

APLICACIÓN WEB PARA GESTIÓN DE INVENTARIOS DE EMISIONES FUGITIVAS  
DE GASES HIDROCARBUROS EN EL SECTOR OIL&GAS PARA LA CORPORACIÓN  
CENTRO DE DESARROLLO TECNOLÓGICO - CDT DE GAS

DIANA CAROLINA MONCADA BOHÓRQUEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS  
BUCARAMANGA  
2024

APLICACIÓN WEB PARA GESTIÓN DE INVENTARIOS DE EMISIONES FUGITIVAS  
DE GASES HIDROCARBUROS EN EL SECTOR OIL&GAS PARA LA CORPORACIÓN  
CENTRO DE DESARROLLO TECNOLÓGICO - CDT DE GAS

DIANA CAROLINA MONCADA BOHÓRQUEZ

Trabajo de grado para optar por el título de:  
Ingeniero de Sistemas

Director

MANUEL GUILLERMO FLÓREZ BECERRA  
Msc. Informática - Ingeniero de Sistemas

Tutor

LUIS ALEJANDRO DUARTE MARTÍNEZ  
Ingeniero de Sistemas

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS  
BUCARAMANGA  
2024

## DEDICATORIA

*Dedico este trabajo principalmente a mi hijo, Joaquín, cuya inocente alegría y curiosidad me han inspirado a dar lo mejor de mí en cada momento. Quincito, tu presencia en mi vida ha sido una fuente constante de alegría y esperanza, ¡Hijo, tú eres mi mayor motivación!*

*A mi esposo, Gabriel, por su comprensión, paciencia y apoyo incondicional durante este arduo camino. Pandita, tu amor y aliento me han sostenido en los momentos más desafiantes de este proceso.*

*A mis padres, Jairo y Elsa, por su sabiduría y por haberme inculcado los valores de la perseverancia y el esfuerzo. Mamá, papá, los lazos familiares son más fuertes que las adversidades, con amor y paciencia todo se puede.*

*A mi hermano, Kike, por su constante ánimo y por ser un ejemplo de dedicación y esfuerzo. Pachote, siempre has estado a mi lado para apoyarme, creyendo en mí más de lo que yo misma confío.*

*A mi niña interior, porque un día lo soñé y hoy lo estoy haciendo realidad.*

*Ustedes son las personas más importantes de mi vida. Sin su apoyo constante, este logro no habría sido posible.*

## **AGRADECIMIENTOS**

Manifiesto gratitud al Universo y a la divinidad. Gracias Dios por haber iluminado mi camino y brindado la oportunidad de superar cada obstáculo en mi travesía hacia convertirme en ingeniera. Ángeles, gracias por su constante guía y protección.

Quiero expresar mi más profundo y sincero agradecimiento a mi amado esposo, Gabriel. En los momentos más oscuros, cuando me encontraba sumida en la desesperación, fuiste tú quien me extendió la mano y me devolvió la esperanza. Gracias por ayudarme a encontrar el valor necesario para seguir adelante y no rendirme.

Al profesor Luis Carlos, director de escuela, cuya incansable motivación y apoyo fueron un faro de luz en los momentos más difíciles. Cuando parecía no haber más salidas, su orientación y ánimo me ayudaron a encontrar el camino. Gracias por ayudarme a confiar en mí misma y darme cuenta de que sí podía hacerlo.

Al profesor Manuel Guillermo, mi director de proyecto, quien compartió su experiencia y conocimiento en la supervisión de este trabajo, gracias por su guía durante todo este proceso. A la Corporación Centro de Desarrollo Tecnológico del Gas (CDT de Gas), gracias por brindarme un entorno profesional enriquecedor y las oportunidades de aprendizaje que me permitieron aplicar y expandir mis conocimientos.

A mi familia y amigos, por su comprensión, paciencia y apoyo emocional a lo largo de este periodo. A mi querida amiga Lady X, cuyo espíritu diligente y generosidad infinita fueron una bendición en mi vida. Gracias por tu cariño y amistad.

Finalmente, a la universidad, por proporcionarme el entorno y los recursos necesarios para mi formación académica y profesional. Gracias por ser un espacio de aprendizaje y crecimiento.

## TABLA DE CONTENIDO

<b>INTRODUCCIÓN .....</b>	<b>16</b>
<b>1. GENERALIDADES DEL PROYECTO .....</b>	<b>18</b>
1.1. DEFINICIÓN DEL PROBLEMA .....	18
1.2. JUSTIFICACIÓN .....	19
1.3. OBJETIVOS.....	21
1.4. METODOLOGÍA .....	22
1.5. ALCANCE DEL PROYECTO .....	24
<b>2. MARCO CONCEPTUAL .....</b>	<b>25</b>
2.1. CORPORACIÓN CENTRO DE DESARROLLO TECNOLÓGICO DEL GAS – CDT DE GAS .....	25
2.2. GASES HIDROCARBUROS .....	27
2.3. EFECTO INVERNADERO, GEI Y CALENTAMIENTO GLOBAL .....	27
2.4. INVENTARIO DE GASES DE EFECTO INVERNADERO .....	29
2.5. EMISIONES POR COMBUSTIÓN .....	30
2.6. EMISIONES FUGITIVAS .....	32
2.7. INGENIERÍA DE SOFTWARE .....	32
2.8. PRUEBAS DE SOFTWARE.....	33
2.9. APLICACIÓN WEB .....	34
2.10. COMPONENTE BACKEND .....	34
<b>3. REQUERIMIENTOS DEL SISTEMA .....</b>	<b>37</b>
3.1. DESCRIPCIÓN DEL SISTEMA .....	38

3.2.	REQUERIMIENTOS FUNCIONALES .....	41
3.3.	REQUERIMIENTOS NO FUNCIONALES.....	60
3.4.	CLASES Y CARACTERÍSTICAS DE USUARIOS .....	61
3.5.	ENTORNO OPERATIVO .....	63
<b>4.</b>	<b>DISEÑO DEL SISTEMA .....</b>	<b>65</b>
4.1.	ARQUITECTURA DEL SISTEMA .....	65
4.2.	DISEÑO DE LA PERSISTENCIA.....	69
4.3.	DISEÑO DEL API .....	74
<b>5.</b>	<b>IMPLEMENTACIÓN DEL SISTEMA .....</b>	<b>80</b>
5.1.	TECNOLOGÍAS UTILIZADAS .....	80
5.2.	ITERACIONES.....	83
5.3	PROTOTIPO DESARROLLADO.....	87
<b>6.</b>	<b>VALIDACIÓN .....</b>	<b>93</b>
6.1.	PLAN DE PRUEBAS .....	93
6.2.	AUTOMATIZACIÓN DEL PLAN DE PRUEBAS.....	99
6.3.	EJECUCIÓN Y RESULTADOS.....	104
<b>7.</b>	<b>CONCLUSIONES.....</b>	<b>108</b>
<b>8.</b>	<b>RECOMENDACIONES.....</b>	<b>110</b>
	<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>111</b>

## LISTA DE ILUSTRACIONES

<i>Ilustración 1. Desarrollo de prototipado evolutivo.....</i>	<i>22</i>
<i>Ilustración 2. Organigrama de CDT de GAS .....</i>	<i>25</i>
<i>Ilustración 3. Proceso de prestación de un servicio de CDT de GAS.....</i>	<i>26</i>
<i>Ilustración 4. Efecto Invernadero y Calentamiento Global.....</i>	<i>28</i>
<i>Ilustración 5. Backend - imagen creada con inteligencia artificial.....</i>	<i>35</i>
<i>Ilustración 6. Diagrama de caso de uso de gestión de usuarios .....</i>	<i>49</i>
<i>Ilustración 7. Diagrama de caso de uso de gestión de entidades .....</i>	<i>50</i>
<i>Ilustración 8. Diagrama de caso de uso de gestión de consultas, reportes y cálculos..</i>	<i>51</i>
<i>Ilustración 9. Vista lógica de la arquitectura del sistema .....</i>	<i>65</i>
<i>Ilustración 10. Vista de despliegue de la arquitectura del sistema .....</i>	<i>68</i>
<i>Ilustración 11. Diagrama Entidad-Relación del sistema SIEGEI .....</i>	<i>70</i>
<i>Ilustración 12. Código del método Crear Empresa con el estándar OpenAPI.....</i>	<i>79</i>
<i>Ilustración 13. Estructura del código del proyecto .....</i>	<i>87</i>
<i>Ilustración 14. Ejemplo Controlador: Clase EmpresaController.....</i>	<i>88</i>
<i>Ilustración 15. Ejemplo método HTTPPost: método AddEmpresa .....</i>	<i>88</i>
<i>Ilustración 16. Ejemplo DTO: Clase AddEmpresaDTO .....</i>	<i>89</i>
<i>Ilustración 17. Ejemplo Modelo Clase Empresa .....</i>	<i>90</i>
<i>Ilustración 18. Ejemplo contexto base de datos: clase DataContext.....</i>	<i>91</i>
<i>Ilustración 19. Presentación del API en Swagger.....</i>	<i>92</i>
<i>Ilustración 20. Visualización de la estructura del proyecto – jerarquía de ramas en Visual Studio.....</i>	<i>100</i>
<i>Ilustración 21. Visualización de la clase TestPlanta con las pruebas asociadas a la entidad Planta .....</i>	<i>101</i>
<i>Ilustración 22. Código de la prueba "Crear Planta Ok".....</i>	<i>102</i>
<i>Ilustración 23. Visualización de explorador de pruebas con las pruebas a ejecutar según el plan de pruebas diseñado .....</i>	<i>105</i>
<i>Ilustración 24. Visualización del explorador de pruebas del sistema.....</i>	<i>106</i>

## LISTA DE TABLAS

<i>Tabla 1. Características del sistema SIEGEI.....</i>	<i>39</i>
<i>Tabla 2. Funcionalidades del sistema SEGEI .....</i>	<i>41</i>
<i>Tabla 3. Requerimientos funcionales de la funcionalidad Gestión de usuarios.....</i>	<i>42</i>
<i>Tabla 4. Requerimientos funcionales de la funcionalidad Gestión de entidades.....</i>	<i>43</i>
<i>Tabla 5. Requerimientos funcionales de la funcionalidad Gestión de consultas, reportes y cálculos.....</i>	<i>44</i>
<i>Tabla 6. Especificación de requerimiento funcional RF.01- Autenticar usuario - inicio de sesión.....</i>	<i>46</i>
<i>Tabla 7. Especificación de requerimiento funcional RF.14- Actualizar sistema. ....</i>	<i>47</i>
<i>Tabla 8. Especificación de requerimiento funcional RF.33 - Consultar archivos multimedia.....</i>	<i>47</i>
<i>Tabla 9. Especificación de requerimiento funcional RF.37 - Consultar reporte mensual por tipo de gas.....</i>	<i>48</i>
<i>Tabla 10. Especificación de caso de uso para registro de usuarios.....</i>	<i>52</i>
<i>Tabla 11. Especificación de caso de uso para creación de fuente de emisión por combustión.....</i>	<i>54</i>
<i>Tabla 12. Especificación de caso de uso para generación de reporte anual acumulado por empresa y planta.....</i>	<i>57</i>
<i>Tabla 13. Requerimientos no funcionales del sistema .....</i>	<i>60</i>
<i>Tabla 14. Clases de usuarios del sistema y sus respectivas características .....</i>	<i>62</i>
<i>Tabla 15. Diccionario de datos para Tabla Empresas de la base de datos.....</i>	<i>71</i>
<i>Tabla 16. Diccionario de datos para Tabla EmisionesCombustion de la base de datos.....</i>	<i>72</i>
<i>Tabla 17. Diccionario de datos para Tabla ReporteMensualGasDetalle de la base de datos .....</i>	<i>73</i>
<i>Tabla 18. Métodos API implementados, con su descripción y su correspondiente método HTTP.....</i>	<i>74</i>
<i>Tabla 19. Documentación API - método Crear Empresa .....</i>	<i>76</i>
<i>Tabla 20. Documentación API - método Consultar Empresa .....</i>	<i>77</i>

<i>Tabla 21. Documentación API – método Actualizar Planta .....</i>	<i>77</i>
<i>Tabla 22. Documentación API – método Listar Plantas .....</i>	<i>78</i>
<i>Tabla 23. Requerimientos funcionales implementados en la primera iteración.....</i>	<i>84</i>
<i>Tabla 24. Requerimientos funcionales implementados en la segunda iteración .....</i>	<i>85</i>
<i>Tabla 25. Plan de pruebas para la entidad Empresa .....</i>	<i>93</i>
<i>Tabla 26. Plan de pruebas para la entidad Planta.....</i>	<i>94</i>
<i>Tabla 27. Plan de pruebas para la entidad Sistema.....</i>	<i>95</i>
<i>Tabla 28. Plan de pruebas para la entidad Emision Combustion .....</i>	<i>95</i>
<i>Tabla 29. Plan de pruebas para la entidad Emision Fugitiva .....</i>	<i>96</i>
<i>Tabla 30. Plan de pruebas para la entidad Evidencia .....</i>	<i>97</i>
<i>Tabla 31. Plan de pruebas para la entidad Reporte Mensual .....</i>	<i>98</i>

## **LISTA DE ANEXOS**

**ANEXO A:** DOCUMENTO VISIÓN

**ANEXO B:** DOCUMENTO DE ESPECIFICACIÓN DE REQUERIMIENTOS DEL SISTEMA

**ANEXO C:** DICCIONARIO DE DATOS

## RESUMEN

**TÍTULO:** APLICACIÓN WEB PARA GESTIÓN DE INVENTARIOS DE EMISIONES FUGITIVAS DE GASES HIDROCARBUROS EN EL SECTOR OIL&GAS PARA LA CORPORACIÓN CENTRO DE DESARROLLO TECNOLÓGICO - CDT DE GAS.\*

**AUTOR:** DIANA CAROLINA MONCADA BOHÓRQUEZ\*\*

**PALABRAS CLAVE:** CDT de GAS, Calentamiento global, Emisiones fugitivas, Gases de efecto invernadero (GEI), Aplicación web, componente Backend.

### DESCRIPCIÓN

La Corporación Centro de Desarrollo Tecnológico del Gas – CDT de GAS, es una institución sólida, humana e innovadora que ha mantenido el liderazgo tecnológico en el sector Gas, aportando calidad, confiabilidad y cumplimiento en los procesos, productos y servicios ofrecidos, en beneficio de las empresas del sector productivo y la comunidad en general.

Ante la creciente preocupación e impacto generados por el cambio climático, la evaluación de emisiones de gases de efecto invernadero (GEI) se ha convertido en un componente fundamental de las estrategias de Responsabilidad Social Empresarial, especialmente para las empresas del sector Oil&Gas. Desde el Centro de Desarrollo Tecnológico del Gas se ofrecen servicios de consultoría para la gestión, cuantificación y reporte de inventarios corporativos de emisiones de GEI, basados en la metodología del GHG Protocol y otras normativas aplicables.

El presente informe aborda el diseño de un aplicativo web para la gestión de inventarios de emisiones fugitivas de gases hidrocarburos, específicamente diseñado para la Corporación Centro de Desarrollo Tecnológico del Gas (CDT de Gas). Este proyecto surge de la necesidad de soportar el proceso de medición de contaminantes de gas natural, con herramientas tecnológicas que permitan un monitoreo preciso y eficiente de las emisiones, cumpliendo con las regulaciones ambientales y mejorando las prácticas operativas dentro del sector Oil&Gas.

La implementación de este sistema tendrá un impacto significativo en CDT de GAS al proporcionar una plataforma software que facilita la recolección, consulta, análisis y reporte de datos de emisiones. Esto no solo mejora la capacidad de la corporación para gestionar el registro de las emisiones de gases de efecto invernadero y brindar información asertiva a sus clientes para la respectiva toma de decisiones y mejoras en los procesos industriales de hidrocarburos, sino que también fortalece su compromiso con la sostenibilidad y la responsabilidad ambiental.

-----  
\* Trabajo de grado

\*\* Facultad de Ingenierías Físico-Mecánicas, Escuela de Ingeniería de Sistemas e Informática. Director: Manuel Guillermo Flórez Becerra, MSC. Informática, Profesor Planta.

## ABSTRACT

**TITLE:** WEB APPLICATION FOR MANAGING FUGITIVE EMISSION INVENTORIES OF HYDROCARBON GASES IN THE OIL & GAS SECTOR FOR THE TECHNOLOGICAL DEVELOPMENT CENTER CORPORATION - CDT OF GAS. \*

**AUTHOR:** DIANA CAROLINA MONCADA BOHÓRQUEZ\*\*

**KEYWORDS:** CDT of GAS, Global warming, Fugitive emissions, Greenhouse gases (GHG), Web application, Backend component.

### DESCRIPTION

The Corporación Centro de Desarrollo Tecnológico del Gas – CDT de GAS, is a solid, humane, and innovative institution that has maintained technological leadership in the Gas sector, providing quality, reliability, and compliance in the processes, products, and services offered, benefiting companies in the productive sector and the community in general.

Given the growing concern and impact generated by climate change, the evaluation of greenhouse gas (GHG) emissions has become a fundamental component of Corporate Social Responsibility strategies, especially for companies in the Oil&Gas sector. The Centro de Desarrollo Tecnológico del Gas offers consulting services for the management, quantification, and reporting of corporate GHG emission inventories, based on the GHG Protocol methodology and other applicable standards.

This report addresses the design of a web application for managing inventories of fugitive hydrocarbon gas emissions, specifically designed for the Corporación Centro de Desarrollo Tecnológico del Gas (CDT de Gas). This project arises from the need to support the natural gas pollutant measurement process with technological tools that allow precise and efficient monitoring of emissions, complying with environmental regulations and improving operational practices within the Oil&Gas sector.

The implementation of this system will have a significant impact on CDT de GAS by providing a software platform that facilitates the collection, consultation, analysis, and reporting of emission data. This not only enhances the corporation's ability to manage the recording of greenhouse gas emissions and provide accurate information to its clients for decision-making and improvements in hydrocarbon industrial processes, but also strengthens its commitment to sustainability and environmental responsibility.

---

\* Bachelor Thesis

\*\* Faculty of Physic Mechanical Engineering, School of Engineering and Information System. Directed by: Manuel Guillermo Flórez Becerra, MSC. Computing, Professor Plant.

## GLOSARIO

**.Net:** Framework de desarrollo de software desarrollado por Microsoft que permite crear aplicaciones web, de escritorio y móviles.

**Angular:** Framework de código abierto desarrollado por Google para crear aplicaciones web SPA (Single Page Application) robustas y escalables.

**API:** (Application Programming Interface) Interfaz de programación de aplicaciones que define cómo interactúan diferentes componentes de software.

**Arquitectura de capas:** Método de diseño de software que organiza la aplicación en capas lógicas para facilitar su mantenimiento y escalabilidad.

**Backend:** La parte de un sitio web o aplicación que no es visible para los usuarios y que se encarga de procesar la lógica de la aplicación, interactuar con la base de datos y proporcionar la funcionalidad necesaria para que el Frontend funcione correctamente.

**CDT de GAS:** Corporación Centro de Desarrollo Tecnológico del Gas

**Combustión:** Correspondientes al gas contenido en los diferentes combustibles - gas natural, crudo, gasoil, no consumido por un equipo de combustión y liberado a la atmósfera. Tales equipos son: hornos, calentadores, calderas, motores de combustión interna, generadores a gas, turbinas a gas.

**CRUD:** Acrónimo de "Crear, Leer, Actualizar, Borrar", que son las cuatro operaciones básicas de persistencia de datos en bases de datos.

**DBMS:** Sistema de gestión de bases de datos, software que permite crear, modificar y administrar bases de datos.

**Descargas o paradas de equipos:** Emisiones producidas por venteo de gas/vapor a la atmósfera por consideraciones operacionales como mantenimiento de equipo o ejercicio de parada de emergencia.

**Diagrama Entidad-Relación:** El Diagrama Entidad-Relación es una herramienta valiosa que ayuda a garantizar la calidad y la integridad de los datos en el sistema, la cual muestra de manera gráfica las entidades, atributos y relaciones del sistema. Este diagrama proporciona una representación visual clara de la estructura de la base de datos y las interacciones entre las diferentes entidades del sistema.

**Dióxido de carbono:** Gas presente en la atmósfera terrestre que contribuye al efecto invernadero y al calentamiento global.

**Disturbios del sistema:** Emisiones producidas por descarga de gas en los equipos por condiciones inesperadas.

**Efecto invernadero:** Fenómeno natural que permite que la Tierra mantenga una temperatura habitable al retener parte del calor del sol en la atmósfera.

**Emisiones fugitivas:** Liberación accidental o no intencional de gases o vapores contaminantes a la atmósfera, típicamente asociada con la industria y el transporte.

**Front-end:** La parte de un sitio web o aplicación que los usuarios ven y con la que interactúan directamente. Incluye el diseño, la interfaz de usuario y la experiencia del usuario.

**Fugitivas/fugas:** Pérdidas no intencionales de gas en los equipos y procesos. Ejemplos incluyen roturas de gasoductos ocasionada por válvulas que no cierran bien, conexiones sueltas entre tuberías, etc.

**Gases hidrocarburos:** Compuestos químicos formados por átomos de carbono e hidrógeno, como el metano y el etano, que pueden encontrarse en estado gaseoso a temperatura ambiente.

**GHG Protocol:** (Protocolo de Gases de Efecto Invernadero) Es uno de los estándares más conocidos para el cálculo de Huella de Carbono de Organización. El GHG Protocol permite contabilizar los seis tipos de gases GEI (CO<sub>2</sub>, CH<sub>4</sub>, N<sub>2</sub>O, HFC, PFC y SF<sub>6</sub>) enfocándose únicamente en la contabilidad y reporte de las emisiones.

**HTML:** Lenguaje de marcado utilizado para crear páginas web, que define la estructura y el contenido de una página mediante etiquetas y atributos.

**IGU:** Interfaz gráfica de usuario, (sus siglas en inglés, **GUI**, *Graphical User Interface*) que permite a los usuarios interactuar con dispositivos electrónicos mediante elementos gráficos como botones, menús y ventanas.

**JavaScript:** Lenguaje de programación utilizado principalmente en el desarrollo web para agregar interactividad y dinamismo a las páginas.

**Metano:** Gas de efecto invernadero que se produce principalmente a partir de la descomposición de materia orgánica y la actividad industrial, y que contribuye al calentamiento global.

**MYSQL:** Sistema de gestión de bases de datos relacional de código abierto ampliamente utilizado en aplicaciones web.

**Navegador/browser:** Aplicación de software utilizada para acceder y visualizar páginas web en internet.

**Quema en tea:** Producidas por la ineficiencia de quema de gas natural y corrientes de desecho de gas/vapor.

**UIS:** Universidad Industrial de Santander

**UML:** Lenguaje unificado de modelado, utilizado en ingeniería de software para visualizar, especificar, construir y documentar los artefactos de un sistema de software.

**Venteo:** Emisiones operativas por diseño de los equipos. Estas emisiones hacen parte de las operaciones normales de los equipos, pero resultan en la liberación de las corrientes de gas y desecho de gas/ vapor a la atmósfera.

**WEB:** Acrónimo de World Wide Web, sistema de información global que permite acceder y compartir información a través de internet.

## INTRODUCCIÓN

La sociedad actual se enfrenta a uno de los desafíos más urgentes y complejos de nuestra era: el calentamiento global. Este fenómeno, vinculado al cambio climático, representa una amenaza significativa para la estabilidad ambiental y el equilibrio climático de nuestro planeta. Entre las diversas causas que contribuyen al calentamiento global, las emisiones de gases de efecto invernadero, como el dióxido de carbono ( $CO_2$ ) y, particularmente, el metano ( $CH_4$ ), juegan un papel fundamental.

El metano, en particular, es un gas de efecto invernadero especialmente potente, con un impacto en el calentamiento global aproximadamente 25 veces mayor que el del dióxido de carbono en un período de 100 años. Las emisiones de metano provienen de diversas fuentes, entre las que se incluyen las actividades humanas, como la industria del petróleo y el gas. En este sector, las emisiones fugitivas de gases hidrocarburos son una preocupación importante, ya que contribuyen significativamente al total de emisiones de gases de efecto invernadero sin contar que también pueden ser nocivos para la salud de los trabajadores del sector.

Actualmente la empresa Corporación Centro de Desarrollo Tecnológico del Gas – CDT de Gas está comprometida con el sector Oil&Gas para mitigar el impacto de dichas emisiones, por lo que ofrece el servicio de toma de muestras de hidrocarburos y medición de contaminantes de Gas Natural. La empresa cuenta con la infraestructura requerida para la toma de muestras de gases y líquidos combustibles en campo. Su personal calificado ha desarrollado procedimientos que permiten prestar los servicios de muestreo siguiendo las normas internacionales: API 14.1 e ISO 10715.

Para apoyar el proceso de medición de emisiones fugitivas en las mediciones de campo, se están utilizando archivos Excel organizados pero aislados, lo que

dificulta la consulta de la información asociada y la correspondiente toma de decisiones. Por este motivo, una herramienta software que facilite el registro, seguimiento y análisis de las emisiones fugitivas, permitiendo una gestión más eficiente y sostenible de las mismas, que pudiera integrar los datos, con facilidad de acceso en cualquier lugar a través de internet, sería muy útil para apoyar este proceso.

Con el objetivo de aportar a corto plazo en la realización de ésta herramienta tecnológica en CDT de GAS, se ha establecido un convenio de práctica empresarial con modalidad de proyecto de grado entre la empresa y la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander – UIS, creando ésta propuesta tecnológica que se adapta a las necesidades del cliente, y cuya principal funcionalidad del aplicativo software es registrar y almacenar datos que se recolectan en campo (eventualmente los pueden facilitar los clientes de la compañía) y contribuir al análisis de la información mediante representaciones gráficas y reportes generados.

Este documento representa el informe final del trabajo de grado, donde se plasma el desarrollo y aplicación de los conocimientos adquiridos en el recorrido temático del plan de estudios, en modalidad de práctica empresarial.

## 1. GENERALIDADES DEL PROYECTO

### 1.1. DEFINICIÓN DEL PROBLEMA

El calentamiento global generado por el incremento de los Gases de Efectos Invernadero (GEI) es una de las más serias amenazas ambientales y las emisiones de fugas por combustión o por emisiones fugitivas de gases de hidrocarburos son razones de peso para que aumenten los GEI.

En una planta de procesamiento de gas se puede presentar fugas de emisiones fugitivas en alguna unidad de ella, lo que comprende las fugas involuntarias provenientes de los componentes de los equipos, incluyendo válvulas, bridas y otras conexiones, bombas, compresores, dispositivos para liberar presión, drenajes del proceso, válvulas de extremos abiertos, sistemas de venteo, entre otros. Estas fuentes de emisiones, como el venteo y quema en antorcha durante el proceso, pérdidas por evaporación y descargas accidentales o fallas en los equipos, suelen ser emisores continuos y tienen emisiones entre bajas y moderadas.

En general, todos los componentes de los equipos producen fugas de una u otra medida, sin embargo, hay fugas suficientemente significativas que urgen la reparación o reemplazo de la unidad. Así mismo, las fugas de emisiones fugitivas de los equipos son una fuente importante, sino la más significativa, de emisiones de metano ( $CH_4$ ) y de hidrocarburos diferentes del metano en las instalaciones de petróleo y gas natural. No obstante, se considera que una instalación está bien mantenida y que las fugas de los equipos están controladas de manera adecuada, si menos del 2% de las fuentes potenciales producen fuga.

## 1.2. JUSTIFICACIÓN

Actualmente la Corporación Centro de Desarrollo Tecnológico - CDT de Gas presta servicios de monitoreo de las emisiones fugitivas de las plantas de procesos de producción y a las redes de transporte y distribución de gas y otros fluidos, lo cual juega un papel fundamental para proporcionar elementos de información en función al impacto ambiental, dado que el metano provoca un efecto invernadero veinte veces mayor que el dióxido de carbono, y en función del manejo de la planta, por lo que la identificación de áreas en las que existe una mayor difusión de algún gas o la presencia de grietas o desperfectos de alguna unidad permite reconocer y examinar la posibilidad de repararla para beneficio del medio ambiente y de la compañía como tal.

A manera de aporte, desde la Corporación Centro de Desarrollo Tecnológico del Gas - CDT de GAS, ha nacido la idea de crear un aplicativo web, enfocado al uso por parte del personal encargado de registrar y consolidar los datos, de las emisiones de fugas y emisiones fugitivas de gases hidrocarburos, recogidos durante las visitas de campo a las plantas de tratamiento de gas, con orientación a facilitar el proceso de gestión de inventarios de gases de hidrocarburos desde la compañía, que integra la lúdica de las aplicaciones web con la capacidad de adaptarse a los requerimientos de las nuevas tecnologías.

La importancia del desarrollo e implementación de esta aplicación radica en mejorar el proceso cómo se gestiona el inventario de dichas fugas que se detectan mediante equipo especializado, como las cámaras de imagen térmica de infrarrojos, y personal capacitado en el aseguramiento metrológico de múltiples unidades y/o estaciones de las distintas plantas donde se toman las muestras de fuentes de combustión para usos energéticos y de fuentes de emisiones fugitivas.

Dado que actualmente se tienen los registros en archivos dispersos, la Corporación quiere que esta solución tecnológica brinde la utilidad de tener todos los registros consolidados en un mismo lugar, visualmente agradables para los usuarios y que permita la generación de reportes para la toma de decisiones.

### **1.3. OBJETIVOS**

#### **1.3.1. OBJETIVO GENERAL**

Diseñar una aplicación web que sirva de apoyo al proceso de gestión de inventarios de emisiones fugitivas de gases hidrocarburos en el sector Oil&Gas de la Corporación Centro de Desarrollo Tecnológico - CDT de Gas

#### **1.3.2. OBJETIVOS ESPECÍFICOS**

1. Identificar y especificar los requerimientos funcionales y no funcionales de la aplicación Web que apoye el proceso gestión de inventarios de emisiones fugitivas de gases hidrocarburos.
2. Diseñar la arquitectura y los componentes software de la aplicación Web que incluya un subconjunto de los principales requerimientos especificados.
3. Implementar un prototipo funcional del componente Backend de la aplicación Web de gestión de inventarios de emisiones fugitivas de gases hidrocarburos basado en el diseño realizado.
4. Validar el prototipo funcional implementado a través de un conjunto de pruebas funcionales.

## 1.4. METODOLOGÍA

### 1.4.1. METODOLOGÍA PARA EL DESARROLLO DEL SOFTWARE. PROTOTIPADO EVOLUTIVO

El desarrollo evolutivo se basa en la idea de desarrollar una implementación inicial, exponiéndola a los comentarios del usuario y refinándola a través de las diferentes versiones hasta que se desarrolla un sistema adecuado, tal como se observa en la ilustración 3. Las actividades de especificación, desarrollo y validación se entrelazan en vez de separarse, con una rápida realimentación entre éstas.



*Ilustración 1. Desarrollo de prototipado evolutivo*

Fuente: página web de Ingeniería de Software

<https://ingsoftware.weebly.com/modelo-evolutivo.html>

El enfoque de prototipado evolutivo tiene varias ventajas y beneficios:

- Retroalimentación temprana del usuario: Al presentar un prototipo inicial al usuario, se obtiene retroalimentación temprana sobre los requisitos y la funcionalidad del sistema, lo que ayuda a identificar y corregir posibles problemas desde etapas tempranas del desarrollo.
- Mayor satisfacción del usuario: Al involucrar al usuario desde el principio y permitirle interactuar con prototipos funcionales, se aumenta la probabilidad de satisfacer sus necesidades y expectativas.
- Mayor flexibilidad y adaptabilidad: El enfoque evolutivo permite adaptarse a los cambios en los requisitos y en el entorno del proyecto de forma más

eficiente, ya que el sistema puede evolucionar y adaptarse gradualmente a medida que se reciben comentarios y se realizan mejoras.

- Reducción de costos: Al identificar y corregir problemas tempranamente, se reducen los costos asociados con cambios y correcciones en etapas avanzadas del desarrollo.
- Mayor calidad del producto final: Al refinar y mejorar el sistema a través de múltiples versiones, se tiende a obtener un producto final de mayor calidad y mejor adaptado a las necesidades del usuario.
- Mayor rapidez en la entrega: Al desarrollar e iterar rápidamente sobre prototipos, se puede acelerar el tiempo de entrega del producto final.

En resumen, el prototipado evolutivo incluye una especificación que puede desarrollarse de forma creciente y un mejor entendimiento del sistema tanto de los usuarios como los desarrolladores, lo cual se refleja en una mejora de la calidad del software, permitiendo un desarrollo más ágil y orientado al usuario, lo que puede resultar en sistemas de software más adecuados y satisfactorios.

## 1.5. ALCANCE DEL PROYECTO

El alcance del proyecto implica el diseño y desarrollo del prototipo funcional del componente Backend de una aplicación web que sirva de apoyo al proceso de gestión de inventarios de emisiones fugitivas de gases hidrocarburos en el sector Oil&Gas de la Corporación Centro de Desarrollo Tecnológico - CDT de Gas. Este prototipo servirá como punto de partida para el establecimiento de un sistema software que se implantará y permitirá el apoyo efectivo en este proceso.

El proyecto se centrará en la identificación y documentación de los requerimientos funcionales y no funcionales, así como en la implementación de las funcionalidades principales de la aplicación. También se integrará la información necesaria para el registro y seguimiento de las emisiones fugitivas, y se desarrollarán funciones de generación de informes básicos y esenciales para respaldar la toma de decisiones en la gestión de las emisiones.

La implementación del prototipo funcional del componente Backend de la aplicación web de gestión de inventarios abarcará un subconjunto de los requerimientos identificados, los cuales serán validados durante esta etapa. Es importante destacar que el objetivo no es el lanzamiento oficial del aplicativo, sino la creación de un prototipo que demuestre la viabilidad y funcionalidad del sistema.

Todos los artefactos generados, incluidos los requerimientos, diseños y prototipos, serán utilizados para madurar el sistema y prepararlo para su futura implantación en la empresa.

## 2. MARCO CONCEPTUAL

### 2.1. CORPORACIÓN CENTRO DE DESARROLLO TECNOLÓGICO DEL GAS – CDT DE GAS

Es un Centro de Desarrollo Tecnológico de derecho privado y participación mixta, legalmente constituida, con carácter científico y tecnológico; hace parte del Sistema Nacional de Ciencia y Tecnología e Innovación (SNCTI), con reconocimiento vigente de Colciencias como Centro de Desarrollo Tecnológico, según la Resolución 2027 del 17 de octubre de 2023.

Desde su creación, se ha mantenido como un ente independiente, para brindar a sus clientes resultados libres de influencias económicas o comerciales, garantizando así, la transparencia de las soluciones entregadas.

El siguiente es el organigrama de la institución:

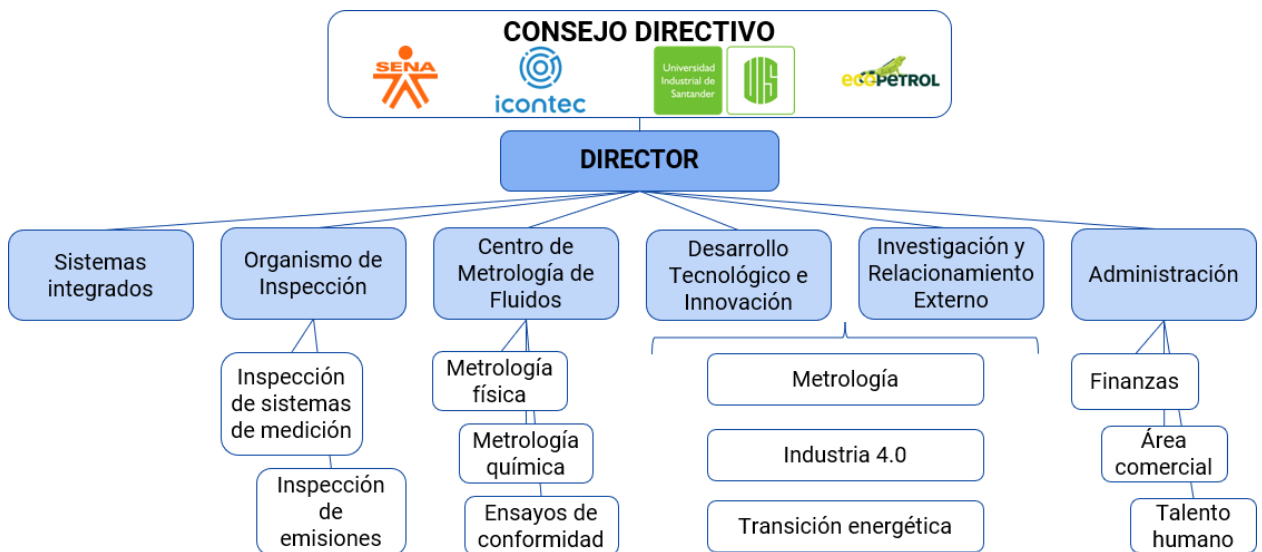


Ilustración 2. Organigrama de CDT de GAS

Fuente: página web de CDT de GAS

<https://cdtdegas.com/index.php/nosotros/quienes-somos>

La Corporación Centro de Desarrollo Tecnológico del Gas - CDT de GAS, ha mantenido el liderazgo tecnológico en el sector gas, aplicando conceptos inherentes a la metrología de fluidos, mediante la utilización de la investigación aplicada y la constante formación de talento humano, caracterizándose por la eficacia y eficiencia de soluciones, contribuyendo así, a la apropiación de los avances de la ciencia y la tecnología que hacen posible el crecimiento sostenido del sector gas y de la industria colombiana en general.

Sus procesos, productos y servicios son transversales a los diversos sectores de la economía nacional, por lo que estudian cuidadosamente la mejor estrategia para atenderlos.

Su misión es ser un equipo de trabajo comprometido, audaz e innovador, que brinda soluciones a retos de la industria por medio de la metrología aplicada, el desarrollo de tecnología, la apropiación y transferencia de conocimiento, la aplicación de IoT y la formación de competencias de personal, para contribuir al crecimiento y el desarrollo sostenible del país.



*Ilustración 3. Proceso de prestación de un servicio de CDT de GAS*

Fuente: página web de CDT de GAS

<https://cdtdegas.com/index.php/nosotros/quienes-somos>

Su visión para el 2025 es ser una institución sólida, humana y efectiva, reconocida internacionalmente por sus desarrollos tecnológicos, gestión del conocimiento y soluciones en metrología aplicada, que apoye a la industria en la transición energética y la inmersión en la cuarta revolución industrial.

## 2.2. GASES HIDROCARBUROS

Los gases hidrocarburos son compuestos químicos formados por átomos de carbono e hidrógeno. Pueden encontrarse en estado gaseoso a temperatura ambiente y se clasifican en dos grandes grupos: los alcanos y los compuestos no alcanos. Los alcanos son los más simples y están formados por una cadena lineal o ramificada de átomos de carbono unidos entre sí por enlaces sencillos. Ejemplos de alcanos incluyen el metano ( $CH_4$ ), el etano ( $C_2H_6$ ), y el propano ( $C_3H_8$ ).

Los compuestos no alcanos, por otro lado, incluyen alquenos, alquinos y aromáticos, que contienen enlaces dobles, triples o anillos de carbono, respectivamente. Ejemplos de compuestos no alcanos son el etileno ( $C_2H_4$ ), el acetileno ( $C_2H_2$ ) y el benceno ( $C_6H_6$ ).

Estos gases son de gran importancia en la industria y la vida cotidiana, ya que se utilizan como combustibles, materias primas en la industria química y como refrigerantes, entre otros usos. Sin embargo, su emisión descontrolada puede tener impactos negativos en el medio ambiente y la salud humana, debido a su contribución al calentamiento global y la formación de smog, entre otros problemas ambientales. Por ello, es importante gestionar adecuadamente su producción y uso.

## 2.3. EFECTO INVERNADERO, GEI Y CALENTAMIENTO GLOBAL

El efecto invernadero es un proceso natural que hace posible la vida en la superficie terrestre. Para mantener su balance energético, la tierra debe remitir al espacio, en forma de calor o radiación infrarroja, la energía recibida del sol que calienta la superficie de la tierra.

En su estado natural, el vapor del agua, el dióxido de carbono y otros gases de efecto invernadero (GEI) atrapan (absorben y remiten) esta radiación, lo que calienta la parte baja de la atmósfera. Este es el llamado efecto invernadero, sin el cual la temperatura promedio de la superficie terrestre sería  $-18^{\circ}\text{C}$ , en vez de  $15^{\circ}\text{C}$ .

A partir de la Revolución Industrial y la expansión de la agricultura hace dos siglos, y especialmente desde la finalización de la Segunda Guerra Mundial, las actividades humanas han generado y puesto en la atmósfera  $\text{CO}_2$ , metano ( $\text{CH}_4$ ), óxido nitroso ( $\text{N}_2\text{O}$ ) y otros GEI a tasas mayores que las que la biósfera y los océanos pueden asimilar, causando incrementos de las concentraciones atmosféricas de los GEI.

Estos incrementos de concentración por encima de los niveles naturales aumentan el efecto invernadero, es decir el atrapamiento de radiación infrarroja en la atmósfera baja, lo que causa el cambio climático. El aumento del efecto invernadero se le conoce también como calentamiento global y tiene origen antropogénico. (Véase Ilustración 4)

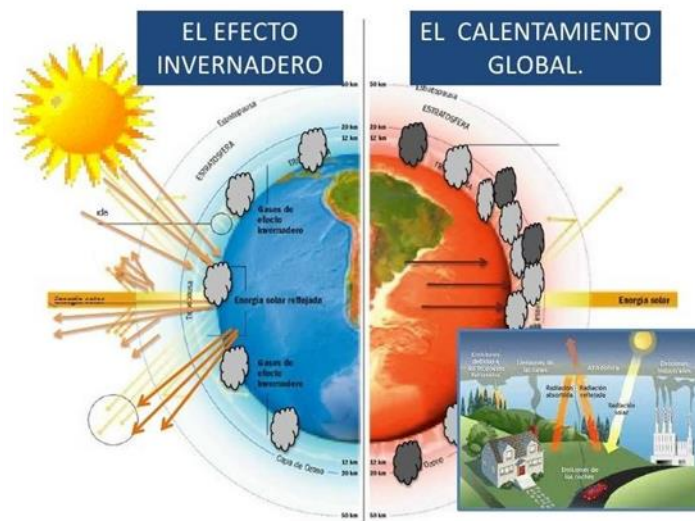


Ilustración 4. Efecto Invernadero y Calentamiento Global

Fuente: Comunidad de geógrafos

<https://www.facebook.com/1917165948331757/photos/a.1918514044863614/1929550007093351/?type=3>

Los GEI son compuestos que, aunque están presentes en la atmósfera en concentraciones muy pequeñas (gases traza), aumentan significativamente la temperatura de la baja atmósfera. Esto se debe a su capacidad para absorber y remitir radiación infrarroja. Aproximadamente tres cuartas partes del efecto invernadero natural se debe al vapor de agua.

La rapidez de los procesos físicos, químicos y biológicos que remueven cada GEI determina su tiempo de vida, el cual puede ser largo o corto. Los GEI de larga vida son dióxido de carbono (tiempo de vida mayor a 100 años), metano (12 años), óxido nitroso (121 años) y clorofluorocarbonos (CFCs - entre 45 y 1020 años). Estos son comúnmente denominados GEI directos.

El principal GEI de corta vida es el ozono ( $O_3$ ) troposférico, un contaminante fotoquímico que se forma en la baja atmósfera a partir de las emisiones de sus precursores óxidos de nitrógeno y compuestos orgánicos volátiles (COVs). Es por esta razón que el inventario de GEI contabiliza estos precursores como GEI “indirectos”.

## **2.4. INVENTARIO DE GASES DE EFECTO INVERNADERO**

Un inventario de gases de efecto invernadero es un reporte, delimitado para un periodo de tiempo y determinado territorio, de la cantidad de gases de efecto invernadero (GEI) emitidos directamente a la atmósfera.

Dado el número, la diversidad y la extensión de las fuentes de emisión y sumideros de GEI, es financiera y lógicamente imposible monitorear todas las fuentes de emisión o sumideros de GEI a escala nacional o departamental en el tiempo y lugar en los que ocurren. Por consiguiente, los inventarios son estimaciones de las emisiones y absorciones, y no una contabilidad de valores perfectamente conocidos.

Las estimaciones se realizan, usualmente, para seis gases de efecto invernadero: dióxido de carbono, metano, óxido nitroso, hidrocarburos, hexafluoruro de azufre y perfluorocarbono.

Los inventarios de emisiones GEI, además de ser un reporte, son una herramienta para orientar en el país, o en determinada organización, la toma de decisiones en materia de implementación de acciones de mitigación.

## 2.5. EMISIONES POR COMBUSTIÓN

Las emisiones relacionadas con el uso de la energía incluyen las emisiones de dióxido de carbono ( $CO_2$ ), metano ( $CH_4$ ), óxido nitroso ( $N_2O$ ), óxidos de nitrógeno ( $NO_x$ ), monóxido de carbono ( $CO$ ) y los compuestos orgánicos volátiles distintos del metano ( $COVDM$ ), así como las emisiones de dióxido de azufre.

Las emisiones de dióxido de carbono proceden de la oxidación del carbono de los combustibles durante la combustión. En condiciones de combustión óptimas, el contenido total de carbono de los combustibles debería convertirse en dióxido de carbono. Sin embargo, los procesos de combustión reales no son perfectos y la consecuencia de ello es que se producen pequeñas cantidades de carbono parcialmente oxidado y no oxidado (la oxidación incompleta se produce como consecuencia de ineficiencias en la combustión).

Debido a la combustión incompleta de los hidrocarburos en el combustible, también se liberan pequeñas proporciones de:

- Metano, las cuales indican una ineficiencia en el proceso de combustión. Las emisiones de metano dependen de la temperatura de la caldera/horno/estufa. En el caso de las instalaciones de combustión eficientes y de gran envergadura la tasa de emisión es muy baja. En fuentes

menores de combustión las tasas de emisión son, en general, más altas, sobre todo cuando se produce combustión sin llama.

- Óxido nitroso, que se producen directamente a partir de la quema de combustibles. Se ha determinado que las temperaturas de combustión más bajas producen mayores emisiones de óxido nitroso.
- Óxidos de nitrógeno, los cuales son gases de efecto invernadero indirecto. Las actividades de quema de combustible son las fuentes antropogénicas más significativas de óxidos de nitrógeno. En el ámbito de la quema de combustibles, las fuentes más importantes son las industrias energéticas y las fuentes móviles.
- Monóxido de carbono, el cual es un gas de efecto invernadero indirecto. La mayor parte de las emisiones de monóxido de carbono de la quema de combustibles procede de los vehículos motorizados. El  $CO$  es un producto intermedio del proceso de combustión y su mecanismo de formación está directamente influenciado por los patrones de uso, tipo de tecnología y tamaño, antigüedad, mantenimiento y modo de funcionamiento de la tecnología.
- Compuestos orgánicos volátiles distintos del metano, los cuales también son gases de efecto invernadero indirectos. Las emisiones de COVDM son productos de la combustión incompleta. La fuente más importante de COVDM a partir de las actividades de quema de combustibles son las fuentes móviles y la combustión residencial, especialmente la combustión de biomasa (por ejemplo, la leña). Las emisiones de las plantas de combustión a gran escala son muy bajas y tienden a disminuir a medida que aumenta el tamaño de la planta y la eficiencia del proceso de combustión.
- Dióxido de azufre, el cual es un precursor de los aerosoles cuya presencia en la atmósfera puede tener un efecto de enfriamiento climático. Las emisiones de  $SO_2$  están íntimamente relacionadas con el contenido de azufre de los combustibles.

## **2.6. EMISIONES FUGITIVAS**

Aparte de las emisiones directas producidas en la realización de procesos incluidos en la actividad de tratamiento de gas, existen las emisiones fugitivas. Las fuentes de estas emisiones, se definen como focos no asociados a ningún proceso específico, sino distribuidos por todo el recinto de la planta de tratamiento de gas.

Las emisiones fugitivas de metano de los sistemas de gas natural representan una fuente importante de contaminación por calentamiento global. Las reducciones en las emisiones de metano son necesarias con urgencia como parte de un esfuerzo más amplio para frenar la tasa de aumento de la temperatura global.

Las emisiones fugitivas de metano ocurren en cada etapa del ciclo de vida del gas natural, sin embargo, la cantidad total de la fuga no está determinada. Las emisiones fugitivas accidentales a lo largo de la red de suministro representan pérdidas de producto que podrían evitarse realizando simples cambios en las prácticas operativas y mejoras en los equipos, trayendo consigo grandes beneficios tales como: ahorros de dinero al evitar la pérdida de gas y emplear éste en la generación de energía, o como combustible; además de mejorar las condiciones de seguridad industrial de las instalaciones y contribuir con la disminución de emisiones GEI y por consiguiente con la mitigación del calentamiento global.

## **2.7. INGENIERÍA DE SOFTWARE**

Es una disciplina que se ocupa del estudio y generación de soluciones lógicas mediante el diseño, desarrollo y mantenimiento de software bajo criterios de estabilidad y calidad.

Entre los objetivos de la ingeniería de software están:

- ✓ Mejorar el diseño de aplicaciones de tal modo que se adapten de mejor manera a las necesidades de las organizaciones para las cuales fueron creadas.
- ✓ Promover mayor calidad al desarrollar aplicaciones complejas.
- ✓ Brindar mayor exactitud en los costos de proyectos y tiempo de desarrollo.
- ✓ Aumentar la eficiencia de los sistemas al introducir procesos que permitan medir mediante normas específicas, la calidad del software desarrollado, buscando siempre la mejor calidad posible según las necesidades y resultados que se quieren generar.
- ✓ Una mejor organización de equipos de trabajo, en el área de desarrollo y mantenimiento de software. Detectar a través de pruebas, posibles mejoras para un mejor funcionamiento del software desarrollado.

## **2.8. PRUEBAS DE SOFTWARE**

Las pruebas de software son un conjunto de actividades cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto a la parte interesada.

Los productos de ingeniería se pueden probar de formas como:

- Pruebas de caja negra. Realizar pruebas de forma que se compruebe que cada función es operativa.
- Pruebas de caja blanca. Desarrollar pruebas de forma que se asegure que la operación interna se ajusta a las especificaciones, y que todos los componentes internos se han probado de forma adecuada.

- Pruebas de receptividad de los usuarios, teniendo presente que sólo en la medida en que el sistema software sirva adecuadamente al sistema humano en el cual operan los usuarios del mismo, podemos asegurar la pertinencia y utilidad del software desarrollado.

## **2.9. APLICACIÓN WEB**

Una aplicación web es un tipo de software que se ejecuta en un navegador web y que permite a los usuarios interactuar con él a través de una interfaz de usuario. A diferencia de las aplicaciones tradicionales, que se instalan en el dispositivo del usuario, las aplicaciones web se almacenan en servidores remotos y se accede a ellas a través de internet.

Las aplicaciones web pueden tener una amplia variedad de funciones y propósitos, desde simples sitios web interactivos hasta complejos sistemas de gestión empresarial. Su desarrollo suele involucrar tecnologías como HTML, CSS y JavaScript para la parte del cliente, y lenguajes de programación como PHP, Python o Ruby para la parte del servidor.

Una de las principales ventajas de las aplicaciones web es que son multiplataforma, lo que significa que pueden ejecutarse en cualquier dispositivo con un navegador web, independientemente de su sistema operativo. Esto las hace accesibles y fáciles de usar para un gran número de usuarios.

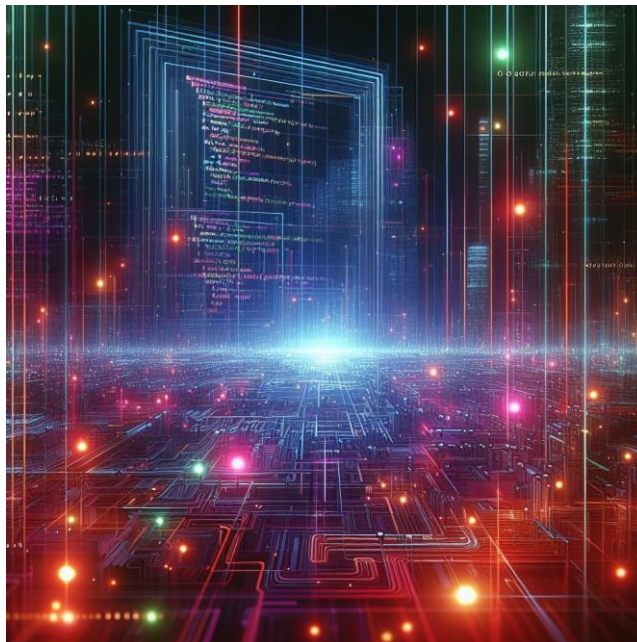
## **2.10. COMPONENTE BACKEND**

Un componente backend, también conocido como back-end, es la parte de un sistema de software que se encarga de procesar los datos y la lógica de negocio de una aplicación. En una arquitectura típica de aplicación web, el backend actúa

como el cerebro de la operación, se encarga de manejar las solicitudes de los clientes, procesar la información, interactuar con la base de datos y generar la respuesta que se envía al cliente.

El backend está compuesto por diferentes componentes, como servidores web, sistemas de gestión de bases de datos, servicios web y APIs (interfaces de programación de aplicaciones). Estos componentes trabajan juntos para proporcionar la funcionalidad necesaria para que la aplicación funcione correctamente.

El desarrollo de un componente backend suele involucrar el uso de lenguajes de programación como Java, Python, PHP o Node.js, así como el uso de frameworks y herramientas específicas para el desarrollo web. El objetivo principal del backend es proporcionar una interfaz segura y eficiente para que los usuarios interactúen con la aplicación y gestionar los datos de manera efectiva.



*Ilustración 5. Backend - imagen creada con inteligencia artificial*  
Fuente: Instrucciones en Copilot

En la anterior imagen creada con inteligencia artificial se observa de manera abstracta el concepto de backend, identificando algunos componentes claves como: servidores, lógica de la aplicación, comunicación con la base de datos y seguridad. Porque en resumen el Back-end es la parte invisible pero crucial que dota de características y funcionalidades a una aplicación web. Sin él, los sitios web y las aplicaciones serían solo páginas estáticas sin ninguna interacción o lógica detrás de ellas.

### 3. REQUERIMIENTOS DEL SISTEMA

El presente capítulo describe los requerimientos del sistema para el diseño y desarrollo del prototipo funcional del componente Backend de la aplicación web destinada a la gestión de inventarios de emisiones fugitivas de gases hidrocarburos en el sector Oil&Gas. Este proyecto, realizado en colaboración con la Corporación Centro de Desarrollo Tecnológico - CDT de Gas, tiene como objetivo principal establecer las bases para un sistema software que apoye efectivamente dicho proceso.

En esta fase de planeación, se identifican, analizan y documentan los requerimientos funcionales y no funcionales necesarios para la implementación de las funcionalidades principales de la aplicación a la medida. Se integra la información requerida para el registro y seguimiento de las emisiones fugitivas, así como la generación de informes básicos para respaldar la toma de decisiones en la gestión de las emisiones.

Para la identificación y documentación de los requerimientos del sistema, se llevó a cabo una serie de reuniones estratégicas y colaborativas con los directivos de la empresa, el ingeniero responsable de las visitas en campo, el ingeniero de sistemas, quien desempeña el rol de tutor en esta práctica empresarial y la estudiante practicante de la UIS, quien escribe este informe final.

Estas reuniones, que se realizaron de manera periódica y meticulosa, permitieron recopilar y analizar exhaustivamente las necesidades y expectativas de los diferentes actores involucrados.

Inicialmente, los resultados de estas reuniones se plasmaron en un documento de visión para el desarrollo de software (Ver Anexo A.), el cual incluye la idea en general, las reglas de negocio y las necesidades básicas del sistema.

Por otra parte, conforme se fueron incrementando las reuniones y se profundizó en el análisis, se elaboró un documento mucho más detallado y especializado en los requisitos, conocido como Documento de Especificaciones de Requerimientos del Sistema (ERS).

En cuanto a este último entregable, el Documento de Especificaciones de Requerimientos del Sistema (ERS), se presenta como anexo a este informe (Ver Anexo B.) y contiene una descripción exhaustiva de los requisitos funcionales y no funcionales del sistema, así como las clases y características de usuarios, una breve descripción del entorno operativo y los requerimientos de interfaces externas, como son la interfaz de usuario, la interfaces de hardware, las interfaces de software y las interfaces de comunicación y otros elementos necesarios para la implementación del software.

En la siguiente sección, se ofrece una descripción detallada del sistema propuesto, con el objetivo de proporcionar una visión más amplia y clara de sus funcionalidades y características.

### **3.1. DESCRIPCIÓN DEL SISTEMA**

Este sistema es el prototipo funcional del componente Backend de un aplicativo web diseñado para apoyar el proceso de gestión de inventarios de emisiones fugitivas de gases hidrocarburos en el sector Oil&Gas para uso de la Corporación Centro de Desarrollo Tecnológico del Gas – CDT de Gas. A este sistema se le ha llamado SIEGEI (Sistema de Inventario de Emisiones de Gases de Efecto Invernadero).

En particular, la aplicación proporcionará a los usuarios la capacidad de registrar, almacenar y gestionar la información relacionada con las emisiones fugitivas, incluyendo datos de empresas (clientes de CDT), plantas (o estaciones), sistemas

(o equipos) y fuentes de emisión. Cabe señalar que también ofrecerá funcionalidades para realizar cálculos de emisiones, consultar información bajo tres criterios y generar informes adecuados para respaldar la toma de decisiones en la gestión de las emisiones.

Las características con las que cuenta el sistema se describen a continuación:

Tabla 1. Características del sistema SIEGEI

ID	Característica	Descripción
C1	Gestión de usuarios, autenticación y autorización	Esta característica agrupa todos los requerimientos que se encargan de la gestión de usuarios, su autenticación dentro del sistema y de los niveles de autorización para el uso de este.
C2	Gestión de entidades	Esta característica contiene los requerimientos que permiten la creación, actualización y consultas de las entidades que utiliza el sistema como son: Empresas, Plantas, Sistemas, Fuentes de emisión y Factores de emisión
C3	Gestión de consultas, reportes y cálculos	Esta característica provee la funcionalidad que permite obtener el valor agregado del sistema y apoyar la toma de decisiones a través de la posibilidad de realizar cálculos de las emisiones, consultar la información de diversa forma y generar los reportes adecuados.

Fuente I. Elaboración propia de la autora

La anterior clasificación hace referencia a las características fundamentales para el funcionamiento del sistema, las cuales no solo facilitan el cumplimiento de las regulaciones ambientales por parte de los clientes de CDT de GAS, sino que también respaldan la toma de decisiones informadas en la corporación.

Así SIEGEI cumple su objetivo de ser un sistema diseñado para ofrecer una solución integral y eficiente para la gestión efectiva de las emisiones de gases de efecto invernadero.

### 3.2. REQUERIMIENTOS FUNCIONALES

Durante el proceso de levantamiento de requerimientos, se emplearon diversas herramientas y técnicas para recopilar información crucial. Entre estas herramientas se incluyeron entrevistas con posibles futuros usuarios del sistema, así como grupos focales con los miembros de la sede de Provenza de CDT de Gas. Estas actividades fueron fundamentales para identificar tanto los problemas existentes en el proceso actual como las oportunidades para mejorar y optimizar el sistema propuesto.

No obstante, este proceso implicó costos significativos. En términos de tiempo, la duración del proceso fue considerable, abarcando varios meses para garantizar un impacto significativo en los resultados. Asimismo, se incurrieron en costos de personal, la estudiante practicante y la participación de otros miembros del equipo, en términos de horas de trabajo dedicadas al proceso. Por último, los costos asociados a la comunicación y las reuniones fueron fundamentales para el desarrollo efectivo del proceso de levantamiento de requerimientos.

En este capítulo del informe final, se presentan los requerimientos identificados durante el proceso de levantamiento. Para aquellos lectores interesados en una visión más detallada y exhaustiva, se recomienda consultar los anexos adjuntos, donde se proporciona información más específica y detallada sobre los requerimientos del sistema.

Cabe recordar las características y/o funcionalidades del software que se describieron anteriormente en la Tabla 1. Características del sistema SIEGEI:

*Tabla 2. Funcionalidades del sistema SEGEI*

F.S. 01	Gestión de usuarios, autenticación y autorización
F.S. 02	Gestión de entidades
F.S. 03	Gestión de consultas, reportes y cálculos

*Fuente II. Elaboración propia de la autora*

La funcionalidad 01 – Gestión de usuarios, autenticación y autorización agrupa todos los requerimientos que se encargan de la gestión de usuarios, su autenticación dentro del sistema y de los niveles de autorización para el uso de este.

En el proceso de autenticación se busca identificar a los usuarios y garantizar que los mismos sean quienes dicen ser para evitar que un usuario no registrado pueda entrar en el sistema o iniciar sesión en la plataforma de forma indebida, sin que realmente sea el usuario legítimo de las credenciales correspondientes. Mientras que en el proceso de autorización se verifican los permisos que corresponden a cada identidad.

A continuación, se lista los requerimientos funcionales asociados a esta funcionalidad:

*Tabla 3. Requerimientos funcionales de la funcionalidad Gestión de usuarios*

R.F. 01	Autenticar usuario - Inicio de Sesión
R.F. 02	Registrar usuario
R.F. 03	Actualizar password de usuario

*Fuente: Elaboración propia de la autora*

La funcionalidad 02 – Gestión de entidades contiene los requerimientos que permiten la creación, actualización y consultas de las entidades que utiliza el sistema como son: Empresas, Plantas, Sistemas, Fuentes de emisión y Factores de emisión.

A continuación, se lista los requerimientos funcionales asociados a esta funcionalidad:

Tabla 4. Requerimientos funcionales de la funcionalidad Gestión de entidades

R.F. 04	Crear Empresa
R.F. 05	Consultar Empresa
R.F. 06	Actualizar Empresa
R.F. 07	Eliminar Empresa
R.F. 08	Crear Planta
R.F. 09	Consultar Planta
R.F. 10	Actualizar Planta
R.F. 11	Eliminar Planta
R.F. 12	Crear Sistema
R.F. 13	Consultar Sistema
R.F. 14	Actualizar Sistema
R.F. 15	Eliminar Sistema
R.F. 16	Crear Fuente de Emisión por Combustión
R.F. 17	Consultar Fuente de Emisión por Combustión
R.F. 18	Actualizar Fuente de Emisión por Combustión
R.F. 19	Eliminar Fuente de Emisión por Combustión
R.F. 20	Crear Fuente de Emisiones Fugitivas
R.F. 21	Consultar Fuente de Emisiones Fugitivas
R.F. 22	Actualizar Fuente de Emisiones Fugitivas
R.F. 23	Eliminar Fuente de Emisiones Fugitivas
R.F. 24	Crear Composición Detallada
R.F. 25	Consultar Composición Detallada
R.F. 26	Actualizar Composición Detallada
R.F. 27	Eliminar Composición Detallada

R.F. 28	Crear Evidencia
R.F. 29	Consultar Evidencia
R.F. 30	Actualizar Evidencia
R.F. 31	Eliminar Evidencia
R.F. 32	Crear Archivos Multimedia
R.F. 33	Consultar Archivos Multimedia
R.F. 34	Actualizar Archivos Multimedia
R.F. 35	Eliminar Archivos Multimedia

*Fuente: Elaboración propia de la autora*

La funcionalidad 03 – Gestión de consultas, reportes y cálculos provee la funcionalidad que permite obtener el valor agregado del sistema y apoyar la toma de decisiones a través de la posibilidad de realizar cálculos de las emisiones, consultar la información de diversa forma y generar los reportes adecuados.

En estos procesos se busca permitir a los usuarios realizar consultas específicas, ofrecer la posibilidad de generar informes detallados; así mismo el sistema tiene funcionalidades como: realizar cálculos automáticos para determinar las emisiones totales con el fin de facilitar la toma de decisiones basada en datos concretos.

A continuación, se lista los requerimientos funcionales asociados a esta funcionalidad.

*Tabla 5. Requerimientos funcionales de la funcionalidad Gestión de consultas, reportes y cálculos*

R.F. 36	Crear Reporte Mensual por Tipo de Gas (detalle sistema)
R.F. 37	Consultar Reporte Mensual por Tipo de Gas (detalle sistema)
R.F. 38	Actualizar Reporte Mensual por Tipo de Gas (detalle sistema)
R.F. 39	Eliminar Reporte Mensual por Tipo de Gas (detalle sistema)

R.F. 40	Crear Reporte Mensual por Planta (detalle gas)
R.F. 41	Consultar Reporte Mensual por Planta (detalle gas)
R.F. 42	Actualizar Reporte Mensual por Planta (detalle gas)
R.F. 43	Eliminar Reporte Mensual por Planta (detalle gas)
R.F. 44	Crear Reporte Anual Acumulado por Empresa y Planta
R.F. 45	Consultar Reporte Anual Acumulado por Empresa y Planta
R.F. 46	Actualizar Reporte Anual Acumulado por Empresa y Planta
R.F. 47	Eliminar Reporte Anual Acumulado por Empresa y Planta
R.F. 48	Calcular Factor de Emisión por Combustión
R.F. 49	Calcular Factor de Emisiones Fugitivas
R.F. 50	Calcular Emisiones por Combustión
R.F. 51	Calcular Emisiones Fugitivas
R.F. 52	Calcular Emisiones Totales

*Fuente: Elaboración propia de la autora*

Ahora veamos alguna información en particular de los diagramas UML realizados a partir de los requerimientos definidos y mostrados anteriormente.

Inicialmente se muestra las especificaciones de cuatro requerimientos funcionales del sistema en específico:

- R.F. 01 - Autenticar usuario – inicio de sesión
- R.F. 14 - Actualizar sistema
- R.F. 33 - Consultar Archivos Multimedia
- R.F. 37 - Consultar reporte mensual por tipo de gas (detalle sistema)

Luego, se muestra los diagramas de casos de uso de las tres funcionalidades del sistema:

- ❖ Gestión de usuarios, autenticación y autorización
- ❖ Gestión de entidades
- ❖ Gestión de consultas, reportes y cálculos

Y, por último, se detallan tres casos de uso en específico, describiendo sus acciones principales, los actores involucrados y los resultados esperados:

- Registro de usuarios
- Creación de fuente de emisión por combustión
- Generación de reporte anual acumulado por empresa y planta.

Estos esquemas y tablas proporcionarán una visión clara y concisa de las funcionalidades principales del sistema, facilitando su comprensión por parte de los lectores.

*Tabla 6. Especificación de requerimiento funcional RF.01- Autenticar usuario - inicio de sesión*

Identificación del requerimiento:	<b>R.F. 01</b>	Tipo de requerimiento:	Funcional
Nombre del requerimiento:	Autenticar usuario - Inicio de Sesión		
Prioridad del requerimiento	Baja	Caso de uso relacionado:	
Características:	Los usuarios deberán identificarse para acceder al sistema.		
Descripción del requerimiento:	El usuario debe contar con las credenciales de acceso suministradas para poder acceder a la información almacenada en el aplicativo web		
Entradas:	<ul style="list-style-type: none"> <li>• Login</li> <li>• Password</li> </ul>	Salidas:	Ingreso al sistema
Fuente:	Formulario de ingreso de datos	Destino:	Base de datos
Requerimiento NO funcional:	<ul style="list-style-type: none"> <li>- R.N.F. 05</li> <li>- R.N.F. 07</li> </ul>	Restricciones:	Cada usuario tendrá un rol como mínimo

*Fuente: Elaboración propia de la autora*

Tabla 7. Especificación de requerimiento funcional RF.14- Actualizar sistema.

Identificación del requerimiento:	<b>R.F. 14</b>	Tipo de requerimiento:	Funcional
Nombre del requerimiento:	Actualizar Sistema		
Prioridad del requerimiento	Alta	Caso de uso relacionado:	
Características:	El sistema debe permitir modificar la información registrada para un sistema o equipo		
Descripción del requerimiento:	El usuario ingresará en la opción de Administración de Sistema, en donde aparecerá un listado de los sistemas o equipos creados o podrá buscarlo a través del código del sistema, si lo conoce. El usuario tendrá la facultad de modificar la información registrada para el Sistema o Equipo si da clic en el icono correspondiente a esta función. El sistema abrirá el formulario con los datos que se permiten actualizar, el usuario introducirá los datos correspondientes y grabará los cambios al presionar el botón de Guardar.		
Entradas:	<ul style="list-style-type: none"> <li>• Código Sistema</li> <li>• Nombre Sistema</li> </ul>	Salidas:	Confirmación por pantalla
Fuente:	Formulario de ingreso de datos: - Nombre del sistema - Descripción	Destino:	Base de datos
Requerimiento NO funcional:	- R.N.F. 02 - R.N.F. 05	Restricciones:	No se eliminará el sistema sin confirmación

Fuente: Elaboración propia de la autora

Tabla 8. Especificación de requerimiento funcional RF.33 - Consultar archivos multimedia

Identificación del requerimiento:	<b>R.F. 33</b>	Tipo de requerimiento:	Funcional
Nombre del requerimiento:	Consultar Archivos Multimedia		
Prioridad del requerimiento	Baja	Caso de uso relacionado:	
Características:	El sistema debe permitir consultar los archivos multimedia registrados como evidencia en la creación de fuentes de emisión.		
Descripción del requerimiento:	El usuario podrá ver un elemento multimedia, llámese video en formato .mp4 y/o imágenes en formato PNG, JPG y JPEG, agregado como evidencia a la creación de fuentes de emisión, desde menú de consultas de las fuentes de emisión. Simplemente escogerá cuál es el tipo de emisión que desea consultar (por combustión o emisiones fugitivas) y luego en la lista de emisiones habrá un ícono de visualización al lado de cada una; con solo dar clic en el ícono correspondiente a la emisión que contenga la evidencia que desea consultar podrás visualizarla.		
Entradas:	<ul style="list-style-type: none"> <li>• Menú consulta</li> </ul>	Salidas:	Confirmación por pantalla
Fuente:	Base de datos	Destino:	Módulo consultas

Requerimiento NO funcional:	- R.N.F. 05 - R.N.F. 07	Restricciones:	Esta opción sólo permite consulta, no eliminación ni actualización.
-----------------------------	----------------------------	----------------	---

Fuente: Elaboración propia de la autora

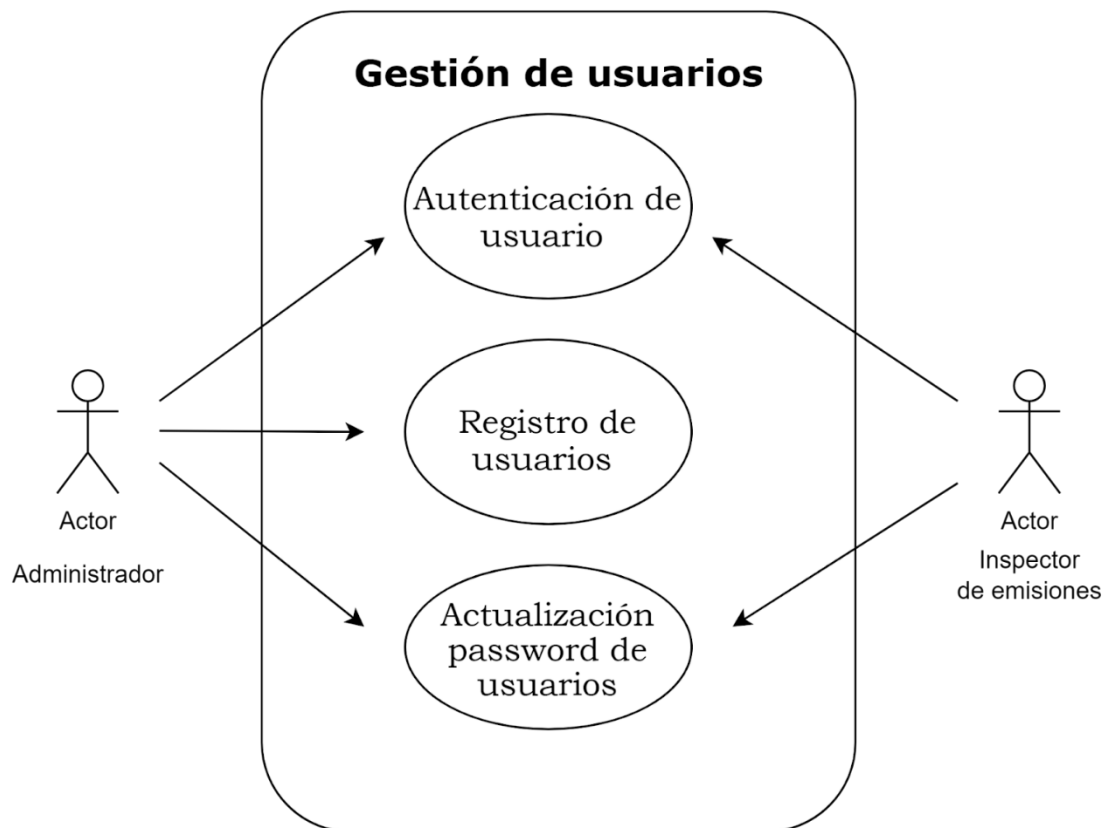
Tabla 9. Especificación de requerimiento funcional RF.37 - Consultar reporte mensual por tipo de gas

Identificación del requerimiento:	<b>R.F. 37</b>	Tipo de requerimiento:	Funcional
Nombre del requerimiento:	Consultar Reporte Mensual por Tipo de Gas (detalle sistema)		
Prioridad del requerimiento	Baja	Caso de uso relacionado:	
Características:	El sistema debe permitir la consulta de reportes guardados, en este caso de reportes detallados por mes por tipo de gas asociado a los sistemas donde se registre su fuga.		
Descripción del requerimiento:	El usuario ingresará en la sección de Gestión de reportes, en donde aparecerá las opciones de reporte que se pueden generar con el sistema. El usuario escogerá la opción de "Reporte mensual por tipo de gas (detalle sistema)", a continuación, el sistema mostrará un listado de los reportes creados y guardados previamente en esta sección. El usuario con un simple clic, sobre el reporte que desee consultar, podrá visualizar la información registrada en el reporte escogido. En esta opción no podrá realizar ningún cambio o alteración a los datos almacenados.		
Entradas:	<ul style="list-style-type: none"> <li>● Gestión de reportes</li> <li>● Reporte mensual por tipo de gas (detalle sistema)</li> </ul>	Salidas:	Confirmación por pantalla
Fuente:	Base de datos	Destino:	Interfaz de usuario
Requerimiento NO funcional:	- R.N.F. 03 - R.N.F. 04 - R.N.F. 05	Restricciones:	En la opción de consulta, no se podrá modificar y/o eliminar el reporte.

Fuente: Elaboración propia de la autora

### 3.2.1. DIAGRAMAS DE CASOS DE USO

En esta sección se presentan tres casos de uso, de las principales funcionalidades del sistema.



*Ilustración 6. Diagrama de caso de uso de gestión de usuarios  
Fuente: Elaboración propia de la autora*

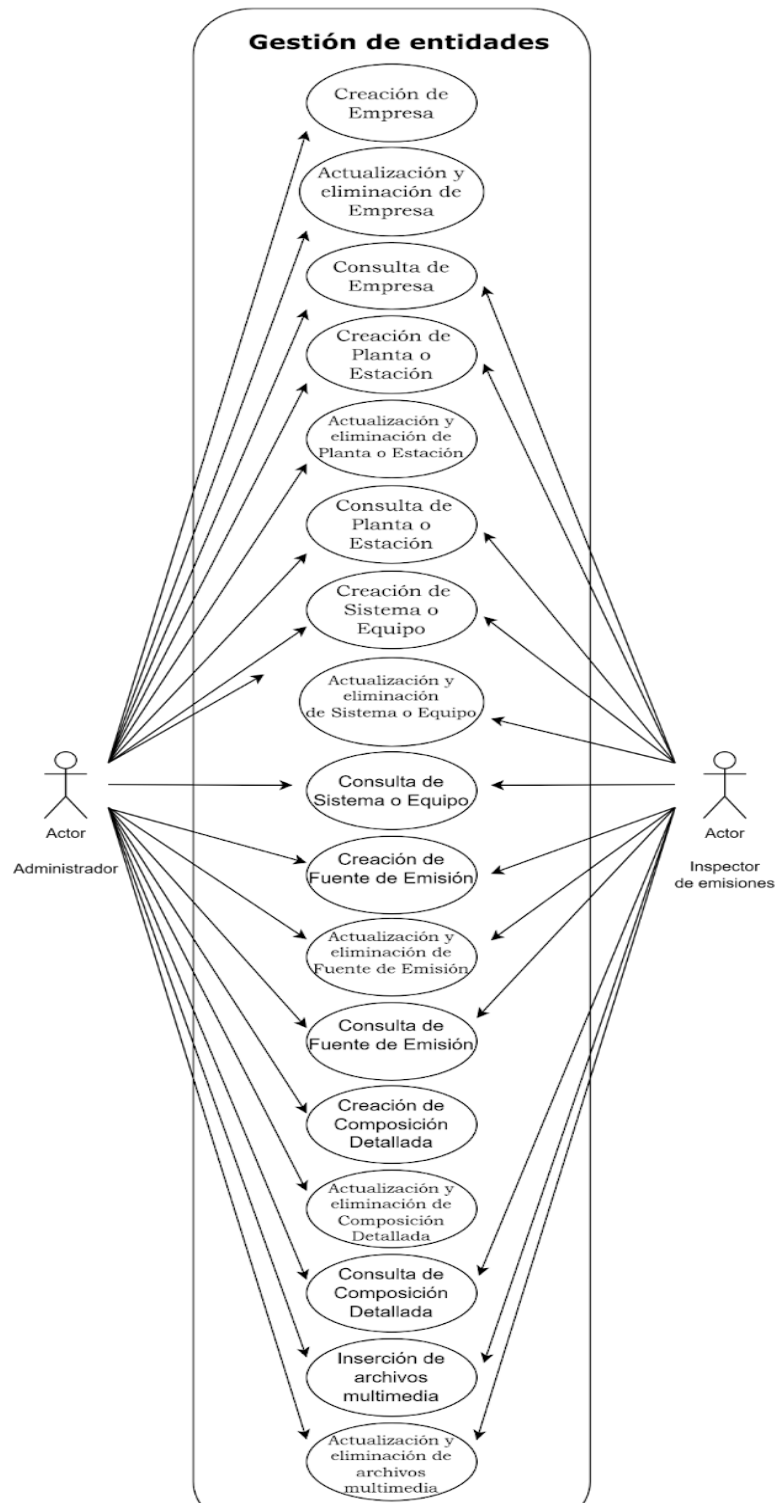


Ilustración 7. Diagrama de caso de uso de gestión de entidades  
Fuente: Elaboración propia de la autora

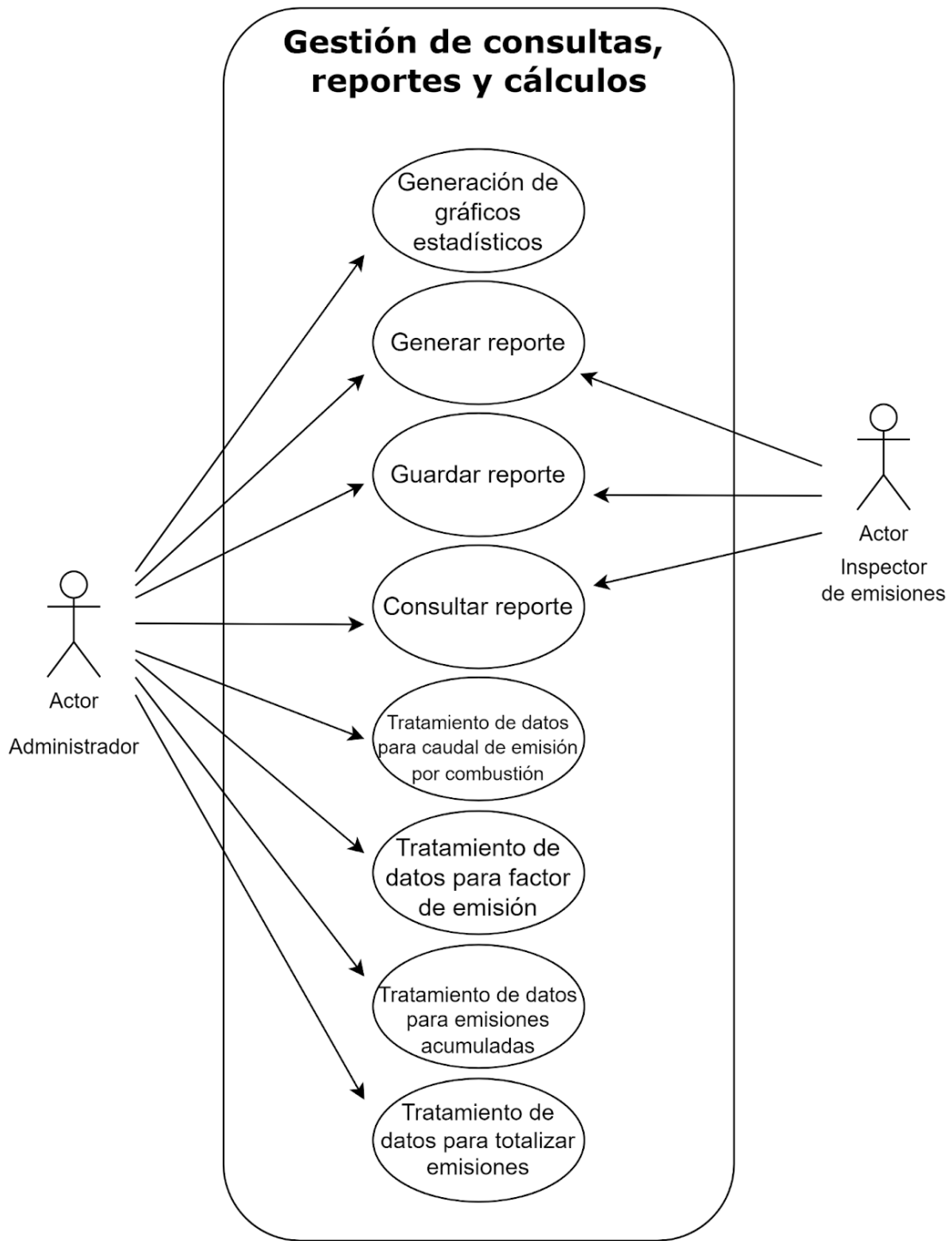


Ilustración 8. Diagrama de caso de uso de gestión de consultas, reportes y cálculos  
Fuente: Elaboración propia de la autora

## 3.2.2. ESPECIFICACIÓN DE ALGUNOS CASOS DE USO

### 3.2.2.1. Gestión de Usuarios

Tabla 10. Especificación de caso de uso para registro de usuarios

<b>Caso de Uso</b>	<b>Registro de usuarios</b>	<b>Identificador:</b> CU002
<b>Actores</b>	Usuario Sistema	
<b>Tipo</b>	Primario	
<b>Referencias</b>	F.S. 01 - Gestión de usuarios, autenticación y autorización R.F. 02 - Registrar usuario	
<b>Precondición</b>	El usuario debe acceder al aplicativo web a través de un explorador o navegador previamente instalado en su computador El usuario no debe estar registrado previamente en el sistema. Los datos ingresados por el usuario deben ser válidos según las reglas del sistema.	
<b>Postcondición</b>	Se crea una cuenta de usuario en el sistema. El usuario puede autenticarse en el sistema con las credenciales registradas.	
<b>Extensiones</b>	Si los datos ingresados para el registro no son válidos, el sistema muestra un mensaje de error y solicita correcciones.	
<b>Descripción</b>	Este caso de uso aborda el proceso de registro de usuarios en el sistema, permitiendo a los usuarios (usuarios operativos y administradores) registrarse para obtener credenciales y poder utilizarlo. Durante el registro, el usuario debe suministrar datos como: ID, Nombres, Apellidos, Cargo, Cédula, E-mail, Usuario y Password.	

### Curso Normal

Nro.	Ejecutor	Paso o Actividad
1	Usuario	El usuario accede al aplicativo desde su navegador web
2	Sistema	El sistema muestra la pantalla de inicio

3	Usuario	El usuario solicita registrarse en el sistema.
4	Sistema	El sistema presenta el formulario de registro de usuario.
5	Usuario	El usuario completa los campos del formulario (ID, Nombres, Apellidos, Cargo, Cédula, E-mail, Usuario y Password).
6	Sistema	El sistema valida los datos ingresados.
7	Sistema	Si los datos son válidos, el sistema crea una cuenta de usuario y muestra un mensaje de éxito.
8	Usuario	El usuario recibe un correo electrónico de confirmación de registro.
9	Usuario	El usuario confirma su registro haciendo clic en el enlace proporcionado en el correo electrónico.

## Cursos Alternos

Nro.	Descripción de acciones alternas
1	El usuario no recibe el correo electrónico de confirmación y solicita reenviarlo.
2	El usuario intenta registrarse con un correo electrónico que ya está en uso en el sistema y recibe un mensaje de error.

## Excepciones

Nro.	Descripción de acciones alternas
1	El usuario intenta registrarse con un correo electrónico que ya está en uso en el sistema
2	Usuario registrado en el sistema
3	Posible usuario declarándose con dos roles en el sistema
4	Datos obligatorios incompletos

### 3.2.2.2. Gestión de Entidades

Tabla 11. Especificación de caso de uso para creación de fuente de emisión por combustión

<b>Caso de Uso</b>	<b>Creación de Fuente de Emisión por Combustión</b>	<b>Identificador:</b> CU016
<b>Actores</b>	Usuario Sistema	
<b>Tipo</b>	Primario	
<b>Referencias</b>	F.S. 02 - Gestión de entidades R.F. 16 - Crear Fuente de Emisión por Combustión	
<b>Precondición</b>	El usuario debe tener los permisos adecuados para administrar las fuentes de emisión por combustión. Los datos ingresados por el usuario para la creación de la fuente de emisión por combustión deben ser válidos según las reglas del sistema.	
<b>Postcondición</b>	Se crea un registro de fuente de emisión por combustión en la base de datos.	
<b>Extensiones</b>	Si la información necesaria para crear una fuente de emisión por combustión no está completa, el sistema muestra un mensaje de error y solicita que se completen los datos faltantes.	
<b>Descripción</b>	Este caso de uso aborda la administración de las fuentes de emisión por combustión en el sistema, permitiendo la creación de registros de cada fuente de emisión obtenida en las visitas de campo. El sistema debe permitir registrar los datos necesarios para la creación de una fuente de emisión, y el usuario tendrá la opción de administrar las fuentes de emisión por combustión, seleccionando la empresa, la estación y el sistema (si corresponde) a los que se le va a asociar dicha fuente de emisión.	

#### Curso Normal

Nro.	Ejecutor	Paso o Actividad
1	Usuario	El usuario solicita administrar las fuentes de emisión por combustión.

2	Sistema	El sistema presenta la opción para crear una nueva fuente de emisión.
3	Usuario	<p>El usuario completa los datos necesarios para la creación de la fuente de emisión (empresa, estación, sistema, etc.):</p> <ul style="list-style-type: none"> <li>• ID</li> <li>• TAG</li> <li>• Descripción</li> <li>• Observación</li> <li>• Tipo de combustible (selecciona tipo de gas)</li> <li>• Factor de emisión</li> </ul> <p>Opción 1 – Simple</p> <ul style="list-style-type: none"> <li>• Potencia</li> <li>• Horas de operación</li> </ul> <p>Opción 2 - Detallada</p> <ul style="list-style-type: none"> <li>• Eficiencia de combustión</li> <li>• Consumo de combustible</li> </ul>
4	Sistema	El sistema verifica que la información necesaria esté completa.
5	Sistema	Si la información está completa, el sistema crea el registro en la base de datos y lo clasifica como fuente de emisión por combustión.
6	Sistema	El sistema presenta al usuario una pantalla de confirmación de los datos ingresados.
7	Usuario	El usuario confirma la creación de la fuente de emisión por combustión.
8	Sistema	El sistema guarda la información en la base de datos.

## Cursos Alternos

Nro.	Descripción de acciones alternas
1	El usuario cancela la creación de la fuente de emisión.
2	El usuario modifica los datos de una fuente de emisión existente.

## Excepciones

Nro.	Descripción de acciones alternas
------	----------------------------------

1	Datos incompletos: Si el usuario intenta crear una fuente de emisión sin proporcionar todos los datos requeridos, el sistema mostrará un mensaje de error y no permitirá la creación hasta que todos los campos obligatorios estén completos.
2	Datos inválidos: Si los datos proporcionados por el usuario no cumplen con las validaciones requeridas (por ejemplo, formato de correo electrónico incorrecto, valor numérico fuera de rango), el sistema mostrará un mensaje de error y solicitará que se corrijan los datos.
3	Falta de permisos: Si el usuario no tiene los permisos necesarios para administrar las fuentes de emisión, el sistema mostrará un mensaje indicando que no está autorizado para realizar esa acción.
4	Cancelación por parte del usuario: Si el usuario decide cancelar la creación de la fuente de emisión en cualquier momento del proceso, el sistema abortará la operación y volverá al estado anterior, sin crear el registro en la base de datos.

### 3.2.2.3. Gestión de Consultas, Reportes y Cálculos

Tabla 12. Especificación de caso de uso para generación de reporte anual acumulado por empresa y planta

<b>Caso de Uso</b>	<b>Generación de reporte anual acumulado por empresa y planta</b>	<b>Identificador: CU044</b>
Actores	Usuario Sistema	
Tipo	Secundario	
Referencias	F.S. 03 - Gestión de consultas, reportes y cálculos R.F. 44 - Crear Reporte Anual Acumulado por Empresa y Planta	
Precondición	El usuario debe tener los permisos adecuados para generar y administrar reportes. La opción ingresada por el usuario para la generación del reporte debe ser válida según las reglas del sistema, en este caso, escoge la opción de reporte anual acumulado por empresa y planta.	
Postcondición	Se genera un reporte con los datos específicos seleccionados por el usuario, un reporte anual de las emisiones totales acumuladas clasificadas por empresa y por planta.	
Extensiones	El usuario puede incluir gráficos estadísticos en el reporte, si así lo desea y si el sistema le brinda la opción. El usuario puede exportar el reporte a diferentes formatos (como PDF).	
Descripción	Este caso de uso permite apoyar la toma de decisiones a través de la generación de reportes. Los usuarios pueden realizar consultas específicas y generar informes detallados. El sistema presenta los datos de emisiones fugitivas de manera visual. Además, el sistema permite administrar los reportes generados por el usuario, incluir gráficos estadísticos en los reportes (si el usuario lo desea) y guardarlos en la base de datos para su posterior consulta.	

## Curso Normal

Nro.	Ejecutor	Paso o Actividad
1	Usuario	El usuario ingresa a la opción de Gestión de Consultas, Reportes y Cálculos.
2	Sistema	El sistema presenta las opciones para generar reportes según ciertas especificaciones, que son: <ul style="list-style-type: none"><li>• Crear reporte mensual por tipo de gas (detalle sistema)</li><li>• Crear reporte mensual planta (detalle gas)</li><li>• Crear reporte anual acumulado por empresa y planta</li></ul>
3	Usuario	El usuario selecciona las opciones deseadas para el reporte, que en este caso de uso es: Crear reporte anual acumulado por empresa y planta.
4	Sistema	El sistema muestra un scroll para escoger la empresa de la que el usuario desea generar el reporte.
5	Usuario	El usuario escoge la empresa de la que quiere el reporte del acumulado anual de emisiones fugitivas.
6	Sistema	El sistema muestra un scroll para escoger la planta asociada a dicha empresa de la que el usuario desea generar el reporte.
7	Usuario	El usuario escoge la planta o estación de la que quiere el reporte del acumulado anual de emisiones fugitivas, asociada a la empresa que previamente se escogió en esta consulta.
8	Sistema	El sistema genera el reporte con los datos específicos seleccionados. En este caso de uso, presenta un reporte anual de las emisiones acumuladas durante un año de la planta o estación asociada a la empresa, escogidas previamente por el usuario.
9	Usuario	El usuario visualiza el reporte generado para su análisis y posterior toma de decisiones.

## Cursos Alternos

Nro.	Descripción de acciones alternas
1	El usuario cancela la generación del reporte en cualquier momento del proceso.

## Excepciones

Nro.	Descripción de acciones alternas
1	Falta de permisos: Si el usuario no tiene los permisos necesarios para generar o administrar reportes, el sistema mostrará un mensaje indicando que no está autorizado para realizar esa acción.
2	Cancelación por parte del usuario: Si el usuario decide cancelar la generación del reporte en cualquier momento del proceso, el sistema abortará la operación y no se generará el reporte.
3	Fallo en la generación del reporte: Si ocurre un error durante la generación del reporte (por ejemplo, problemas de conexión, error en el cálculo de datos), el sistema mostrará un mensaje indicando que la generación del reporte ha fallado y proporcionará detalles sobre el error.

### 3.3. REQUERIMIENTOS NO FUNCIONALES

Dado que el sistema debe cumplir con esquemas mínimos de disponibilidad del 90% del tiempo de uso estimado, se debe contemplar el diseño basado en web que debe presentar el look and feel institucional de la Corporación Centro de Desarrollo Tecnológico - CDT de GAS.

El sistema debe permitir su navegación a través de los exploradores más comunes como Google Chrome, Microsoft Edge y Mozilla Firefox. Otra característica de la interfaz de usuario es que debe también ser autoajutable a cualquier tamaño y resolución de pantalla del consumidor dependiendo del dispositivo.

Es importante destacar que el navegador utilizado no deberá requerir modificaciones ni la instalación de complementos u otros elementos adicionales para el funcionamiento del software. Asimismo, el sistema debe ser fácil de operar para el usuario, sin necesidad de soporte técnico adicional.

*Tabla 13. Requerimientos no funcionales del sistema*

R.I.U. 01	El sistema debe permitir el acceso concurrente de 10 usuarios vía web.
R.I.U. 02	El sistema debe ofrecer compatibilidad de aplicaciones y servicios con navegadores como Google Chrome, Microsoft Edge y Mozilla Firefox.
R.I.U. 03	El sistema debe ofrecer compatibilidad y ofrecer servicios a través de sistemas móviles como teléfonos inteligentes y tablets, para Apple, Android y Windows Phone, a nivel de plataforma Web.
R.I.U. 04	Se ha de implementar en el sistema una interfaz de usuario sencilla y completa, orientada a los tres tipos de usuarios (administrador, registrado y estándar o anónimo)
R.I.U. 05	El sistema debe contar con documentación de usuario en idioma español
R.I.U. 06	El sistema debe cumplir con esquemas de disponibilidad de mínimo 90% de tiempo de uso estimado.
R.I.U. 07	El sistema debe estar alineado con la imagen corporativa de la Corporación CDT de GAS.
R.I.U. 08	Las pantallas del sistema y pantallas de errores deben estar en idioma español.

### 3.4. CLASES Y CARACTERÍSTICAS DE USUARIOS

Este aplicativo web está diseñado para satisfacer las necesidades de dos clases principales de usuarios: el Administrador y el Usuario Operativo (Inspector de Emisiones). Cada clase de usuario tiene características específicas que determinan su rol, privilegios y funcionalidades permitidas dentro del sistema. A continuación, se detallan las características y funciones de cada clase de usuario:

#### **Administrador:**

- **Descripción:** El Administrador es responsable de supervisar y gestionar todas las funcionalidades del sistema. Esta clase de usuario actúa como administrador de la información, siendo tanto consumidor como productor de datos relevantes para la gestión de inventarios.
- **Privilegios:** El Administrador tiene privilegios de nivel alto, lo que le permite acceder a todas las funcionalidades del sistema sin restricciones. Esto incluye la capacidad de modificar configuraciones, añadir o eliminar usuarios, y gestionar la información de manera integral.
- **Tipo de Usuario:** Usuario Administrativo.
- **Funcionalidades Permitidas:** El Administrador tiene acceso completo a las funcionalidades del sistema, incluyendo la visualización y edición de los datos de inventario, la generación de informes y la configuración del sistema.

#### **Usuario Operativo (Inspector de Emisiones):**

- **Descripción:** El Usuario Operativo es aquel que, mediante un ID de usuario y una clave de acceso obtenidos con previo registro, tiene derechos especiales de ingreso al sistema. Su función principal es la de realizar inspecciones y reportar información sobre las emisiones fugitivas de gases hidrocarburos.

- **Privilegios:** A diferencia del Administrador, el Usuario Operativo tiene funcionalidades permitidas con ciertas restricciones. Esto limita su acceso a ciertas áreas y funcionalidades del sistema, garantizando que solo pueda realizar acciones específicas relacionadas con su rol de inspector.
- **Tipo de Usuario:** Usuario Final Directo.
- **Funcionalidades Permitidas:** El Usuario Operativo puede acceder a funcionalidades específicas relacionadas con la inspección de emisiones, como la introducción de datos de campo, la generación de informes de inspección y la comunicación de hallazgos al Administrador.

Tabla 14. Clases de usuarios del sistema y sus respectivas características

<b>CLASES DE USUARIOS CARACTERÍSTICAS</b>	<b>ADMINISTRADOR</b>	<b>USUARIO OPERATIVO (Inspector de emisiones)</b>
<b>Descripción:</b>	Persona que ha superado las fases de desarrollo del software y tiene los permisos para gestionar todas las funcionalidades del sistema.	Persona que tiene los derechos especiales de ingreso al sistema mediante un ID de usuario y una clave de acceso, obtenidos con previo registro .
<b>Rol:</b>	Administrador de la información	Consumidor y productor de información
<b>Privilegios:</b>	Nivel Alto	Nivel Medio
<b>Tipo de Usuario:</b>	Usuario Administrativo	Usuario Final Directo
<b>Funcionalidades o características del sistema permitidas:</b>	<ul style="list-style-type: none"> <li>• F.S. 01</li> <li>• F.S. 02</li> <li>• F.S. 03</li> </ul>	<ul style="list-style-type: none"> <li>• F.S. 01 (con restricciones)</li> <li>• F.S. 02 (con restricciones)</li> <li>• F.S. 03 (con restricciones)</li> </ul>

Fuente: Elaboración propia de la autora

Ambas clases de usuarios desempeñan roles fundamentales en el uso efectivo y eficiente del sistema, contribuyendo así a la gestión eficaz de este prototipo funcional para la gestión de inventarios de emisiones fugitivas en el sector Oil&Gas.

### 3.5. ENTORNO OPERATIVO

El entorno operativo de la aplicación web para la gestión de inventarios de emisiones fugitivas de gases hidrocarburos en el sector Oil&Gas se compone de dos módulos principales: el módulo de emisiones por combustión para usos energéticos y el módulo de emisiones fugitivas. Estos módulos forman parte del proyecto de grado en modalidad práctica empresarial desarrollado por una estudiante de Ingeniería de Sistemas de la Universidad Industrial de Santander, en colaboración con la empresa Corporación Centro de Desarrollo Tecnológico del Gas.

El módulo de emisiones por combustión para usos energéticos se centra en el registro y seguimiento de las emisiones generadas por procesos de combustión en instalaciones energéticas. Por otro lado, el módulo de emisiones fugitivas se enfoca en el control y gestión de las emisiones que se escapan o fugan durante los procesos industriales.

El entorno operativo de la aplicación web también debe considerar las posibles amenazas y errores involuntarios de configuración que podrían afectar su funcionamiento. Entre las amenazas más comunes se encuentran:

- **Ataques cibernéticos:** La aplicación puede ser vulnerable a ataques como inyecciones SQL, cross-site scripting (XSS) o ataques de denegación de servicio (DDoS), que podrían comprometer la seguridad y la integridad de los datos.
- **Errores de configuración:** Configuraciones incorrectas en el servidor web, en la base de datos o en otros componentes de la aplicación pueden provocar fallos en el sistema o vulnerabilidades de seguridad.

→ **Fallas en la red:** Problemas en la red, como cortes de conexión o interferencias, pueden afectar la disponibilidad y el rendimiento de la aplicación.

Para mitigar estas amenazas, es importante implementar medidas de seguridad y buenas prácticas de configuración, como:

- ♣ Mantener actualizados todos los componentes de la aplicación, incluyendo el sistema operativo, el servidor web y la base de datos.
- ♣ Utilizar firewalls y sistemas de detección de intrusiones para proteger la aplicación contra ataques cibernéticos.
- ♣ Realizar copias de seguridad periódicas de los datos para evitar la pérdida de información en caso de fallo del sistema.
- ♣ Implementar políticas de seguridad que incluyan la educación y concientización de los usuarios sobre buenas prácticas de seguridad informática.

Además, es importante realizar pruebas de seguridad de forma regular para identificar posibles vulnerabilidades y corregirlas antes de que sean explotadas por posibles atacantes.

Para conocer las especificaciones detalladas de software y hardware requeridas para el funcionamiento de la aplicación, se recomienda consultar la sección 9 (Requerimientos de interfaces externas) del documento Anexo B. Documento de Especificación de Requerimientos del Sistema, donde se detallan los requisitos necesarios para garantizar un funcionamiento óptimo y eficiente de la aplicación en su entorno operativo.

## 4. DISEÑO DEL SISTEMA

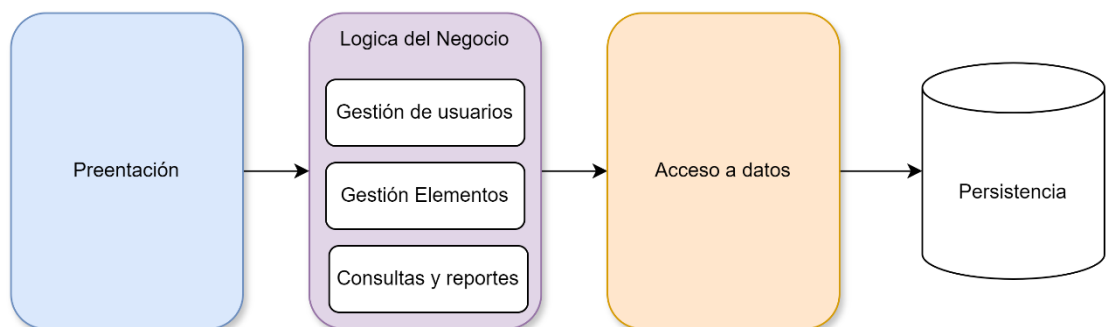
### 4.1. ARQUITECTURA DEL SISTEMA

En esta sección se presentará la arquitectura del sistema software de inventario de controles de emisiones fugitivas, es importante aclarar que las decisiones arquitecturales han sido tomadas basadas en el hecho que se va a realizar como una aplicación web SPA cuya funcionalidad Backend será desplegada como un monolito.

Con el fin de entender mejor la arquitectura se presenta dos vistas, la vista lógica que muestra cómo se organiza lógicamente los componentes del sistema y la vista de despliegue que presenta como son desplegados los componentes en la infraestructura de cómputo al momento de la ejecución del sistema.

#### VISTA LÓGICA

El sistema se ha organizado basado en el estilo por capas, en este caso la arquitectura presenta 4 capas descritas a continuación:



*Ilustración 9. Vista lógica de la arquitectura del sistema  
Fuente: Elaboración propia de la autora*

**Persistencia:** Esta capa se encarga de almacenar los datos que se ingresan al sistema y permite su consulta. En este sistema se han decidido usar dos modelos de persistencia, uno basado en una base de datos relacional para almacenar la mayoría de la información agregada y otro en forma de sistema de archivos para almacenar las imágenes del sistema.

**Acceso a datos:** Esta capa permite a la aplicación hacer abstracción de la tecnología utilizada para la persistencia (por ejemplo, SQL Server) y presenta a la lógica de negocios una vista de los datos en un modelo orientado a objetos que pueda ser fácilmente manipulado esa capa.

Esta capa se encarga de realizar las funcionalidades centrales de la aplicación y refuerza las reglas asociadas al negocio, se encarga de realizar la gestión de las entidades como empresa, sistema, plantas, emisiones. Igualmente se encarga de la gestión de las consultas y cálculos asociados a las emisiones. En esta capa se distinguen tres componentes que la conforman, descritos a continuación:

- Gestión de Usuarios: este componente contiene la lógica asociada a la gestión de usuarios, incluida su creación y actualización, y la gestión de la autenticación de los usuarios en la aplicación con un sistema de usuario-clave, y también se responsabiliza de gestionar autorizaciones en el sistema según el rol del usuario que accede.
- Gestión de Entidades: este componente se encarga de la gestión de ciclo de vida de las entidades que forman parte del sistema, dentro de estas entidades se distinguen empresa, planta, sistema, emisiones, etc.
- Gestión de Consultas y Reportes: este componente se encarga de las consultas que proveen el valor agregado para la toma de decisiones, así como la generación de reportes para presentar información de alto nivel.

**Presentación:** Esta capa se encarga de presentar la interfaz de usuario vía una interfaz gráfica de usuario (IGU) definida en HTML, con hojas de estilo CSS. En cuanto a lo que corresponde a la lógica de programación de la capa de presentación se realiza en JavaScript a través del framework Angular.

## VISTA DE DESPLIEGUE

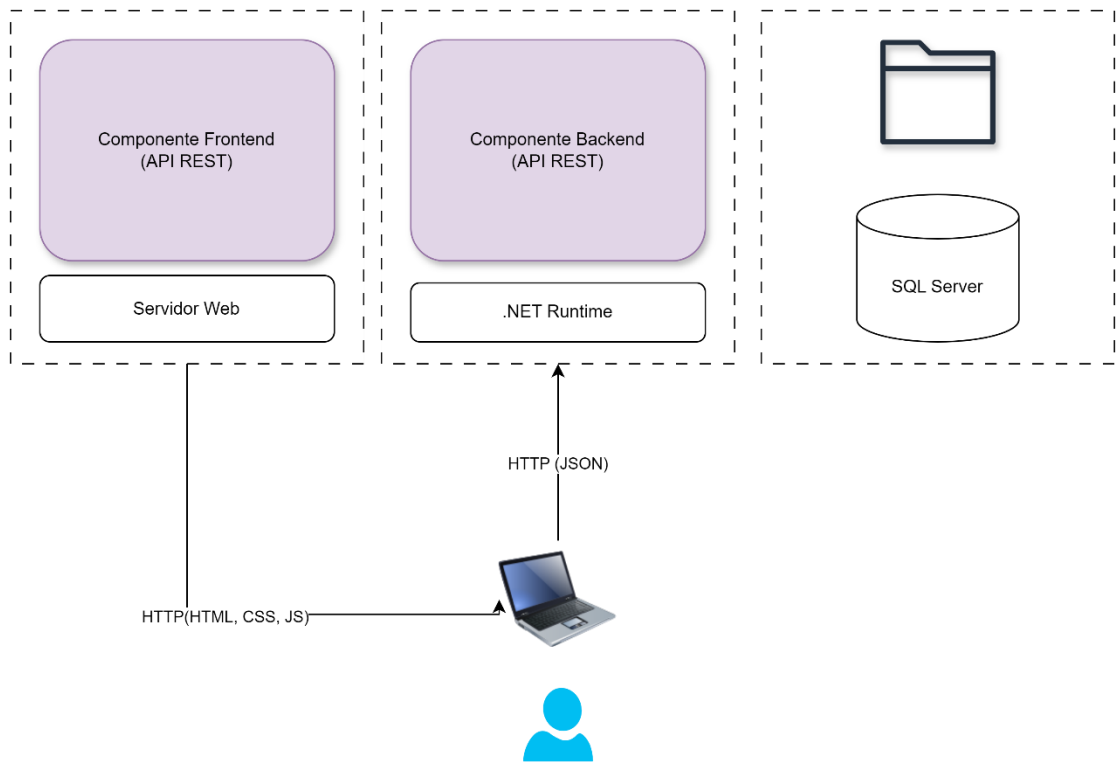
En la vista arquitectural de despliegue la aplicación se diseñó para ser desplegada en tres nodos computo, aunque es posible desplegarlo en un único nodo con todos los componentes necesarios.

**Nodo de persistencia:** En este nodo se debe desplegar la base de datos SQL Server versión 2022 y crear el esquema de base de datos correspondiente al sistema desarrollado (*ver sección 4.2. Diseño de la persistencia*). También en este nodo se debe poner a disposición un folder compartido de un sistema de archivos para almacenar las imágenes que se recopilen en la aplicación, correspondientes a las evidencias.

**Nodo de Backend:** En este nodo se debe desplegar el runtime de .Net versión 8.0 que sirve de soporte de ejecución de la aplicación. Además, se instala el componente Backend del sistema que da soporte a los componentes lógicos de Lógica de Negocio y Acceso a datos. El componente Backend ofrece su funcionalidad a través de una interface de tipo API REST, el componente de Frontend es el encargado de consumir la funcionalidad a medida que los usuarios utilizan la interfaz gráfica de usuario.

**Nodo Frontend:** En este nodo se desplegará el componente de Frontend de la aplicación que consiste en un conjunto de recursos HTML, CSS y JavaScript. Para proveer esta IGU se hace necesario desplegar en este nodo un servidor web como NGinx o Apache que entregue estos archivos a la primera solicitud.

Una vez el sitio o interfaz de usuario es descargada hacia el navegador no se requiere más interacción con este nodo y el usuario interactúa desde su navegador en adelante con el nodo Backend a través del API REST que es consumido por la lógica JavaScript del Frontend.



*Ilustración 10. Vista de despliegue de la arquitectura del sistema  
Fuente: Elaboración propia de la autora*

## 4.2. DISEÑO DE LA PERSISTENCIA

Después de completar el proceso de levantamiento de requerimientos del sistema y la elaboración de los diagramas UML pertinentes para la etapa de análisis y diseño del software, se procedió a desarrollar el modelo de datos con el fin de explicar, de manera descriptiva, la estructura de datos del sistema y la forma en que se relacionan estos últimos con los componentes del sistema.

El diseño de la base de datos se apoya en la información recolectada durante la fase de levantamiento de requerimientos del sistema, así como en conceptos básicos para la construcción de base de datos que abarcan tipos de relaciones, proceso de normalización, conversión a tablas, restricciones, entre otros aspectos.

Para el diseño de la base de datos del sistema SIEGEI se utilizó la herramienta SQL Server, lo cual permitió modelar de manera eficiente la estructura de la base de datos. Además, se elaboró un diagrama entidad-relación (DER) que muestra de manera gráfica las entidades, atributos y relaciones del sistema, brindando una representación visual clara de la estructura de la base de datos y las interacciones entre las diferentes entidades del sistema. Este enfoque integral en el diseño de la base de datos garantiza que la estructura de datos del sistema sea coherente, eficiente y cumpla con los requerimientos funcionales y no funcionales del sistema SIEGEI, contribuyendo así a su correcto funcionamiento y facilitando su mantenimiento y evolución en el tiempo.

Para visualizar el Diagrama Entidad-Relación (DER) del sistema SIEGEI, se presenta la siguiente imagen (*ver Ilustración 11. Diagrama Entidad-Relación del sistema SIEGEI*). Se ha detallado las especificaciones de cada tabla del sistema, abarcando información sobre sus campos, como el tipo de datos, descripción y la capacidad de aceptar valores nulos. Además, se ha definido las llaves primarias y foráneas de cada una de las tablas (*ver tablas 15, 16 y 17*).

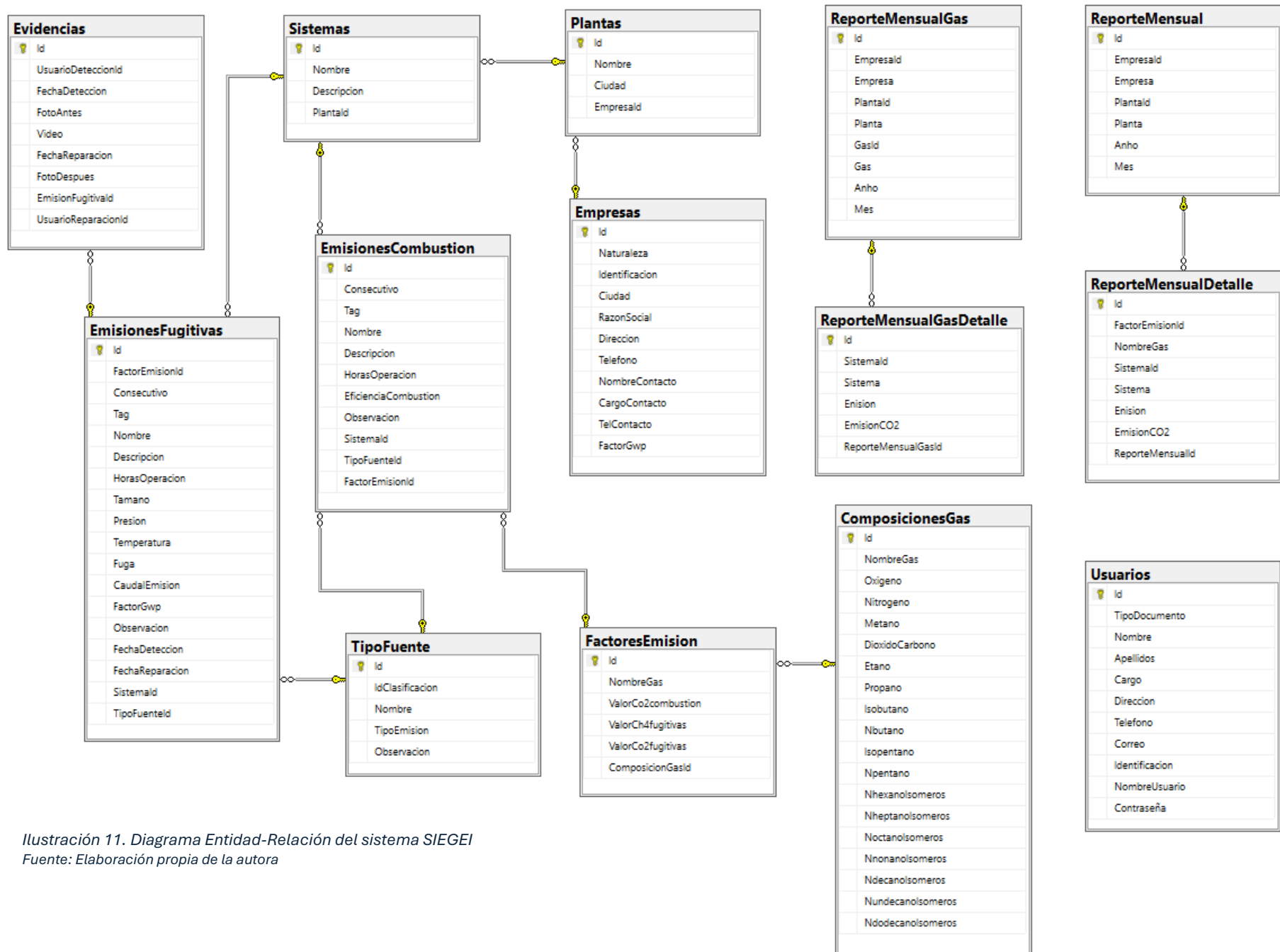


Ilustración 11. Diagrama Entidad-Relación del sistema SIEGEI  
 Fuente: Elaboración propia de la autora

Para brindar mayor claridad, se presenta esta información de manera ilustrativa para las tablas "Empresas", "EmisionCombustion" y "ReporteMensualGasDetalle".

## EMPRESAS

Tabla 15. Diccionario de datos para Tabla Empresas de la base de datos

### TABLA Empresas

<b>Descripción</b>	En esta tabla se registra la descripción de cada campo en la tabla Empresas de la base de datos SIEGEI		
<b>CAMPOS</b>			
<b>Nombre</b>	<b>Tipo de dato</b>	<b>Null</b>	<b>Descripción</b>
<b>Id</b>	int	Not null	Corresponde al identificador de cada empresa
<b>Naturaleza</b>	nvarchar	Not null	Corresponde a la estructura legal de la empresa, ejemplo: empresa individual, sociedad limitada, sociedad anónima, cooperativa, etc.
<b>Identificación</b>	nvarchar	Not null	Corresponde al NIT (Número de Identificación Tributaria)
<b>Ciudad</b>	nvarchar	Not null	Corresponde a la ciudad donde se encuentra la empresa
<b>RazonSocial</b>	nvarchar	Not null	Corresponde al nombre legal de la empresa
<b>Direccion</b>	nvarchar	Not null	Corresponde a la ubicación física donde se encuentran las instalaciones de la empresa
<b>Telefono</b>	nvarchar	Not null	Corresponde al número telefónico utilizado para contactarse con el personal de la empresa
<b>NombreContacto</b>	nvarchar	Not null	Corresponde al nombre de la persona con quien se tiene contacto directo en la empresa cliente
<b>CargoContacto</b>	nvarchar	Not null	Corresponde al cargo de la persona con quien se tiene contacto directo en la empresa cliente
<b>TelContacto</b>	nvarchar	Not null	Corresponde al número telefónico de la persona con quien se tiene contacto en la empresa cliente
<b>FactorGwp</b>	int	Null	Corresponde al potencial de calentamiento atmosférico que maneja esta empresa en específico (GWP - Global Warming Potential)
<b>Llave Primaria</b>	Id		
<b>Llave(s) Foránea(s)</b>	-		
<b>Otras Restricciones</b>	-		

Fuente: Elaboración propia de la autora

# EMISIONES COMBUSTIÓN

Tabla 16. Diccionario de datos para Tabla EmisionesCombustion de la base de datos

**TABLA**  
**EmisionesCombustion**

<b>Descripción</b>	En esta tabla se registra la descripción de cada campo en la tabla EmisionesCombustion de la base de datos SIEGEI		
<b>CAMPOS</b>			
<b>Nombre</b>	<b>Tipo de dato</b>	<b>Null</b>	<b>Descripción</b>
<b>Id</b>	int	Not null	Corresponde al identificador de cada emisión por combustión creada.
<b>Consecutivo</b>	nvarchar	Not null	Corresponde al consecutivo de las emisiones creadas en el histórico de las visitas de campo realizadas desde el principio con esa empresa.
<b>Tag</b>	nvarchar	Not null	Corresponde a la etiqueta única o identificador alfanumérico asignado a un equipo o componente específico dentro de la planta
<b>Nombre</b>	nvarchar	Not null	Corresponde al nombre del componente o equipo que tiene la fuga por combustión
<b>Descripcion</b>	nvarchar	Not null	Corresponde una breve descripción del componente o equipo, donde se puede detallar su función.
<b>HorasOperacion</b>	int	Not null	Corresponde al tiempo total durante el cual ese componente está en funcionamiento (en términos de tiempo al año)
<b>EficienciaCombustion</b>	decimal	Not null	Corresponde a la medida en que un equipo o componente de una planta de tratamiento de gas es capaz de quemar un combustible de manera completa y eficiente, convirtiéndolo en productos de combustión menos contaminantes
<b>Observacion</b>	nvarchar	Not null	Corresponde al donde se puede detallar la ubicación en la planta, y a veces incluye información adicional sobre el componente como el modelo, el tamaño, la capacidad, etc.
<b>Llave Primaria</b>	Id		
<b>Llave(s) Foránea(s)</b>	SistemaId (FK, int, not null) TipoFuenteId (FK, int, not null) FactorEmisionId (FK, int, not null)		

Fuente: Elaboración propia de la autora

# REPORTE MENSUAL GAS DETALLE

Tabla 17. Diccionario de datos para Tabla ReporteMensualGasDetalle de la base de datos

**TABLA**  
**ReporteMensualGasDetalle**

<b>Descripción</b>	En esta tabla se registra la descripción de cada campo en la tabla ReporteMensualGasDetalle de la base de datos SIEGEI, que corresponde al reporte mensual detallado por los sistemas que reporten fugas de un determinado gas.
--------------------	---

**CAMPOS**

Nombre	Tipo de dato	Null	Descripción
<b>Id</b>	int	Not null	Corresponde al identificador de cada reporte generado en el sistema, que informe de las fugas de un determinado gas registradas por sistemas.
<b>Sistemald</b>	int	Not null	Corresponde al id del sistema, asociado al gas consultado, es decir el aplicativo escoge los sistemas que tengan reportadas fugas por ese gas determinado.
<b>Sistema</b>	nvarchar	Not null	Corresponde al nombre del sistema, asociado al gas consultado, es decir el aplicativo escoge los sistemas que tengan reportadas fugas por ese gas determinado.
<b>Emision</b>	decimal	Not null	Corresponde al dato exacto reportado por el usuario cuando se registró la fuga de ese determinado gas, en ese sistema o equipo. Este dato es de las emisiones fugitivas.
<b>EmisionCO2</b>	decimal	Not null	Corresponde al dato exacto reportado por el usuario cuando se registró la fuga de ese determinado gas, en ese sistema o equipo. Este dato es de fuga por combustión.
<b>Llave Primaria</b>	Id		
<b>Llave(s)</b>	ReporteMensualGasId (FK, int, not null)		
<b>Foránea(s)</b>	ReporteMensualGasDetalle_ReporteMensualGas_ReporteMensualGasId		
<b>Otras Restricciones</b>	-		

Fuente: Elaboración propia de la autora

Para acceder a la información completa y detallada de todas las tablas del sistema, se puede consultar el diccionario de datos correspondiente al anexo C. Diccionario de datos. Este documento proporciona una guía exhaustiva que describe cada aspecto de la estructura de la base de datos, facilitando así su comprensión y mantenimiento a lo largo del tiempo.

### 4.3. DISEÑO DEL API

En esta sección se describe la interfaz programática del componente Backend del sistema software. Esta interfaz está implementada como una API REST usando JSON como formato de intercambio de datos y el protocolo HTTP como protocolo base de comunicación.

El API REST diseñado contiene un total de 52 métodos, que usan los tipos de solicitudes HTTP POST, GET y PUT, para la creación, consulta y actualización de recursos respectivamente.

En la siguiente tabla se presentan los métodos del API:

*Tabla 18. Métodos API implementados, con su descripción y su correspondiente método HTTP*

<b>Método</b>	<b>Descripción</b>	<b>Método HTTP</b>
<b>Crear empresa</b>	Crea un nuevo registro de una empresa	POST
<b>Actualizar Empresa</b>	Actualiza el registro de una empresa existente	PUT
<b>Consultar Empresa</b>	Consulta si una empresa existe en el sistema	GET
<b>Listar Empresas</b>	Lista todas las empresas del sistema	GET
<b>Crear Planta</b>	Crea un nuevo registro de una planta	POST
<b>Actualizar Planta</b>	Actualiza el registro de una planta existente	PUT
<b>Consultar Planta</b>	Consulta si una planta existe en el sistema	GET
<b>Listar Plantas</b>	Lista todas las plantas asociadas a una empresa	GET
<b>Crear Sistema</b>	Crea un nuevo registro de una Sistema	POST

<b>Actualizar Sistema</b>	Actualiza el registro de un sistema existente	PUT
<b>Consultar Sistema</b>	Consulta si u sistema existe en el sistema	GET
<b>Listar Sistemas</b>	Lista todos los sistemas asociados a una planta	GET
<b>Crear Emision Fugitiva</b>	Crea un nuevo registro de una emision fugitiva	POST
<b>Actualizar Emision Fugitiva</b>	Actualiza el registro de una emision fugitiva existente	PUT
<b>Consultar Emision Fugitiva</b>	Consulta si una emision fugitiva existe en el sistema	GET
<b>Listar Emisiones Fugitivas</b>	Lista todas las emisiones fugitivas asociados a un sistema	GET
<b>Crear Emision Combustion</b>	Crea un nuevo registro de una emision por combustión	POST
<b>Actualizar Emision Combustion</b>	Actualiza el registro de una emision por combustión existente	PUT
<b>Consultar Emision Combustion</b>	Consulta si una emision por combustión existe en el sistema	GET
<b>Listar Emisiones Combustion</b>	Lista todas las emisiones por combustión asociados a un sistema	GET
<b>Crear Evidencia</b>	Crea un nuevo registro en el sistema de una evidencia asociada a una emisión fugitiva	POST
<b>Actualizar Evidencia</b>	Actualiza un registro de evidencia que exista en el sistema	PUT
<b>Consultar Evidencia</b>	Consulta si una evidencia existe en el sistema	GET
<b>Consultar Evidencias Emision</b>	Consulta todas las evidencias existentes y asociadas a una emision fugitiva	GET
<b>Crear Imagen Deteccion</b>	Crea una imagen de detección asociada a la evidencia	POST
<b>Crear una Imagen Reparacion</b>	Crea una imagen de reparación asociada a la evidencia	POST
<b>Consultar Imagen Deteccion</b>	Consulta una imagen de detección por ID de la evidencia	GET
<b>Consultar Imagen Reparacion</b>	Consulta una imagen de detección por ID de la evidencia	GET

<b>Generar Reporte Mensual</b>	Genera un reporte mensual para una planta de una empresa con los totales de emisión, dado un mes y un año específico	POST
<b>Consultar Reporte Mensual</b>	Consulta un reporte mensual para una planta de una empresa	GET
<b>Listar Reportes Mensuales</b>	Lista un listado de reportes mensuales existentes	GET
<b>Generar Reporte Mensual por Gas</b>	Genera un reporte mensual por Gas para una planta de una empresa con los totales de emisión, dado un mes y un año	POST
<b>Consultar Reporte Mensual por Gas</b>	Consulta un reporte mensual por Gas para una planta de una empresa	GET
<b>Listar Reportes Mensuales por Gas</b>	Lista un listado de reportes mensuales por Gas existentes	GET

Fuente: Elaboración propia de la autora

Con el objetivo de ilustrar el API se presenta el detalle de cuatro métodos en las siguientes tablas; especificando para cada uno su nombre, descripción, URL de acceso, método HTTP, parámetros de entrada y la salida obtenida. El método "Crear Empresa" permite registrar una nueva empresa en el sistema con la información proporcionada en el cuerpo de la solicitud.

Tabla 19. Documentación API - método Crear Empresa

Método		Crear Empresa	
<b>Descripción</b>	Crea un nuevo registro de una empresa		
<b>RUTA</b>	/api/Empresa	<b>TIPO</b>	POST
<b>PARÁMETROS ENTRADA</b>		<b>PARÁMETROS SALIDA</b>	
<pre>{   "naturaleza": "string",   "identificacion": "string",   "ciudad": "string",   "razonSocial": "string",   "direccion": "string",   "telefono": "string",   "nombreContacto": "string",   "cargoContacto": "string",   "telContacto": "string",   "factorGwp": 0 }</pre>		<pre>{   "id": 0,   "naturaleza": "string",   "identificacion": "string",   "ciudad": "string",   "razonSocial": "string",   "direccion": "string",   "telefono": "string",   "nombreContacto": "string",   "cargoContacto": "string",   "telContacto": "string",   "factorGwp": 0 }</pre>	

El método "Consultar Empresa" permite obtener información detallada de una empresa específica a partir de su identificador único. A continuación, se detallan los parámetros de entrada, así como el formato de la respuesta esperada.

Tabla 20. Documentación API - método Consultar Empresa

Método		Consultar Empresa	
<b>Descripción</b>	Consulta si una empresa existe en el sistema		
<b>RUTA</b>	/api/Empresa/{id}}	<b>TIPO</b>	GET
<b>PARÁMETROS ENTRADA</b>		<b>PARÁMETROS SALIDA</b>	
id: Identificador único de la empresa a buscar		<pre>{   "id": 0,   "naturaleza": "string",   "identificacion": "string",   "ciudad": "string",   "razonSocial": "string",   "direccion": "string",   "telefono": "string",   "nombreContacto": "string",   "cargoContacto": "string",   "telContacto": "string",   "factorGwp": 0 }</pre>	

El método "Actualizar Planta" permite modificar la información de una planta existente en el sistema utilizando su identificador único. A continuación, se detallan los parámetros de entrada necesarios y el formato de la respuesta esperada.

Tabla 21. Documentación API – método Actualizar Planta

Método		Actualizar Planta	
<b>Descripción</b>	Actualiza el registro de una planta existente		
<b>RUTA</b>	/api/Planta	<b>TIPO</b>	PUT
<b>PARÁMETROS ENTRADA</b>		<b>PARÁMETROS SALIDA</b>	
<pre>{   "id": 0,   "nombre": "string",   "ciudad": "string",   "empresaId": 0 }</pre>		<pre>{   "id": 0,   "nombre": "string",   "ciudad": "string",   "empresaId": 0 }</pre>	

El método "Listar Plantas" permite obtener una lista de todas las plantas registradas en el sistema. Esta funcionalidad es útil para visualizar todas las plantas disponibles y sus respectivos detalles. Este método no requiere parámetros adicionales en la solicitud. A continuación, se detallan los parámetros de entrada necesarios y el formato de la respuesta esperada.

Tabla 22. Documentación API – método Listar Plantas

Método		Listar Plantas	
<b>Descripción</b>	Lista todas las plantas asociadas a una Empresa		
<b>RUTA</b>	/api/Empresa/GetSet/{Empresald}}	<b>TIPO</b>	GET
<b>PARÁMETROS ENTRADA</b>		<b>PARÁMETROS SALIDA</b>	
EmpresaId: id de la empresa		<pre>[   {     "id": 0,     "nombre": "string",     "ciudad": "string",     "empresaId": 0   } ]</pre>	

Fuente: Elaboración propia de la autora

Finalmente, es importante aclarar que el API se ha especificado utilizando el estándar OpenAPI, que permite definir los métodos implementados por el Backend mediante un archivo en formato YAML.

La especificación en formato OpenAPI proporciona una descripción detallada de cómo interactuar con el API, incluyendo los endpoints disponibles, los parámetros requeridos, los formatos de datos admitidos y los códigos de estado esperados. Esta descripción clara facilita que se comprenda rápidamente cómo utilizar el API en una aplicación, lo que mejora la eficiencia y reduce los errores de integración. Además, al seguir un estándar común, como OpenAPI, se promueve la interoperabilidad entre diferentes sistemas, ya que los desarrolladores pueden confiar en que el API seguirá las mismas convenciones de diseño que otras APIs compatibles con OpenAPI, lo cual aporta un plus en futuras versiones de este

sistema. A continuación, se muestra un fragmento de código que ilustra cómo se especifica el método Crear Empresa.

```
/api/Empresa:
  post:
    tags:
      - Empresa
    requestBody:
      content:
        application/json:
          schema:
            type: object
            properties:
              naturaleza:
                type: string
                nullable: true
              identificacion:
                type: string
                nullable: true
              ciudad:
                type: string
                nullable: true
              razonSocial:
                type: string
                nullable: true
              direccion:
                type: string
                nullable: true
              telefono:
                type: string
                nullable: true
              nombreContacto:
                type: string
                nullable: true
              cargoContacto:
                type: string
                nullable: true
              telContacto:
                type: string
                nullable: true
              factorGwp:
                type: integer
                format: int32
                nullable: true
            additionalProperties: false
```

*Ilustración 12. Código del método Crear Empresa con el estándar OpenAPI  
Fuente: Elaboración propia de la autora – Visual Studio*

La especificación del API se puede consultar en su totalidad en la siguiente URL

## 5. IMPLEMENTACIÓN DEL SISTEMA

### 5.1. TECNOLOGÍAS UTILIZADAS

En esta sección se expondrán las diversas tecnologías que han sido seleccionadas para el desarrollo del presente proyecto.

**Herramienta:** .Net

**Tipo:** Framework

**Versión:** 8.0

**Descripción:** .NET es una plataforma para la ejecución de aplicaciones segura, confiable y de alto rendimiento. Esta plataforma es de código abierto y es mantenida por la Fundación .Net que tiene como principal respaldo a Microsoft. La plataforma es muy versátil ya que puede ser utilizada para construir aplicaciones de escritorio, móviles, web, videojuegos, cloud entre otras y tiene a disposición una gran cantidad de librerías o componentes que se gestiona en un sistema de paquetes llamado NuGet. La plataforma se puede utilizar en diversos sistemas operativos como Windows, Linux o iOS.

**Herramienta:** C#

**Tipo:** Lenguaje de programación

**Versión:** 12

**Descripción:** C# es un lenguaje de programación orientado a objetos y de propósito general utilizado para desarrollar una amplia variedad de aplicaciones como software empresarial, videojuegos y aplicaciones móviles. Fue introducido por la empresa Microsoft en 2002 y es el lenguaje principal de la plataforma .Net. C# es un lenguaje fuertemente tipado, orientada a objetos, con soporte a la programación funcional y sintaxis sencilla y limpia.

**Herramienta:** Visual Studio

**Versión:** 2022

**Tipo:** Entorno de Desarrollo

**Descripción:** Visual Studio es un entorno de desarrollo integrado (IDE) desarrollado por Microsoft que permite crear aplicaciones para distintas plataformas, incluyendo Windows, Android, iOS y Linux. Entre sus principales características se encuentra su soporte al ciclo completo de desarrollo, edición profesional de código, compilación y depuración y su flexibilidad para ser instalado dependiendo de la carga del trabajo a realizar.

**Herramienta:** SQL Server

**Versión:** 2022

**Tipo:** Base de datos

**Descripción:** SQL Server es un sistema de gestión de bases de datos relacional (RDBMS) desarrollado por Microsoft. Se utiliza para almacenar, manipular y recuperar datos en una amplia gama de aplicaciones, desde pequeñas aplicaciones de escritorio hasta sistemas empresariales de gran escala. Tiene una característica de integración de fácil uso con la plataforma .Net.

**Herramienta:** GitHub

**Versión:** 2.44.0

**Tipo:** Plataforma de desarrollo colaborativo

**Descripción:** GitHub es una plataforma de desarrollo colaborativo que utiliza el sistema de control de versiones Git. Permite a los desarrolladores almacenar y gestionar proyectos de software, así como colaborar con otros desarrolladores mediante el control de versiones de los archivos, la gestión de problemas, la revisión de código y la implementación de cambios de forma colaborativa y transparente. GitHub es la plataforma líder en el mundo para el desarrollo de software, la colaboración y la seguridad.

**Herramienta:** Lucidchart

**Versión:** 4.6

**Tipo:** Aplicación de Diseño UML

**Descripción:** Lucidchart es una herramienta en línea para crear diagramas y visualizaciones. Permite a los usuarios diseñar diagramas de flujo, organigramas, diagramas UML, mapas mentales y otros tipos de diagramas de forma colaborativa y en tiempo real. Es ampliamente utilizado en entornos empresariales y educativos debido a su facilidad de uso y a sus capacidades de colaboración, en lo que permite a individuos y equipos visualizar procesos, sistemas y estructuras organizacionales. En resumen, Lucidchart es una herramienta poderosa para visualizar ideas complejas de manera rápida, clara y colaborativa.

**Herramienta:** xUnit.net

**Versión:** 2.7.1

**Tipo:** Pruebas unitarias de .Net Framework

**Descripción:** xUnit.net es una herramienta de pruebas unitarias gratuita y de código abierto diseñada para el .NET Framework. Fue creado por el inventor original de NUnit v2 y se utiliza para probar aplicaciones escritas en C#, F#, VB.NET y otros lenguajes de la plataforma .NET.

## 5.2. ITERACIONES

Como se expuso anteriormente el enfoque metodológico del presente proyecto está basado en el modelo de prototipos evolutivos, por lo tanto, para este marco se hicieron 2 iteraciones, las cuales se describen a continuación.

### 5.2.1. ITERACIÓN 1

En esta interacción se realizaron las siguientes actividades:

- **Levantamiento de requerimientos:** Se llevó a cabo un proceso de recolección y análisis de los requisitos del sistema, en el cual se identificaron las necesidades y funcionalidades clave que el sistema debe cumplir. Se obtuvo un primer prototipo de requerimientos funcionales y no funcionales del sistema, donde se elaboró una tabla con 28 de los primeros y 15 de los segundos.
- **Diseño de la arquitectura:** Se diseñó la estructura general del sistema, definiendo los componentes principales y la forma en que interactúan entre sí para lograr los objetivos del proyecto.
- **Diseño de la persistencia:** Se definió cómo se almacenarán y gestionarán los datos del sistema, incluyendo la selección de la tecnología de base de datos y el diseño de las tablas necesarias.
- **Configuración de tablas base:** Se crearon las tablas iniciales en la base de datos del sistema, estableciendo la estructura básica para almacenar la información de las empresas, plantas y sistemas relacionados.
- **Implementación CRUD de Empresa, Planta, Sistema:** Se desarrollaron las operaciones básicas de Crear, Consultar, Actualizar y Eliminar (CRUD) para las entidades de Empresa, Planta y Sistema, permitiendo la gestión de estos datos de manera eficiente en el sistema.

En la siguiente tabla se muestra los requerimientos implementados en la primera iteración:

Tabla 23. Requerimientos funcionales implementados en la primera iteración

R.F. 04	Crear Empresa
R.F. 05	Consultar Empresa
R.F. 06	Actualizar Empresa
R.F. 07	Eliminar Empresa
R.F. 08	Crear Planta
R.F. 09	Consultar Planta
R.F. 10	Actualizar Planta
R.F. 11	Eliminar Planta
R.F. 12	Crear Sistema
R.F. 13	Consultar Sistema
R.F. 14	Actualizar Sistema
R.F. 15	Eliminar Sistema

Fuente: Elaboración propia de la autora

## 5.2.2. ITERACIÓN 2

En esta interacción se realizaron las siguientes actividades:

- **Refinamiento de requerimientos:** Se realizó una validación los expertos del dominio en CDT a fin de completar y afianzar la información recopilada. De esta actividad surgió una nueva versión del documento de requerimientos.

En este proceso se levantaron 52 requerimientos funcionales y 8 requerimientos no funcionales para el sistema. Sin embargo, para el cumplimiento de los objetivos de este proyecto se escogió un subconjunto de esos requerimientos para implementar su componente backend.

- **Refinamiento de la persistencia:** se complementó el diseño de la base de datos con la actualización del documento de requerimientos, se creó el nuevo esquema en SQL Sever.
- **Implementación funcionalidad de cálculos:** Una vez establecido claramente como se calculan los caudales de emisión se implementó el algoritmo para realizar los cálculos.
- **Implementación CRUD Emisiones Fugitivas y Emisiones por combustión:** Se habilitó la capacidad de gestionar los datos relacionados con las emisiones fugitivas y las emisiones por combustión, permitiendo su creación, consulta, actualización y eliminación a través de su implementación en el componente backend.
- **Implementación CRUD de evidencias:** Se creó la funcionalidad para gestionar las evidencias relacionadas con las emisiones. Esta funcionalidad tenía el reto de enviar archivos multimedia hacia el Backend, donde se referencian para ser utilizados como parte de la visualización.
- **Implementación de la funcionalidad Backend de los reportes:** Se desarrolló la lógica necesaria en el backend del sistema para generar los reportes requeridos por el usuario, asegurando que la información se presente de manera clara.

En la siguiente tabla se muestra los requerimientos implementados en la segunda iteración:

*Tabla 24. Requerimientos funcionales implementados en la segunda iteración*

R.F. 16	Crear Fuente de Emisión por Combustión
R.F. 17	Consultar Fuente de Emisión por Combustión
R.F. 18	Actualizar Fuente de Emisión por Combustión
R.F. 19	Eliminar Fuente de Emisión por Combustión
R.F. 20	Crear Fuente de Emisiones Fugitivas
R.F. 21	Consultar Fuente de Emisiones Fugitivas

R.F. 22	Actualizar Fuente de Emisiones Fugitivas
R.F. 23	Eliminar Fuente de Emisiones Fugitivas
R.F. 28	Crear Evidencia
R.F. 29	Consultar Evidencia
R.F. 30	Actualizar Evidencia
R.F. 31	Eliminar Evidencia
R.F. 32	Crear Archivos Multimedia
R.F. 33	Consultar Archivos Multimedia
R.F. 34	Actualizar Archivos Multimedia
R.F. 35	Eliminar Archivos Multimedia
R.F. 36	Crear Reporte Mensual por Tipo de Gas (detalle sistema)
R.F. 37	Consultar Reporte Mensual por Tipo de Gas (detalle sistema)
R.F. 38	Actualizar Reporte Mensual por Tipo de Gas (detalle sistema)
R.F. 39	Eliminar Reporte Mensual por Tipo de Gas (detalle sistema)
R.F. 40	Crear Reporte Mensual por Planta (detalle gas)
R.F. 41	Consultar Reporte Mensual por Planta (detalle gas)
R.F. 42	Actualizar Reporte Mensual por Planta (detalle gas)
R.F. 43	Eliminar Reporte Mensual por Planta (detalle gas)
R.F. 50	Calcular Emisiones por Combustión
R.F. 51	Calcular Emisiones Fugitivas

Fuente: Elaboración propia de la autora

### 5.3 PROTOTIPO DESARROLLADO

En esta sección, se presenta el modelo utilizado para desarrollar el prototipo del componente Backend del sistema software. El prototipo está desarrollado en lenguaje C# sobre la plataforma .NET. Se distinguen diversos tipos de clases en el modelo y para una mejor comprensión de su estructura se muestra a continuación una ilustración detallada.

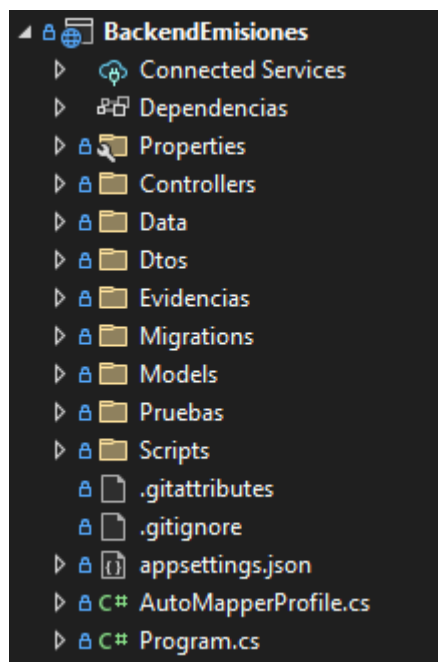


Ilustración 13. Estructura del código del proyecto  
Fuente: Elaboración propia de la autora

#### **Clases Controlador:**

Estas clases son las encargadas de recibir las solicitudes HTTP de parte del componente Frontend, validan los datos recibidos y aplican las reglas del negocio asociadas al método que se va a ejecutar. Estas clases hacen parte del componente Lógica del Negocio de la arquitectura, en total se desarrollaron 7 clases de tipo controlador en el prototipo.

A manera de explicación en la siguiente ilustración se presenta la definición de la clase *EmpresaController*. Se puede observar que la clase es anotada como `[ApiController]` para indicar a la plataforma que se trata de un controlador. Igualmente se indica la ruta que será utilizada para acceder a este controlador con la anotación `[Route]`. Además, la clase define los campos de tipo `DataContext` para acceder al componente de acceso a datos y otro de tipo `IMapper`, que es una utilidad para mapear objetos DTO (Data Transfer Object) a objetos del modelo de persistencia.

```
[ApiController]
[Route("api/[controller]")]
1 referencia
public class EmpresaController : ControllerBase
{
    private readonly DataContext _context;
    private readonly IMapper _mapper;

    0 referencias
    public EmpresaController(DataContext context, IMapper mapper)
    {
        _context = context;
        _mapper = mapper;
    }
}
```

Ilustración 14. Ejemplo Controlador: Clase *EmpresaController*

Fuente: Elaboración propia de la autora

Adicionalmente, se presenta como el controlador de empresa implementa el método para crear una nueva empresa en la siguiente figura. Se indica que el método es de tipo POST con la anotación `[HttpPost]`, y que recibe como parámetro un DTO de tipo `AddEmpresaDto`, que se serializa como un formato JSON en la solicitud. Dado que empresa es el elemento más simple del sistema, la validación

```
[HttpPost]
0 referencias
public ActionResult<Empresa> AddEmpresa(AddEmpresaDto newEmpresa)
{
    var empresa = _mapper.Map<Empresa>(newEmpresa);
    _context.Empresas.Add(empresa);
    _context.SaveChanges();
    return Ok(empresa);
}
```

Ilustración 15. Ejemplo método HTTPPost: método *AddEmpresa*

Fuente: Elaboración propia de la autora

de las reglas de negocio es básica y luego se hace uso del componente de acceso a datos para guardar en la base de datos SQL Server la nueva empresa.

### **Clases DTO (Data Transfer Object):**

Estas clases siguen el patrón de diseño arquitectural DTO y se utilizan como método de transferencia entre Backend y Frontend. Dependiendo del tipo de operación pueden tener más o menos información relacionada a la entidad que representan. Las clases de tipo DTO hacen parte del componente Lógica del Negocio de la arquitectura y son serializados en formato de JSON para ser transportados por HTTP. En total se implementaron 8 clases de tipo DTO.

A manera de ilustración, se presenta en la Figura la clase AddEmpresaDTO con los campos que se utilizan en la creación de una nueva empresa.

```
2 Referencias
public class AddEmpresaDto
{
    0 referencias
    public string Naturaleza { get; set; }
    0 referencias
    public string Identificacion { get; set; }
    0 referencias
    public string Ciudad { get; set; }
    0 referencias
    public string RazonSocial { get; set; }
    0 referencias
    public string Direccion { get; set; }
    0 referencias
    public string Telefono { get; set; }
    0 referencias
    public string NombreContacto { get; set; }
    0 referencias
    public string CargoContacto { get; set; }
    0 referencias
    public string TelContacto { get; set; }
    0 referencias
    public int? FactorGwp { get; set; }
}
```

Ilustración 16. Ejemplo DTO: Clase AddEmpresaDTO  
Fuente: Elaboración propia de la autora

## Clases Modelo:

Este conjunto de clases representa las diferentes entidades que son persistidas en la base de datos y son utilizadas para mapear esas entidades SQL en objetos como lo requiere el framework EntityFramework, utilizado en este prototipo para mapeo objeto-relacional. Estas clases hacen parte del componente de acceso a datos de la arquitectura. Es importante que estas clases se mantengan sincronizadas con cualquier cambio del esquema de base de datos. En total se implementaron 16 clases de tipo Modelo.

A manera de ilustración se presenta en la Figura la clase Empresa con los atributos que representan los campos de la tabla Empresa en la base de datos.

```
public partial class Empresa
{
    4 referencias
    public int Id { get; set; }
    2 referencias
    public string Naturaleza { get; set; }
    2 referencias
    public string Identificacion { get; set; }
    2 referencias
    public string Ciudad { get; set; }
    4 referencias
    public string RazonSocial { get; set; }
    2 referencias
    public string Direccion { get; set; }
    2 referencias
    public string Telefono { get; set; }
    2 referencias
    public string NombreContacto { get; set; }
    2 referencias
    public string CargoContacto { get; set; }
    2 referencias
    public string TelContacto { get; set; }
    2 referencias
    public int? FactorGwp { get; set; }
    2 referencias
    public List<Planta>? Plantas { get; set; }
}
```

Ilustración 17. Ejemplo Modelo Clase Empresa  
Fuente: Elaboración propia de la autora

Esta clase incluye una lista de Plantas, para representar la relación de tipo maestro-detalle donde una empresa posee muchas plantas.

### **Clase Contexto Datos:**

Esta clase es el núcleo de la implementación del componente de acceso a datos, hereda de la clase DbContext provista por EntityFramework. En esta clase se deben declarar todas las entidades que se van a mapear desde la base de datos como objetos.

La figura presenta la definición de la clase DataContext

```
public class DataContext : DbContext
{
    0 referencias
    public DataContext(DbContextOptions<DataContext> options) : base(options)
    {
    }
}

7 referencias
public DbSet<Empresa> Empresas { get; set; }
6 referencias
public DbSet<Planta> Plantas { get; set; }
7 referencias
public DbSet<Sistema> Sistemas { get; set; }
7 referencias
public DbSet<EmisionFugitiva> EmisionesFugitivas { get; set; }
3 referencias
public DbSet<EmisionCombustion> EmisionesCombustion { get; set; }
0 referencias
public DbSet<TipoFuente> TipoFuente { get; set; }
2 referencias
public DbSet<FactorEmision> FactoresEmision { get; set; }
0 referencias
public DbSet<ComposicionGas> ComposicionesGas { get; set; }
4 referencias
```

*Ilustración 18. Ejemplo contexto base de datos: clase DataContext  
Fuente: Elaboración propia de la autora*

Finalmente, el sistema cuando es ejecutado genera una documentación de los métodos desarrollados usando el estándar OPENAPI 3.0 y la herramienta Swagger para generar una interfaz de visualización, navegación e interacción con el Backend. Esta interfaz permite apoyar las tareas futuras de desarrollo del Backend porque soporta la interacción fácil con el Frontend.

En la siguiente figura se presenta la página de interacción generada por .Net en Visual Studio.

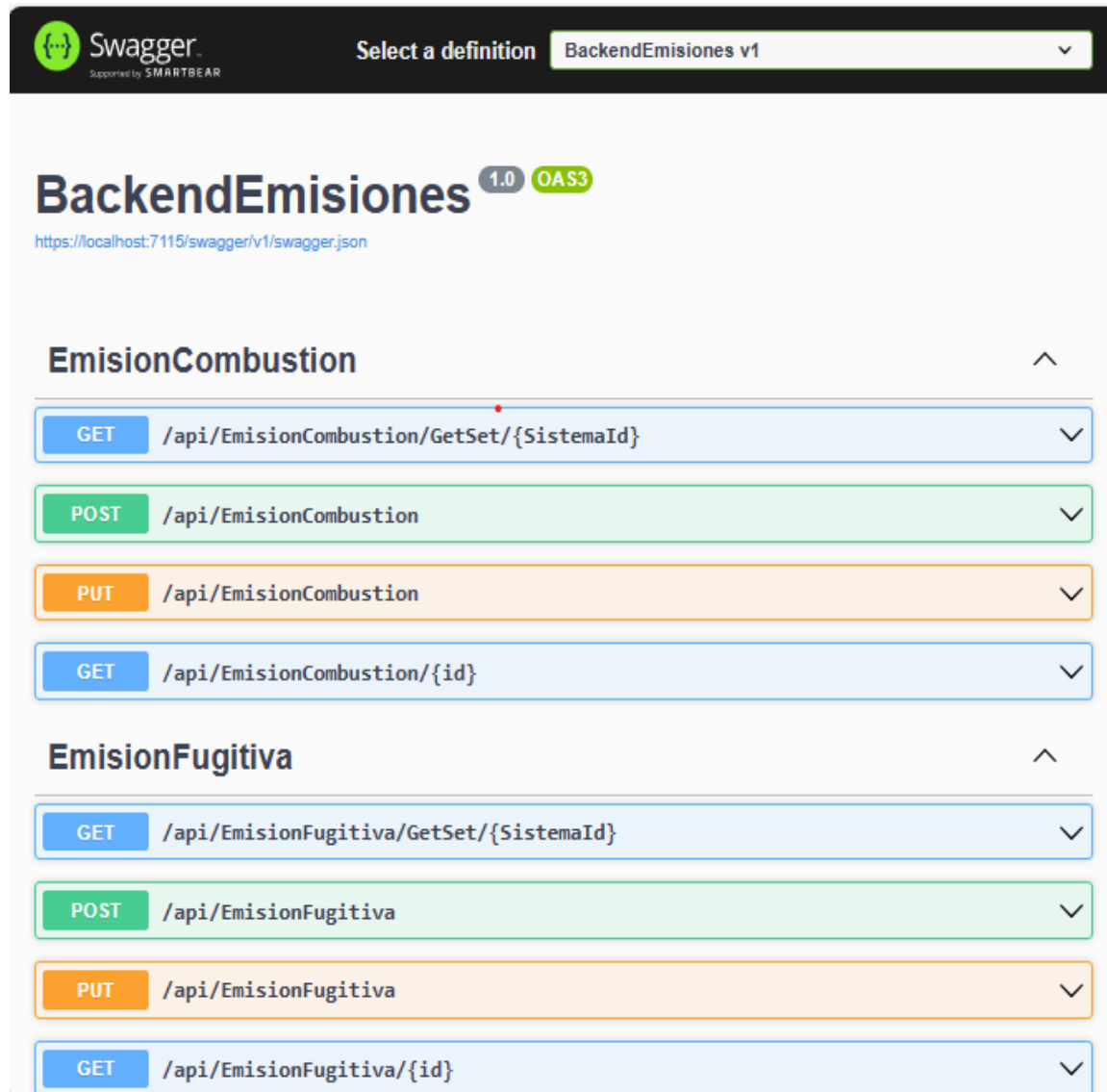


Ilustración 19. Presentación del API en Swagger  
Fuente: Elaboración propia de la autora

## 6. VALIDACIÓN

En el presente capítulo se presentará el mecanismo de validación utilizado para determinar el nivel de cumplimiento de los requerimientos del prototipo funcional del Backend presentado en el capítulo anterior. Para tal propósito, se diseñará un plan de pruebas, el cual abarcará todos los aspectos a evaluar en cada entidad del sistema. Posteriormente, se explicará cómo este plan de pruebas fue automatizado en su totalidad, utilizando herramientas y técnicas específicas para garantizar la correspondiente evaluación. Finalmente, se describirá el proceso de ejecución del plan de pruebas y se presentarán los resultados obtenidos.

### 6.1. PLAN DE PRUEBAS

Con el fin de validar el prototipo funcional desarrollado, se ha diseñado un plan de pruebas que busca abarcar varios escenarios posibles de ejecución por cada requerimiento funcional. Cada prueba definida en el plan incluye una descripción detallada que contempla su identificador, el nombre de la prueba, el objetivo de la prueba, las entradas posibles, las salidas esperadas y el método de verificación de las salidas. Estas pruebas se presentan de manera estructurada mediante tablas, las cuales se han organizado por las pruebas correspondientes a cada entidad del sistema.

#### Entidad 1: EMPRESA

Tabla 25. Plan de pruebas para la entidad Empresa

Identificador	Nombre Prueba	Objetivo	Requerimiento
PF-1	Crear Empresa OK	Probar la creación de una entidad de tipo Empresa, que tiene las condiciones para ser exitosa, en su ejecución se realiza correctamente.	R.F. 04
PF-2	Crear Empresa Incompleta	Probar que la funcionalidad de la creación de una entidad de tipo Empresa valida que todos sus campos obligatorios existen para ser creada.	R.F. 04

<b>PF-3</b>	Actualizar Empresa OK	Probar que la actualización de una entidad de tipo Empresa se realiza de forma exitosa cuando las condiciones están correctas.	R.F. 06
<b>PF-4</b>	Actualizar Empresa No Existente	Probar que la consulta de una entidad de tipo Empresa, no existente en la BD, informa que no existe.	R.F. 06
<b>PF-5</b>	Consultar Empresa OK	Probar que la consulta de una entidad de tipo Empresa se realiza correctamente cuando la entidad existe en la BD.	R.F. 05
<b>PF-6</b>	Consultar Empresa No Existente	Probar que la consulta de una entidad de tipo Empresa no existente, informa que no existe la misma.	R.F. 05
<b>PF-7</b>	Consultar Conjunto de Empresas	Probar que es posible de consultar el conjunto de empresas existente.	R.F. 05

Fuente: Elaboración propia de la autora

## Entidad 2: PLANTA

Tabla 26. Plan de pruebas para la entidad Planta

Identificador	Nombre Prueba	Objetivo	Requerimiento
<b>PF-8</b>	Crear Planta OK	Probar la creación de una entidad de tipo Planta, que tiene las condiciones para ser exitosa, en su ejecución se realiza correctamente.	R.F. 08
<b>PF-9</b>	Crear Planta Empresa No Existente	Probar que la funcionalidad de la creación de una entidad de tipo Planta valida que la empresa asociada exista.	R.F. 08
<b>PF-10</b>	Actualizar Planta OK	Probar que la actualización de una entidad de tipo Planta se realiza de forma exitosa cuando las condiciones están correctas.	R.F. 10
<b>PF-11</b>	Actualizar Planta No Existente	Probar que no es posible actualizar una entidad de tipo Planta cuando el id correspondiente no existe en la BD.	R.F. 10
<b>PF-12</b>	Consultar Planta OK	Probar que la consulta de una entidad de tipo Planta se realiza correctamente cuando la entidad existe en la BD.	R.F. 09
<b>PF-13</b>	Consultar Planta No Existente	Probar que la consulta de una entidad de tipo Planta no existente en la BD, informa que no existe.	R.F. 09
<b>PF-14</b>	Consultar Conjunto de Plantas	Probar que es posible consultar el conjunto de Plantas asociadas a una Empresa en específico.	R.F. 09

Fuente: Elaboración propia de la autora

### Entidad 3: SISTEMA

Tabla 27. Plan de pruebas para la entidad Sistema

Identificador	Nombre Prueba	Objetivo	Requerimiento
PF-15	Crear Sistema OK	Probar la creación de una entidad de tipo Sistema, que tiene las condiciones para ser exitosa en su ejecución, se realiza correctamente.	R.F. 12
PF-16	Crear Sistema Planta No Existente	Probar que la funcionalidad de la creación de una entidad de tipo Sistema valida que la Planta asociada exista.	R.F. 12
PF-17	Actualizar Sistema OK	Probar que la actualización de una entidad de tipo Sistema se realiza de forma exitosa cuando las condiciones están correctas.	R.F. 14
PF-18	Actualizar Sistema No Existente	Probar que no es posible actualizar una entidad de tipo Sistema cuando el id correspondiente no existe en la BD.	R.F. 14
PF-19	Consultar Sistema OK	Probar que la consulta de una entidad de tipo Sistema se realiza correctamente cuando la entidad existe en la BD.	R.F. 13
PF-20	Consultar Sistema No Existente	Probar que la consulta de una entidad de tipo Sistema, no existente en la BD, informa que no existe.	R.F. 13
PF-21	Consultar Conjunto de Sistemas	Probar que es posible consultar el conjunto de Sistemas asociadas a una determinada Planta.	R.F. 13

Fuente: Elaboración propia de la autora

### Entidad 4: EMISION COMBUSTION

Tabla 28. Plan de pruebas para la entidad Emision Combustion

Identificador	Nombre Prueba	Objetivo	Requerimiento
PF-22	Crear Emision Combustion OK	Probar la creación de una entidad de tipo EmisionCombustion, que tiene las condiciones para ser exitosa en su ejecución, se realiza correctamente.	R.F. 16
PF-23	Crear Emision Combustion Sistema No Existente	Probar que la funcionalidad de la creación de una entidad de tipo EmisionCombustion valida que el Sistema asociado exista.	R.F. 16

<b>PF-24</b>	Actualizar Emision Combustion OK	Probar que la actualización de una entidad de tipo EmisionCombustion se realiza de forma exitosa cuando las condiciones están correctas.	R.F. 18
<b>PF-25</b>	Actualizar Emision Combustion No Existente	Probar que no es posible actualizar una entidad de tipo EmisionCombustion cuando el id correspondiente no existe en la BD.	R.F. 18
<b>PF-26</b>	Consultar Emision Combustion OK	Probar que la consulta de una entidad de tipo EmisionCombustion se realiza correctamente cuando la entidad existe en la BD.	R.F. 17
<b>PF-27</b>	Consultar Emision Combustion No Existente	Probar que la consulta de una entidad de tipo EmisionCombustion no existente en la BD, informa que no existe.	R.F. 17
<b>PF-28</b>	Consultar Conjunto de Emisiones Combustion	Probar que es posible consultar el conjunto de entidades EmisionCombustion asociadas a una Planta.	R.F. 17

Fuente: Elaboración propia de la autora

## Entidad 5: EMISION FUGITIVA

Tabla 29. Plan de pruebas para la entidad Emision Fugitiva

<b>Identificador</b>	<b>Nombre Prueba</b>	<b>Objetivo</b>	<b>Requerimiento</b>
<b>PF-29</b>	Crear Emision Fugitiva OK	Probar la creación de una entidad de tipo EmisionFugitiva, que tiene las condiciones para ser exitosa en su ejecución, se realiza correctamente.	R.F. 20
<b>PF-30</b>	Crear Emision Fugitiva Sistema No Existente	Probar que la funcionalidad de la creación de una entidad de tipo EmisionFugitiva valida que el Sistema asociado exista.	R.F. 20
<b>PF-31</b>	Actualizar Emision Fugitiva OK	Probar que la actualización de una entidad de tipo EmisionFugitiva se realiza de forma exitosa cuando las condiciones están correctas.	R.F. 22
<b>PF-32</b>	Actualizar Emision Fugitiva No Existente	Probar que no es posible actualizar una entidad de tipo EmisionFugitiva cuando el id correspondiente no existe en la BD.	R.F. 22
<b>PF-33</b>	Consultar Emision Fugitiva OK	Probar que la consulta de una entidad de tipo EmisionFugitiva se realiza correctamente cuando la entidad existe en la BD.	R.F. 21

<b>PF-34</b>	Consultar Emision Fugitiva No Existente	Probar que la consulta de una entidad de tipo EmisionFugitiva no existente en la BD, informa que no existe.	R.F. 21
<b>PF-35</b>	Consultar Conjunto de Emisiones Fugitivas	Probar que es posible consultar el conjunto de entidades EmisionFugitiva asociadas a una Planta.	R.F. 21

Fuente: Elaboración propia de la autora

## Entidad 6: EVIDENCIA

Tabla 30. Plan de pruebas para la entidad Evidencia

Identificador	Nombre Prueba	Objetivo	Requerimiento
<b>PF-36</b>	Crear Evidencia OK	Probar la creación de una entidad de tipo Evidencia, que tiene las condiciones para ser exitosa en su ejecución, se realiza correctamente.	R.F. 28
<b>PF-37</b>	Crear Evidencia Emision No Existente	Probar que la funcionalidad de la creación de una entidad de tipo Evidencia valida que el Sistema asociado exista.	R.F. 28
<b>PF-38</b>	Actualizar Evidencia OK	Probar que la actualización de una entidad de tipo Evidencia se realiza de forma exitosa cuando las condiciones están correctas.	R.F. 30
<b>PF-39</b>	Actualizar Evidencia No Existente	Probar que no es posible actualizar una entidad de tipo Evidencia cuando el id correspondiente no existe en la BD.	R.F. 30
<b>PF-40</b>	Consultar Evidencia OK	Probar que la consulta de una entidad de tipo Evidencia se realiza correctamente cuando la entidad existe en la BD.	R.F. 29
<b>PF-41</b>	Consultar Evidencia No Existente	Probar que la consulta de una entidad de tipo Evidencia no existente en la BD informa que no existe.	R.F. 29
<b>PF-42</b>	Consultar Conjunto de Evidencias	Probar que es posible consultar el conjunto de entidades Evidencia asociadas a una Emision Fugitiva.	R.F. 29
<b>PF-43</b>	Consultar Archivo Deteccion	Probar que es posible obtener un archivo multimedia asociado a la detección de una EmisionFugitiva.	R.F. 33

Fuente: Elaboración propia de la autora

## Entidad 7: REPORTE MENSUAL

Tabla 31. Plan de pruebas para la entidad Reporte Mensual

Identificador	Nombre Prueba	Objetivo	Requerimiento
PF-44	Generar Reporte Mensual OK	Probar que el reporte Mensual de emisiones fugitivas se genera correctamente dada una planta, un año y un mes.	R.F. 40
PF-45	Generar Reporte Mensual Gas OK	Probar que el reporte Mensual por tipo de Gas de emisiones fugitivas se genera correctamente dada una planta, un año y un mes.	R.F. 36
PF-46	Generar Reportar Mensual Planta No Existente	Probar que el reporte Mensual de emisiones fugitivas no se genera cuando la planta indicada no existe.	R.F. 40
PF-47	Generar Reportar Mensual Gas Planta No Existente	Probar que el reporte Mensual por tipo de Gas de emisiones fugitivas no se genera cuando la planta indicada no existe.	R.F. 36
PF-48	Consultar Reporte Mensual No Existente	Probar que la consulta de un reporte mensual, que no existe en la BD, regresa el mensaje de consulta no existente.	R.F. 41
PF-49	Consultar Reporte Mensual Gas No Existente	Probar que la consulta de un reporte mensual, por tipo de gas, que no existe en la BD, regresa el mensaje de consulta no existente.	R.F. 37
PF-50	Consultar Conjunto Reporte Mensual	Probar que se consulta adecuadamente el conjunto de reportes de emisiones fugitivas mensuales que existen.	R.F. 41
PF-51	Consultar Conjunto Reporte Mensual Gas	Probar que se consulta adecuadamente el conjunto de reportes de emisiones fugitivas mensuales por tipo de gas que existen.	R.F. 37

Fuente: Elaboración propia de la autora

## 6.2. AUTOMATIZACIÓN DEL PLAN DE PRUEBAS

Con el objetivo de mejorar el proceso de desarrollo y permitir que el conjunto de pruebas definido en el plan de pruebas pueda ser ejecutado cada vez que se actualice el código del Backend, las pruebas se automatizaron en su totalidad. Para tal propósito, se utilizó el framework XUnit para la definición y ejecución de pruebas dentro del ambiente .Net. Además, se empleó la característica MVC.Testing de .Net, que permite realizar pruebas de integración End-To-End, ejecutando un cliente HTTP que consume el API tal como lo haría el Frontend o una herramienta externa como Postman.

Las pruebas se desarrollaron en el lenguaje C# y están diseñadas para ser incluidas en un pipeline de CI/CD en futuras versiones, permitiendo así la integración y el despliegue continuos de la aplicación. Esta automatización no solo agiliza la validación del código, sino que también asegura una cobertura completa y repetible de las pruebas. A largo plazo, en una siguiente versión de este trabajo de grado, este enfoque permitirá un desarrollo más robusto y confiable, alineado con metodologías orientadas por pruebas como TDD (Desarrollo Guiado por Pruebas).

Durante el proceso de automatización, se enfrentaron varios desafíos, como la configuración adecuada del entorno de pruebas y la gestión de datos de prueba consistentes. Estos desafíos se abordaron mediante la implementación de entornos de prueba aislados y la utilización de datos de prueba estandarizados. La automatización del plan de pruebas ha resultado en una mejora significativa en la eficiencia del proceso de desarrollo y en la calidad del prototipo. Este enfoque permitió centrarse en la implementación de las funcionalidades con la confianza de que cualquier regresión sería detectada y abordada de manera oportuna.

El proyecto **TestBackendEmisiones** de tipo pruebas fue creado en la herramienta Visual Studio para automatizar las pruebas. La estructura del proyecto es presentada en la ilustración 20.

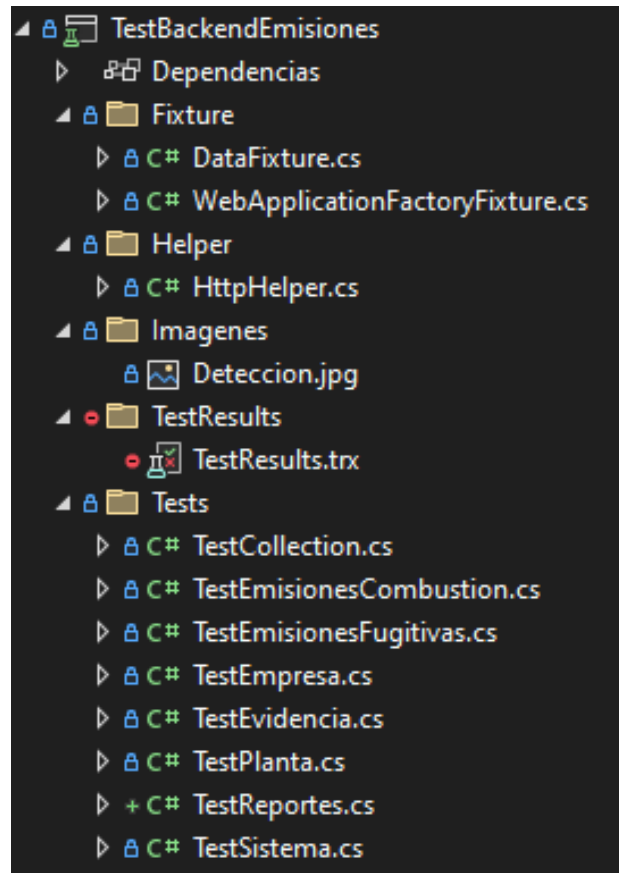


Ilustración 20. Visualización de la estructura del proyecto –jerarquía de ramas en Visual Studio  
Fuente: Elaboración propia de la autora

Es importante señalar que el folder **Tests** contiene el conjunto de clases que ejecutan las pruebas. En este caso hay 7 Clases de pruebas: *TestEmpresa*, *TestPlanta*, *TestSistema*, *TestEmisionCombustion*, *TestEmisionFugitiva*, *TestEvidencia* y *TestReportes*. Estas clases se corresponden con la distribución de las pruebas tal cual como se definieron en el plan de pruebas.

Las clases que se encuentran en el folder **Fixture** ofrecen funcionalidad de contexto para inicializar la base de datos del ambiente de pruebas. En este caso, la base de datos se inicializa la base de datos con más de 500 registros de diverso tipo para asegurar unas pruebas en un ambiente adecuado de datos. Por otra parte, el contexto proporcionado por estas clases permite acceder a estos datos

para compararlos con los resultados obtenidos mediante la ejecución de las pruebas (consumo del API), asegurando así la coherencia de los datos.

```
[Collection("Backend EmisionCDT Collection")]
1 referencia
public class TestPlanta
{
    private readonly WebApplicationFactoryFixture _fixture;
    0 referencias
    public TestPlanta(WebApplicationFactoryFixture fixture) {...}

    [Fact(DisplayName = "Crear Planta OK")]
    0 referencias
    public async Task CrearPlantaOK() {...}

    [Fact(DisplayName = "Crear Planta Empresa No Existente")]
    0 referencias
    public async Task CrearPlantaEmpresaNoExistente() {...}

    [Fact(DisplayName = "Actualizar Planta OK")]
    0 referencias
    public async Task ActualizarPlantaOK() {...}

    [Fact(DisplayName = "Actualizar Planta No Existente")]
    0 referencias
    public async Task ActualizarPlantaNoExistente() {...}

    [Fact(DisplayName = "Consultar Planta OK")]
    0 referencias
    public async Task ConsultarPlantaOK() {...}

    [Fact(DisplayName = "Consultar Planta No Existente")]
    0 referencias
    public async Task ConsultarPlantaNoExistente() {...}

    [Fact(DisplayName = "Consultar Conjunto de Plantas")]
    0 referencias
    public async Task ConsultaConjuntoPlantas() {...}
}
```

Ilustración 21. Visualización de la clase TestPlanta con las pruebas asociadas a la entidad Planta  
Fuente: Elaboración propia de la autora

Con el fin de ilustrar mejor la definición de las pruebas se presenta en la ilustración 21, la clase **TestPlanta**, que define todas las pruebas asociadas a las operaciones definidas en el API que tienen relación con la entidad **Planta** (ver sección anterior). La clase **TestPlanta** pertenece a la colección de pruebas denominada "*Backend EmisionesCDT Collection*" y define siete métodos de pruebas, cada uno marcado con la anotación **Fact** y con el nombre que se corresponde del plan de pruebas.

Entrando en el detalle de una prueba específica, en la Ilustración 22 se presenta la prueba "**Crear Planta OK**" (una de las siete pruebas definidas en **TestPlanta**). Esta prueba corresponde a un método decorado con la anotación **Fact** (de Xunit), que indica que se trata de una prueba y tiene como nombre: "**Crear Planta OK**", que se utiliza para su identificación.

```
[Fact(DisplayName = "Crear Planta OK")]
0 referencias
public async Task CrearPlantaOK()
{
    // Arrange
    var client = _fixture.Client;
    var planta = CrearPlantaDTO();
    var content = HttpHelper.GetJsonHttpContent(planta);

    // Act
    var response = await client.PostAsync("api/Planta", content);
    var plantaStr = response.Content.ReadAsStringAsync();
    var plantaRta = JsonConvert.DeserializeObject<Planta>(plantaStr.Result);

    // Assert
    // Evalua que el código HTTP de respuesta es 200 (OK)
    Assert.Equal(HttpStatusCode.OK, response.StatusCode);
    var plantaDB = _fixture.GetPlantaPorId(plantaRta.Id);

    // Valida que los atributos son iguales a la BD
    Assert.Equal(plantaDB.Nombre, plantaRta.Nombre);
    Assert.Equal(plantaDB.Ciudad, plantaRta.Ciudad);
    Assert.Equal(plantaDB.EmpresaId, plantaRta.EmpresaId);
}
```

Ilustración 22. Código de la prueba "Crear Planta Ok"  
Fuente: Elaboración propia de la autora

La prueba contenida en el método se divide en tres secciones: preparar (**Arrange**), ejecutar (**Act**) y validar (**Assert**), las cuales se identifican mediante comentarios en el método.

En la sección de **preparar (Arrange)**, se crea el cliente HTTP y se instancia un objeto de tipo **Planta** que corresponde al DTO. Luego, este DTO se serializa en formato JSON para ser incluido en la solicitud HTTP.

En la sección de **ejecutar (Act)**, se realiza la solicitud HTTP al servidor a la URL “**api/Planta**” utilizando el método POST (*ver API*), enviando el contenido definido previamente. Una vez ejecutada la solicitud, se recupera la respuesta y se realizan las transformaciones necesarias para convertirla del JSON devuelto al tipo **Planta**, facilitando así la fase de validación.

Finalmente, en la sección de **validación (Assert)**, se consulta la base de datos con el ID devuelto por la solicitud para obtener un objeto correspondiente a la BD (**plantaDB**) y se comparan los atributos de la creación (**plantaRta**) con los que existen en la base de datos. Adicionalmente, se valida que la respuesta HTTP tenga código 200.

De forma similar, se implementaron 51 pruebas dentro de las siete clases pertenecientes al proyecto de automatización de pruebas.

## 6.3. EJECUCIÓN Y RESULTADOS

### 6.3.1. EJECUCIÓN

#### **Definición del ambiente de pruebas**

Para la ejecución de las pruebas, se realizó el despliegue de la aplicación en un ambiente de pruebas en un equipo de cómputo con las siguientes especificaciones:

- Procesador: 4 Cores
- Memoria: 4 GB
- Sistema Operativo: Windows 10 Pro

Además, se instaló el *runtime* de .Net versión 8.0 y el sistema de base datos SQL Server 2022 y se garantizó una interfaz de red con conexión a internet y se inicializó la base de datos a través de la herramienta de pruebas utilizada, para asegurar el funcionamiento adecuado del sistema.

#### **Ejecución**

La ejecución de las pruebas se llevó a cabo principalmente en el entorno de Visual Studio, como se muestra en la ilustración 23, ya que este ofrece la capacidad de ejecutar subconjuntos de pruebas y depurarlas, lo que proporciona un control más fino de la ejecución. También se ejecutaron pruebas a través de la CLI de .Net.

Es importante destacar que las pruebas se ejecutaron a medida que se avanzaba con el desarrollo del proyecto, y no solo al final del mismo. Esto permitió detectar problemas con la lógica del negocio implementada en el Backend de manera temprana y corregirlos de manera oportuna.

Durante la ejecución de las pruebas automatizadas, se identificaron casos en los que las pruebas no pasaron exitosamente. En estos casos, se siguió un proceso de gestión de errores para identificar la causa raíz y tomar las acciones correctivas necesarias.

Una vez que se corregía el error, se volvía a ejecutar la prueba para asegurar que el problema se hubiera solucionado correctamente. En caso de persistir el error, se realizaba un análisis más profundo para determinar la causa subyacente.

Gracias a este proceso de gestión de errores, se logró un nivel óptimo de calidad en el software, garantizando que todas las funcionalidades se comportaran como se esperaba incluso después de realizar cambios en el código.

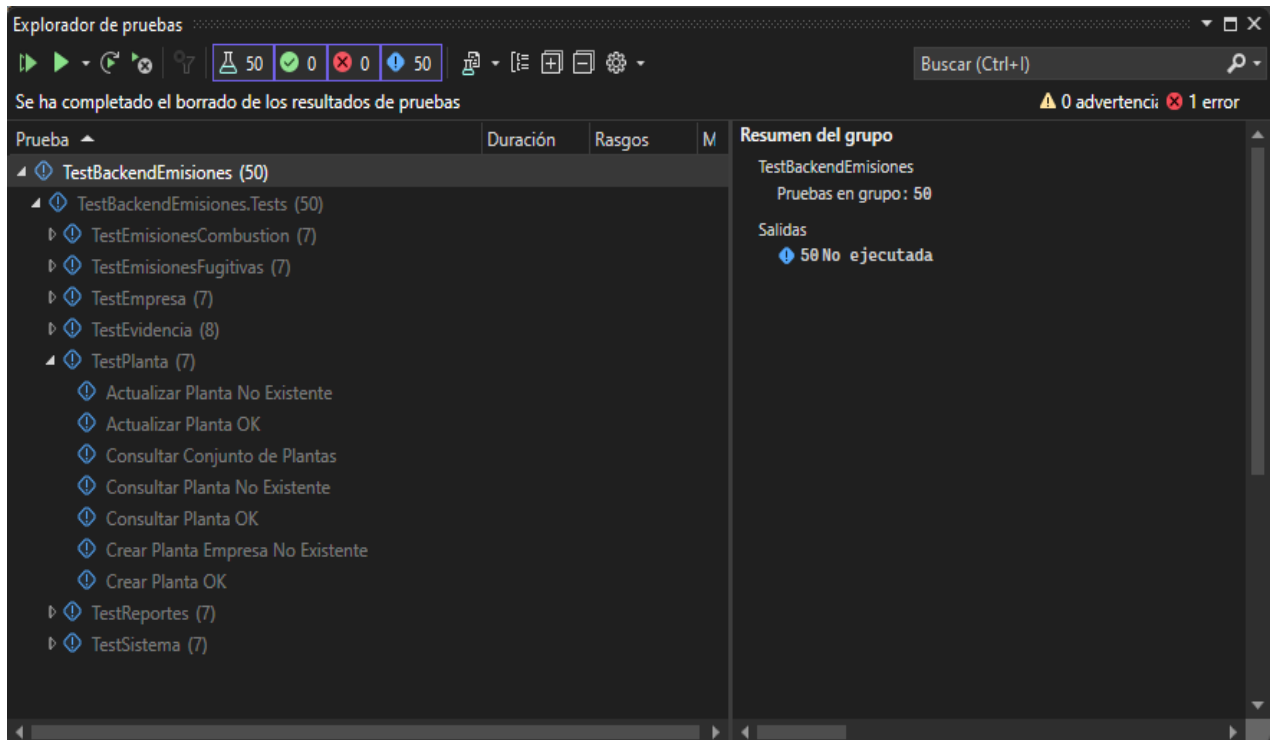
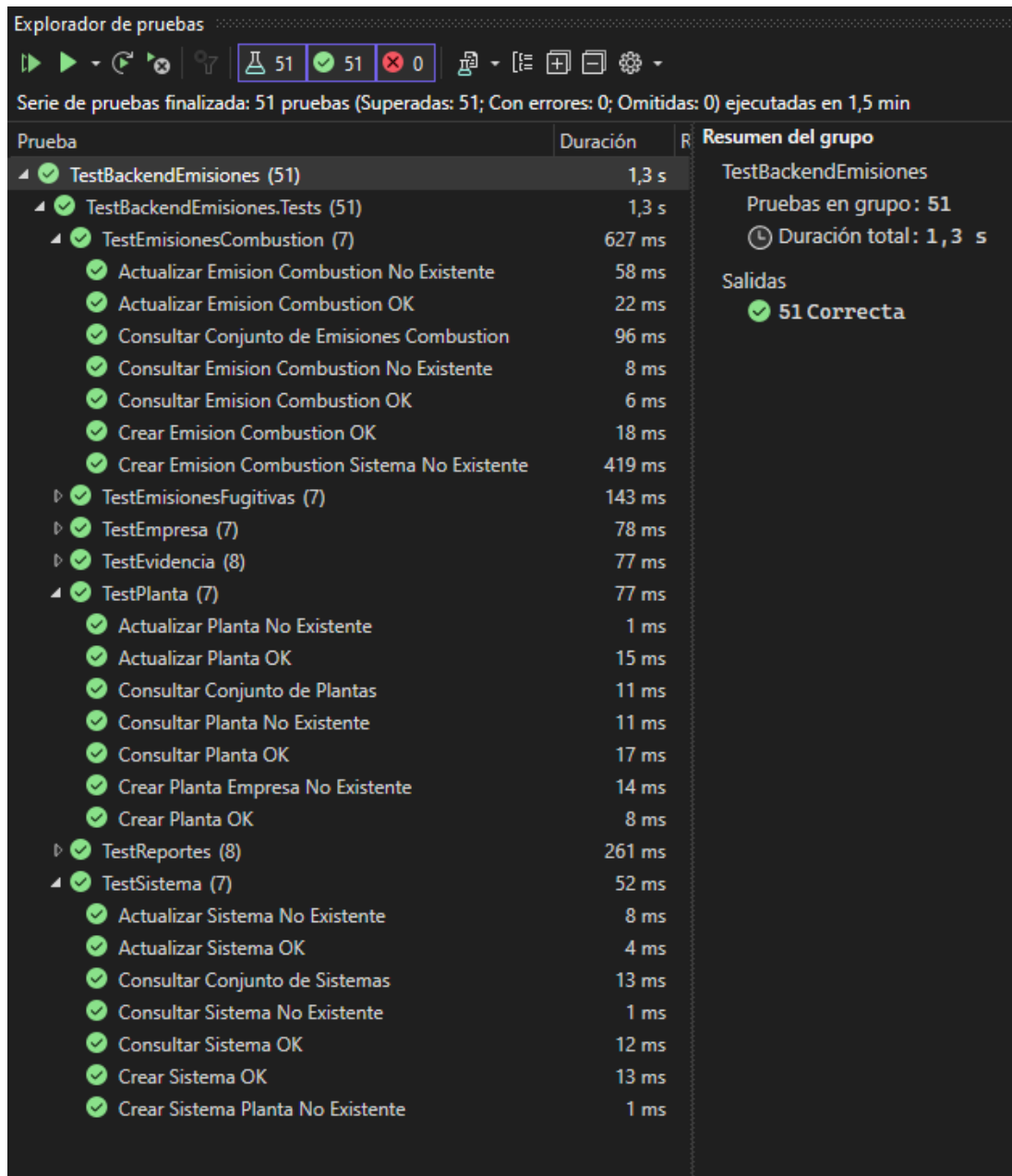


Ilustración 23. Visualización de explorador de pruebas con las pruebas a ejecutar según el plan de pruebas diseñado  
Fuente: Elaboración propia de la autora

### 6.3.2. RESULTADOS

El plan de pruebas fue superado en un 100% por el prototipo de Backend implementado. Sin embargo, es importante mencionar que las pruebas no fueron superadas en su totalidad al primer intento, sino que a medida que se avanzaba en el desarrollo se iban corriendo las pruebas automatizadas y esto permitía detectar problemas con la lógica del negocio implementada en el Backend.



Explorador de pruebas

Serie de pruebas finalizada: 51 pruebas (Superadas: 51; Con errores: 0; Omitidas: 0) ejecutadas en 1,5 min

Prueba	Duración	R	Resumen del grupo
TestBackendEmisiones (51)	1,3 s		TestBackendEmisiones
TestBackendEmisiones.Tests (51)	1,3 s		Pruebas en grupo: 51
TestEmisionesCombustion (7)	627 ms		Duración total: 1,3 s
Actualizar Emision Combustion No Existente	58 ms		Salidas
Actualizar Emision Combustion OK	22 ms		51 Correcta
Consultar Conjunto de Emisiones Combustion	96 ms		
Consultar Emision Combustion No Existente	8 ms		
Consultar Emision Combustion OK	6 ms		
Crear Emision Combustion OK	18 ms		
Crear Emision Combustion Sistema No Existente	419 ms		
TestEmisionesFugitivas (7)	143 ms		
TestEmpresa (7)	78 ms		
TestEvidencia (8)	77 ms		
TestPlanta (7)	77 ms		
Actualizar Planta No Existente	1 ms		
Actualizar Planta OK	15 ms		
Consultar Conjunto de Plantas	11 ms		
Consultar Planta No Existente	11 ms		
Consultar Planta OK	17 ms		
Crear Planta Empresa No Existente	14 ms		
Crear Planta OK	8 ms		
TestReportes (8)	261 ms		
TestSistema (7)	52 ms		
Actualizar Sistema No Existente	8 ms		
Actualizar Sistema OK	4 ms		
Consultar Conjunto de Sistemas	13 ms		
Consultar Sistema No Existente	1 ms		
Consultar Sistema OK	12 ms		
Crear Sistema OK	13 ms		
Crear Sistema Planta No Existente	1 ms		

Ilustración 24. Visualización del explorador de pruebas del sistema  
Fuente: Elaboración propia de la autora

Una vez se superaron todas las pruebas definidas, como lo muestra el explorador de pruebas terminada la ejecución, se consideró que el desarrollo cumplió con el nivel de calidad exigido. En la anterior imagen, la ilustración 24, se muestra que fueron superadas 51 pruebas ejecutadas de las 51 pruebas definidas.

## 7. CONCLUSIONES

- Se realizó una actividad de levantamiento de requerimientos que involucró la interacción con expertos del dominio con el fin de identificar y especificar cuáles serán las principales funcionalidades del sistema software de inventarios de emisiones fugitivas. Igualmente, se identificaron los requerimientos no funcionales que son las características de calidad asociadas al sistema. Esta actividad dio como resultado el documento de especificación de requerimientos que se incluye como anexo.
- Basado en el conjunto de requerimientos del sistema y habiendo escogido las tecnologías a utilizar, se realizó el diseño arquitectural y de detalle del sistema software a implementar. La arquitectura está basada en un estilo de capas y para el diseño de detalle se aplicaron principios y patrones de diseño orientado a objetos. En la actividad también se diseñó el modelo de persistencia relacional y la interfaz programática (API) que se debe utilizar para interactuar con el componente Backend del sistema software.
- Como medio de validación del diseño del sistema que es el objetivo central de este proyecto se implementó un prototipo funcional del componente Backend que incluye un subconjunto de los requerimientos identificados. El desarrollo se hizo utilizando un enfoque de prototipos evolutivos con 2 ciclos. Este prototipo se implementó sobre la plataforma .Net con persistencia en base de datos SQL Server y comunicación mediante un API REST, así mismo el prototipo fue desplegado en un servidor para ser probado.
- Se realizó un plan de pruebas funcionales para validar el correcto funcionamiento del prototipo funcional a través del API REST definido lo que permitirá a futuro integrar la capa de interacción con usuarios Frontend. Además, las pruebas fueron automatizadas en su totalidad posibilitando su integración en un marco de CI/CD.

- Los diversos artefactos logrados (requerimientos, diseño, prototipo y pruebas) constituyen una base fundamental para CDT de Gas en su objetivo de evolucionar el presente proyecto hacia un sistema software completo y que se incorpore en su pila tecnológica como elemento diferenciador de este centro de tecnología en el área del Gas.
- Elegir la modalidad de trabajo de grado mediante práctica empresarial es de gran valor para el estudiante tanto en lo personal como para su formación profesional. Esta modalidad no solo permite medir la capacidad y habilidad para plantear y resolver problemas, trabajar en equipo y adquirir experiencia en el campo laboral, sino que también brinda la oportunidad de interactuar en un ambiente profesional y realizar una experiencia que involucra diversas fases en el desarrollo de un producto software.

## 8. RECOMENDACIONES

- Se recomienda implementar la interfaz de usuario (componente Frontend) en una tecnología que permita su fácil evolución como Angular o React. Es importante que este Frontend se haga del API REST definido en el diseño e implementado en el prototipo del Backend para obtener y modificar los datos de la aplicación.
- Se recomienda utilizar un modelo de seguridad para el consumo del API ya que la implementación actual se enfocó en la parte funcional y no implementó seguridad de acceso, una opción adecuada es JWT (*JSON Web Token*)
- Se recomienda escoger un modelo de despliegue basado en contenedores que permita su fácil implantación en ambientes Cloud.
- Se recomienda implementar un módulo de exportación de datos a fin de que sean consumidos por herramientas de visualización y análisis como PowerBI.
- Se recomienda implementar en un trabajo futuro del sistema, gestionando los componentes principales del sistema en forma de módulos así: un módulo de registro de emisiones, un módulo de seguimiento de emisiones y un módulo de generación de informes. Con el objetivo que estos módulos trabajen en conjunto para proporcionar a los usuarios una herramienta integral y eficiente para la gestión de inventarios de emisiones fugitivas de gases hidrocarburos en el sector Oil&Gas.
- Se sugiere para futuras iteraciones o trabajos relacionados con este sistema, se considere la creación e implementación de un Manual del Usuario, que podría incluir información sobre la navegación en la interfaz. Este manual proporcionaría a los usuarios finales una guía detallada sobre cómo utilizar el sistema de manera eficaz, lo que mejoraría la experiencia del usuario.

## REFERENCIAS BIBLIOGRÁFICAS

[ 1 ] AMELL ARRIETA, Andrés, et al. Consultoría técnica para el fortalecimiento y mejora de la base de datos de factores de emisión de los combustibles colombianos- FECOC [en línea]. Medellín: Informe Final Proyecto Incombustion-Natura, 2016. 52 páginas. Investigación e innovación en combustión avanzada de uso industrial. Disponible para consulta:

<[http://www.upme.gov.co/calculadora\\_emisiones/aplicacion/Informe\\_Final\\_FECOC\\_Correcciones\\_UPME\\_FunNatura.pdf](http://www.upme.gov.co/calculadora_emisiones/aplicacion/Informe_Final_FECOC_Correcciones_UPME_FunNatura.pdf)>.

[ 2 ] BRADBURY, James, et al. CLEARING THE AIR: REDUCING UPSTREAM GREENHOUSE GAS EMISSIONS FROM U.S. NATURAL GAS SYSTEMS. En: World Resources Institute [en línea]. Abril, 2013. p. 41. 60 páginas. [Consultado en Junio de 2022]. Disponible para consulta: <[http://pdf.wri.org/clearing\\_the\\_air\\_full.pdf](http://pdf.wri.org/clearing_the_air_full.pdf)>.

[ 3 ] BUITRAGO BENAVIDES, Dalia Mercedes, et al. IDEAM TOMO II Cap 2. Inventario Nacional De Emisiones De Gases De Efectos Invernaderos. Cambio climático [página web]. (2007). 34 páginas. [Consultado en febrero de 2023]. Disponible para consulta: <<http://www.cambioclimatico.gov.co/documents/40860/219937/2<sup>a</sup>+Comunicación+Capítulo+2.pdf/81b221d1-1823-4a0b-adbd-d8b85f1b846c>>.

[ 4 ] CENTRO DE documentación de Entity Framework [Anónimo]. Microsoft Learn: Build skills that open doors in your career [página web]. [Consultado en Mayo de 2024]. Disponible para consulta: <<https://learn.microsoft.com/es-es/ef/>>.

[ 5 ] CERVANTES, Humberto y KAZMAN, Rick. Designing software architectures: a practical approach. Boston: Addison-Wesley, 2016. 372 páginas. ISBN 0134390784.

[ 6 ] DÁVILA, Manuel. Introducción a xUnit: Una poderosa herramienta para Unit Testing en .NET. DEV Community [página web]. (12, junio, 2023). [Consultado en Mayo de 2024]. Disponible para consulta: <<https://dev.to/maadcode/introduccion-a-xunit-una-poderosa-herramienta-para-unit-testing-en-net-34b0>>.

[ 7 ] DE 0 A DEV, Daniel. Pruebas Unitarias en .Net Core con xUnit – 1 – Lo Básico – d0a.dev. d0a.dev – Desarrolla tu Carrera [página web]. [Consultado en Junio de 2024]. Disponible para consulta: <<https://d0a.dev/2021/05/14/pruebas-unitarias-en-net-core-con-xunit/>>.

UPME [página web]. [Consultado el 7, junio, 2024]. Disponible en Internet: <[http://www.upme.gov.co/calculadora\\_emisiones/aplicacion/calculadora.html](http://www.upme.gov.co/calculadora_emisiones/aplicacion/calculadora.html)>.

[ 8 ] Ecopetrol et al. Metodología para la identificación y cuantificación de emisiones fugitivas de metano en campos de producción, 2014. 66 páginas. ISBN: 978-958-9287-.35-4. Consultado en Julio de 2019. Disponible para consulta: <[http://www.minambiente.gov.co/images/cambioclimatico/pdf/Geses\\_efecto\\_invernadero/Gu%C3%ADa\\_Metano\\_con\\_ISBN\\_\\_pdf.pdf](http://www.minambiente.gov.co/images/cambioclimatico/pdf/Geses_efecto_invernadero/Gu%C3%ADa_Metano_con_ISBN__pdf.pdf)>

[ 9 ] GÉRALD BARRÉ. Meziantou's Blog – Blog about Microsoft technologies (.NET, ASP.NET core, Blazor, EF core, WPF, Typescript, etc) [Consultado en Mayo de 2024] Disponible para consulta : <<https://www.meziantou.net/quick-introduction-to-xunitdotnet.htm>>

[ 10 ] GUIA DISEÑO DE SERVICIOS REST [Anónimo]. [s.l.]: [s.n.]. Informe de Parques Nacionales Naturales de Colombia - Código: GTSI\_GU\_02. [Consultado en Agosto de 2023]. Disponible para consulta: <[https://old.parquesnacionales.gov.co/portal/wp-content/uploads/2022/08/gtsi\\_gu\\_02\\_guia-diseno-de-servicios-rest\\_v\\_1.pdf](https://old.parquesnacionales.gov.co/portal/wp-content/uploads/2022/08/gtsi_gu_02_guia-diseno-de-servicios-rest_v_1.pdf)>

[ 11 ] Home 2024 - OpenAPI Initiative. Consultado en Abril de 2024. Disponible para consulta: <<https://www.openapis.org/>>

[ 12 ] IDEAM, PNUD, MADS, DNP, CANCELLEERÍA, 2015. Inventario Nacional de Gases de Efecto Invernadero (GEI) de Colombia. Tercera Comunicación Nacional de Cambio Climático de Colombia. IDEAM, PNUD, MADS, DNP, CANCELLEERÍA, FMAM, Bogotá, Colombia, 2015. 36 páginas. ISBN: 978-958-8902-94-4. Consultado en Julio de 2019. Disponible para consulta: <[https://documentacion.ideam.gov.co/openbiblio/bvirtual/023421/cartilla\\_INGEI.pdf](https://documentacion.ideam.gov.co/openbiblio/bvirtual/023421/cartilla_INGEI.pdf)>

[ 13 ] IDEAM, PNUD, MADS, DNP, CANCELLEERÍA, 2016. Inventario nacional y departamental de Gases de Efecto Invernadero - Colombia. Tercera Comunicación Nacional de Cambio Climático. IDEAM, PNUD, MADS, DNP, CANCELLEERÍA, FMAM, Bogotá D.C., Colombia, 2016. 73 páginas. ISBN obra impresa: 978-958-8971-25-4. Consultado en Julio de 2019. Disponible para consulta: <[https://www.undp.org/content/dam/colombia/docs/cop22/IDEAM\\_TCNCC\\_INGEI\\_WEB.pdf](https://www.undp.org/content/dam/colombia/docs/cop22/IDEAM_TCNCC_INGEI_WEB.pdf)>

[ 14 ] INITIATIVE, Global Methane. Metano de los Sistemas de Petróleo y Gas: Reducción de las Emisiones, Avance de la Recuperación y Utilización. EPA - United States Environmental Protection Agency [página web]. (Septiembre, 2011). [Consultado en marzo de 2020]. Disponible para consulta: <[https://www.globalmethane.org/documents/oil-gas\\_fs\\_spa.pdf](https://www.globalmethane.org/documents/oil-gas_fs_spa.pdf)>

[ 15 ] JORGENSEN, Paul C. Software Testing A Craftsman's Approach. 4a ed. Boca Ratón, Florida: CRC Press Taylor & Francis Group, 2014. 470 páginas.

[ 16 ] Marco Tulio Valente. Software Engineering: A Modern Approach, 2024.

[ 17 ] MYERS, GLENFORD J.; BADGETT, TOM y SANDLER, COREY. THE ART OF SOFTWARE TESTING. 3a ed. Hoboken, New Jersey: John Wiley & Sons, Inc, 2012. 254 páginas. ISBN 978-1-118-03196-4.

[ 18 ] PULIDO GUÍO, ANA DERLY y SIERRA URREGO, MAGDA MALLEN. CAPÍTULO DOS - MÓDULO DE ENERGÍA. MinAmbiente [página web]. (2009). 50 páginas. [Consultado en febrero de 2023]. Disponible para consulta: <[https://archivo.minambiente.gov.co/images/cambioclimatico/pdf/Inventario\\_del\\_GEI\\_2000\\_-\\_2004/CAP.\\_2\\_MODULO\\_ENERGIA.pdf](https://archivo.minambiente.gov.co/images/cambioclimatico/pdf/Inventario_del_GEI_2000_-_2004/CAP._2_MODULO_ENERGIA.pdf)>.

[ 19 ] PUERTA GONZÁLEZ, José Manuel. Desarrollo de una API para la descripción y gestión de Servicios Web REST [en línea]. Trabajo fin de máster. Castelló, España: Universitat Jaume I, 2015. 66 páginas. Disponible en Repositori Institucional de la Universitat Jaume I: <<https://core.ac.uk/download/pdf/61473672.pdf>>.

[ 20 ] Sommerville, Ian. Software engineering. 9na edición. Editorial Pearson education, Inc. 2011. 792 páginas. ISBN-13: 978-0137035151

[ 21 ] TUTORIAL: CREATE a web API with ASP.NET Core [Anónimo]. Microsoft Learn: Build skills that open doors in your career [página web]. [Consultado en Mayo de 2024]. Disponible para consulta: <<https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-8.0&tabs=visual-studio>>.

[ 22 ] United Nations Framework Convention on Climate Change (UNFCCC). Grupo consultivo de expertos sobre las comunicaciones nacionales de las partes no incluidas en el anexo I de la convención (GCE) Manual sobre el sector de la energía. Emisiones fugitivas. United Nations Climate Change [página web] 42 páginas. [Consultado en Julio de 2019]. Disponible para consulta: <<https://unfccc.int/sites/default/files/8-bis-handbook-fugitive-emissions.pdf>>.

[ 23 ] United Nations Framework Convention on Climate Change (UNFCCC). Grupo consultivo de expertos sobre las comunicaciones nacionales de las partes no incluidas en el anexo I de la convención (GCE) Manual sobre el sector de la energía. Quema de Combustibles. United Nations Climate Change [página web] 43 páginas. [Consultado en

Julio de 2019]. Disponible para consulta: <<https://unfccc.int/sites/default/files/7-bis-handbook-on-energy-sector-fuel-combustion.pdf>>.

[ 24 ] United Nations Framework Convention on Climate Change. GHG Inventory Software Manual. UNFCCC non-Annex I Greenhouse Gas Inventory Software, 2007. 31 páginas. Consultado en Julio de 2019. Disponible para consulta: <[http://unfccc.int/resource/cd\\_roms/na1/ghg\\_inventories/unfccc\\_software/user\\_manual/Software\\_Manual.pdf](http://unfccc.int/resource/cd_roms/na1/ghg_inventories/unfccc_software/user_manual/Software_Manual.pdf)>