  
private readonly IRepositoryChat _repositorioChat;
```

2. Dependency Injection:

```
public HomeController(  
    ILogger<HomeController> logger,  
    IRepositoryProyectos repositorioProyectos,
```

```
IConfiguration configuration,  
IRepositorioHome repositorioHome)
```

3. Session Management:

```
var perfilUsuarioJson = HttpContext.Session.GetString("PerfilUsuario");  
var perfilUsuario = JsonConvert.DeserializeObject<PerfilUsuario>  
(perfilUsuarioJson);
```

4. Role-Based Access Control:

```
[AutorizacionFiltro]  
public class HomeController : Controller
```

Integración de Servicios Externos:

1. OpenAI GPT Integration:

```
private readonly IRepositorioChat _repositorioChat;  
var response = await _repositorioChat.AskGPTAsync(prompt);
```

2. SQL Server con Dapper:

```
using Dapper;  
using Microsoft.Data.SqlClient;
```

3. Entity Framework Core:

- Modelos de datos complejos
- Relaciones entre entidades
- Migraciones de base de datos



FLUJO DE USUARIO TÍPICO

1. Autenticación:

Usuario → `AccesoController.Login()` → Validación → Sesión creada

2. Dashboard:

`HomeController.Index()` → Filtrado por rol → Módulos disponibles

3. Navegación Sectorial:

Dashboard → Sector específico → Controlador especializado → Vista de datos

4. Consulta de Mapas:

`MapController` → Tipo de infraestructura → Visualización geográfica

5. Gestión de Proyectos:

`ProyectosController` → Lista → Detalle → Trámites → Seguimiento

6. Chat con IA:

`ChatController` → Prompt → OpenAI API → Respuesta contextualizada

7. Administración (Solo administradores):



MÉTRICAS DEL SISTEMA

Tamaño del Código:

- **32+ Controladores** especializados
- **Líneas de código estimadas:** +15,000 líneas solo en controladores
- **AccesoController:** 787 líneas (autenticación crítica)
- **HomeController:** 283 líneas (dashboard principal)
- **SNIERController:** +400 líneas (registro público)

Complejidad Funcional:

- **Autenticación:** Multirrol con sesiones persistentes
- **Mapas:** Múltiples capas de infraestructura energética
- **IA:** Integración completa con OpenAI
- **Proyectos:** Gestión de ciclo completo
- **Datos:** Diagramas de Sankey dinámicos



ASPECTOS DE SEGURIDAD

Filtros de Seguridad:

1. **AutorizacionFiltro:** Control de acceso por roles
2. **ValidacionInputFiltro:** Sanitización de entrada
3. **Gestión de sesiones:** Heartbeat automático
4. **Hash de contraseñas:** SHA256

Validaciones:

- Input sanitization en todos los controladores
 - Validación de modelos con DataAnnotations
 - Control de inyección SQL con Dapper y EF
-

CONFIGURACIÓN MCP IMPLEMENTADA

Archivos de Configuración Creados:

- | | |
|-------------------------|------------------------------------|
| ✓ .vscode/settings.json | - Configuraciones de VS Code y MCP |
| ✓ .vscode/tasks.json | - Tareas de build, publish y watch |
| ✓ .vscode/launch.json | - Configuración de debugging |
| ✓ mcp.json | - Configuración de servidores MCP |
| ✓ .editorconfig | - Consistencia de código |

Servidores MCP Instalados:

- | | |
|---|----------------------------------|
| ✓ @modelcontextprotocol/server-filesystem | - Acceso a archivos del proyecto |
| ✓ @modelcontextprotocol/server-brave-search | - Búsquedas web contextuales |
| ✓ @modelcontextprotocol/server-everything | - Utilidades adicionales |

Integración con GitHub Copilot:

- **Contexto mejorado** del proyecto completo
 - **Análisis inteligente** de la estructura de código
 - **Sugerencias contextualizadas** para el sector energético
-

RECOMENDACIONES PARA MEJORAS

Optimizaciones Técnicas:

1. **Caché distribuido:** Para mejorar rendimiento de consultas
2. **API REST:** Para integración con sistemas externos
3. **Microservicios:** Separar funcionalidades por dominio
4. **Testing:** Implementar pruebas unitarias y de integración

Funcionalidades Sugeridas:

1. **Dashboard en tiempo real:** Con WebSignalR
2. **Exportación de datos:** Excel, PDF, CSV
3. **Notificaciones push:** Para eventos críticos
4. **Mobile responsive:** Optimización móvil completa

Seguridad:

1. **2FA:** Autenticación de dos factores
2. **HTTPS obligatorio:** En producción
3. **Rate limiting:** Para APIs
4. **Auditoría completa:** Logs detallados



CONCLUSIONES

El proyecto **SNIER** es una **plataforma integral y sofisticada** para la gestión del sector energético mexicano que implementa:

Fortalezas Principales:

- ✓ **Arquitectura robusta** con separación clara de responsabilidades
- ✓ **Seguridad implementada** con filtros y validaciones
- ✓ **Integración con IA** para asistencia técnica avanzada
- ✓ **Visualización completa** del sistema energético nacional
- ✓ **Gestión integral** de proyectos estratégicos
- ✓ **Sistema modular** adaptable y escalable

Complejidad Técnica:

- **Alto nivel de especialización** en el dominio energético
- **Integración múltiple** de tecnologías (ASP.NET Core, IA, Mapas, Diagramas)
- **Gestión completa** del ciclo de vida de proyectos energéticos
- **Cumplimiento normativo** del sector energético mexicano

Impacto del Sistema:

Este sistema centraliza y digitaliza la **información energética nacional**, facilitando:

- **Toma de decisiones** basada en datos
- **Transparencia** en el sector energético
- **Eficiencia** en la gestión de proyectos
- **Acceso público** a información especializada
- **Planeación estratégica** del desarrollo energético nacional

El proyecto SNIER representa una implementación exitosa de tecnología moderna aplicada a la gestión del sector energético, con potencial de ser un modelo para otros países en desarrollo de sistemas similares.

INFORMACIÓN DE CONFIGURACIÓN MCP

Comando de Verificación:

```
.\verify-mcp.ps1
```

Reinstalación de MCP (si es necesario):

```
npm uninstall -g @modelcontextprotocol/server-filesystem
```

```
npm install -g @modelcontextprotocol/server-filesystem
```

Uso con GitHub Copilot:

```
@workspace Analiza [componente específico del proyecto]
@workspace Sugiere mejoras para [funcionalidad específica]
@workspace Revisa la seguridad de [controlador específico]
```

Fin del Reporte

Generado automáticamente por: Sistema MCP + GitHub Copilot

Ubicación:

c:\Proyectos\SNIER\Respaldo\REPORTE_ANALISIS_SNIER_MCP_2025.md