



SCAI MANUAL TÉCNICO

ANTECEDENTES	3
METODOLOGÍA	4
EQUIPO DE TRABAJO	5
REQUISITOS DEL SISTEMAS	6
Requisitos de Hardware	6
Requisitos de Software	7
DISEÑO DE SOFTWARE	9
Diseño de Componente de Datos	9
Diseño de Componente de Interfaz de Usuario (UI)	9
Diseño de Componente de Lógica de Negocio	11
Diseño de Componente de Comunicación y Redes	12
Diseño de Componente de Seguridad	12
Diseño de Componente de Pruebas	14
ARQUITECTURA	15
INFRAESTRUCTURA	17
Enfoque de Desarrollo utilizando ORM	17
Enfoque de Desarrollo utilizando WebServices y Procedimientos	18
INSTALACIÓN Y CONFIGURACIÓN	19
Introducción	19
Instalación para el servidor de aplicación	19
Configuración de arquitectura	20
GUÍA DE DESARROLLO	23
Visual Studio Team Foundation Server (TFS)	23
Estándares de Base de Datos	25
Autenticación	25
MANTENIMIENTO Y SOPORTE	26
ANEXO	27

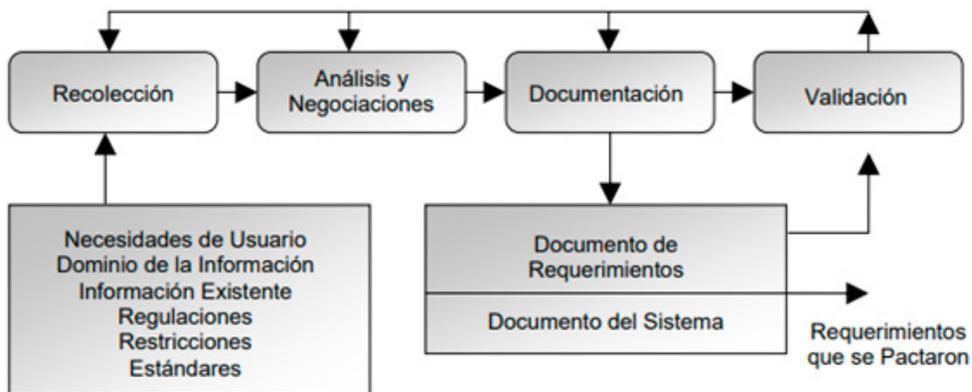
El propósito de este manual es describir los aspectos técnicos de una forma clara y concisa sobre el proceso de análisis, diseño, desarrollo e implementación del Sistema de Control y Seguimiento de Auditorías Internas del sector público (SCAI).

El Sistema de Control y Seguimiento de Auditorías Internas del sector público (SCAI), iniciativa presentada por la Dirección Nacional de Auditoría General (DINAG) de la Contraloría General de la República de Panamá (CGR), que tiene por objetivo contar con una herramienta digital a disposición de las Unidades de Auditoría Interna de las Entidades Gubernamentales que brindarán información relevante, oportuna, completa y precisa de su perfil de Recurso Humano, de los Resultados de las Auditorías Internas y el cumplimiento del Plan Anual de Auditorías Internas para enviarlo de forma más eficiente para el análisis de la CGR para la toma de decisiones en la definición de estrategias para capacitaciones de la UAI y creación de indicadores para el reforzamiento de estas unidades de control interno.

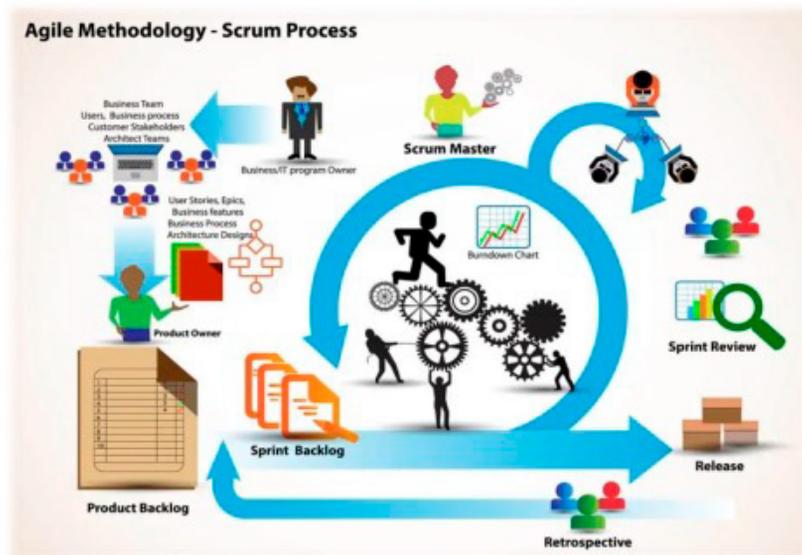
La iniciativa se desarrolla en el marco del Proyecto regional “Fortalecimiento del Control Externo para la prevención y combate eficaz contra la corrupción”, implementado por la Cooperación Alemana, por medio de la GIZ, en alianza con la Organización Latinoamericana y del Caribe de Entidades Fiscalizadores Superiores (OLACEFS). El apoyo provee contratación de consultoría como soporte técnico al equipo de desarrolladores de la Contraloría General de la República de Panamá en la ejecución de las actividades correspondientes al desarrollo del SCAI.

El inicio del proyecto por parte de la CGR se cuenta a partir de abril de 2023, donde se procede a recopilar y analizar los requerimientos para el SCAI, y a partir de agosto de 2023 se inició el desarrollo de esta solución tecnológica utilizando herramientas y recursos propios de la CGR a través de la Dirección Nacional de Tecnología de la Información y Comunicaciones (DNTIC). El 15 de noviembre de 2023 se suma al equipo el apoyo técnico otorgado por la GIZ.

La Dirección Nacional de Tecnología de la Información y Comunicaciones (DNTIC) utiliza SCRUM como marco de trabajo para el desarrollo de Software basado en un proceso ágil, iterativo e incremental, donde en una primera etapa se recopila información para el análisis de requerimientos a través del rol de Product Owner utilizando herramientas tales como cuestionarios, reuniones de trabajo, encuestas, entre otras. Esta actividad se repite mientras dure la ejecución del proyecto.



En la medida que se vayan pactando los requerimientos a desarrollar con el usuario Recor (DINAG), se van ingresando al TFS (Team Foundation Server - sistema para la administración de proyectos y de código fuente de la DNTIC), se asigna un equipo de diferentes departamentos que incluyen los roles de Scrum Master y el Equipo de desarrollo.



Product Owner

1 analista que interactúa con las partes interesadas, los usuarios finales y el equipo de Scrum para asegurarse de que el producto final cumpla con los requisitos del usuario y se alinee con los objetivos del proyecto.

Scrum Master

1 guía técnico, facilitador y responsable de resolver los impedimentos externos al Equipo Scrum.

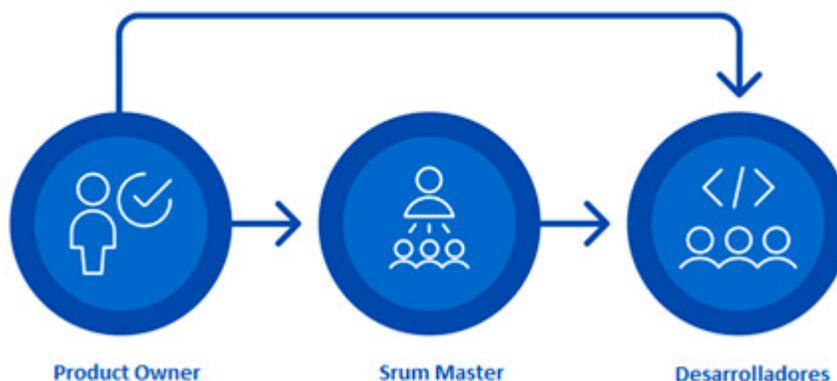
Equipo de Desarrolladores

Conformado por 4 programadores, 3 QA, 1 DBA y 1 Administrador de Servidores. Encargados de crear un incremento terminado a partir de los elementos del Product Backlog seleccionados como Sprint Backlog durante el Sprint Planning.

Consultor

1 especialista en desarrollo de software encargado de analizar el código desarrollado, sugerir mejores prácticas, ayudar al equipo de desarrollo con la detección y resolución de problemas, desarrollar nuevos componentes. Potenciar al equipo con sesiones de capacitación en nuevos enfoques de desarrollo eficientes. Establecer un cronograma de trabajo con metas a corto plazo y estrategias para alcanzarlas en un periodo de 5 meses.

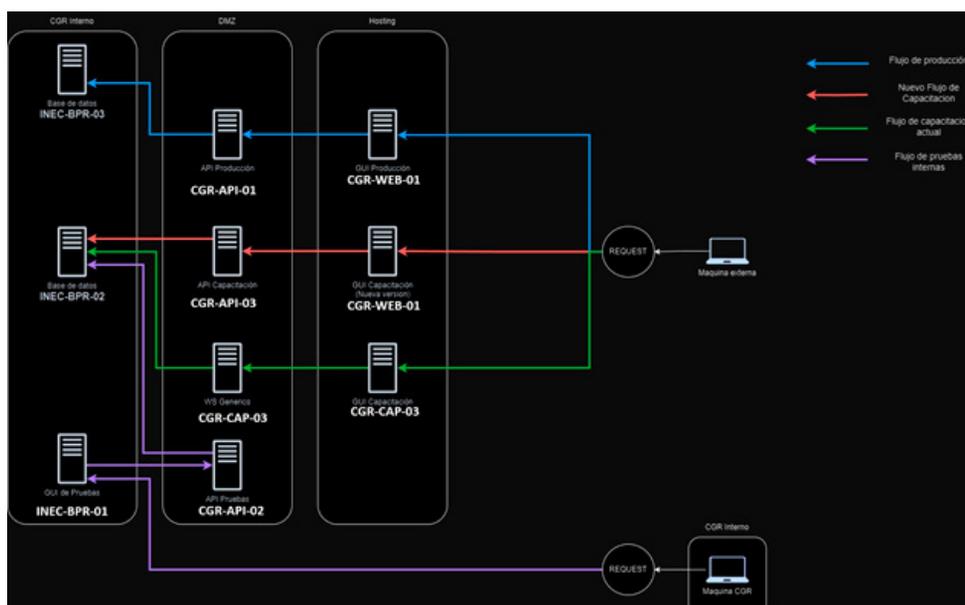
Equipo en el Scrum



1. Requisitos de Hardware

1.1 Servidores de Aplicaciones y Base de Datos

Servidores de PROYECTO SCAI						
Servidores	Sistema Operativo	CPU	Memoria	Disco	Capacidad	Descripción
CGR-API-01	Microsoft Windows Server 2022 Datacenter	2	4 GB	Hard disk 1	40 GB	Producción
CGR-API-02	Microsoft Windows Server 2022 Datacenter	2	4 GB	Hard disk 1	40 GB	Prueba
CGR-APR-01 (1)	Microsoft Windows Server 2019 Datacenter	4	12 GB	Hard disk 1	61 GB	Aplicación prueba
				Hard disk 2	15	
INEC-BPR-03	Microsoft Windows Server 2012 (64-bit)	2	8 GB	Hard disk 1	122 GB	BD Producción
INEC-BPR-01	Microsoft Windows Server 2012 (64-bit)	1	4 GB	Hard disk 1	102 GB	BD Desarrollo
INEC-BPR-02	Microsoft Windows Server 2012 (64-bit)	4	4 GB	Hard disk 1	102	BD Prueba
CGR-WEB-01	Microsoft Windows Server 2019 Datacenter	10	16 GB	Hard disk 1	143 GB	Aplicación Producción hosting
				Hard disk 2	40 GB	Aplicación Prueba hosting



2. Requisitos de Software

2.1 Sistemas de autenticación y autorización

Es fundamental garantizar la seguridad y la privacidad de los datos, así como controlar el acceso a las diferentes funciones y recursos del sistema. En la CGR utilizamos Sistema de Seguridad Externa para Usuarios (SSAE) para todas las autenticaciones y autorizaciones de usuarios de todas las aplicaciones desarrolladas por la DNTIC.

- **Autenticación:** Contar con un sistema que permita identificar a los usuarios de manera segura, utilizar credenciales de usuario, contraseña, token de seguridad, entre otros.
- **Autorización:** Contar con un sistema o base de datos para el control de acceso de los usuarios, sus permisos y a qué recursos pueden acceder dentro del sistema. SCAI trabaja con asignación de roles, que a su vez permiten el acceso a un grupo determinado de opciones dentro del sistema.

2.2 Entorno de Desarrollo Integrado (IDE)

Microsoft Visual Studio Professional 2022 (64-bit): Entorno principal para el desarrollo del proyecto con herramientas avanzadas de desarrollo y depuración.

2.3 Desarrollo de API Rest

API Rest desarrollada con C# y Entity Framework con .NET 8: Se utilizó para el desarrollo de servicios web eficientes y escalables. Uso del LINQ (Language Integrated Query) para la consulta de datos, garantizando un acceso ágil a la información almacenada.

2.4 Formato de Datos

JSON (JavaScript Object Notation): Es el formato de datos para la comunicación entre componentes del sistema, asegurando la interoperabilidad y la eficiencia en el intercambio de información.

2.5 Interfaz de Usuario

Se desarrolló utilizando WebForms con VB.NET en .NET Framework 4.7.2: Esta versión es compatible con sistemas operativos Windows y proporciona una plataforma sólida para aplicaciones web basadas en WebForms. Si bien WebForms es una tecnología más antigua, permite construir aplicaciones web robustas con una rica interacción de usuario, lo cual era un requisito clave para SCAI.

2.6 Base de Datos

SQL Server 2016: Uso del sistema de gestión de bases de datos relacional para almacenar y administrar los datos de manera eficiente y segura.

2.7 Versión Híbrida

Manejo de API y Webservice: La versión híbrida del software permite la integración y el manejo tanto de la API Rest desarrollada con .NET 8 como de un Webservice WCF con .NET Framework 4.7.2. asegurando la flexibilidad y la interoperabilidad del sistema en diferentes entornos y escenarios de implementación.

1. Diseño del Componente de Datos

Estructurar cómo se almacenan, organizan y acceden los datos.

- ANEXO - Diagramas ER (Entidad-Relación) SCAI.

2. Diseño del Componente de Interfaz de Usuario (UI)

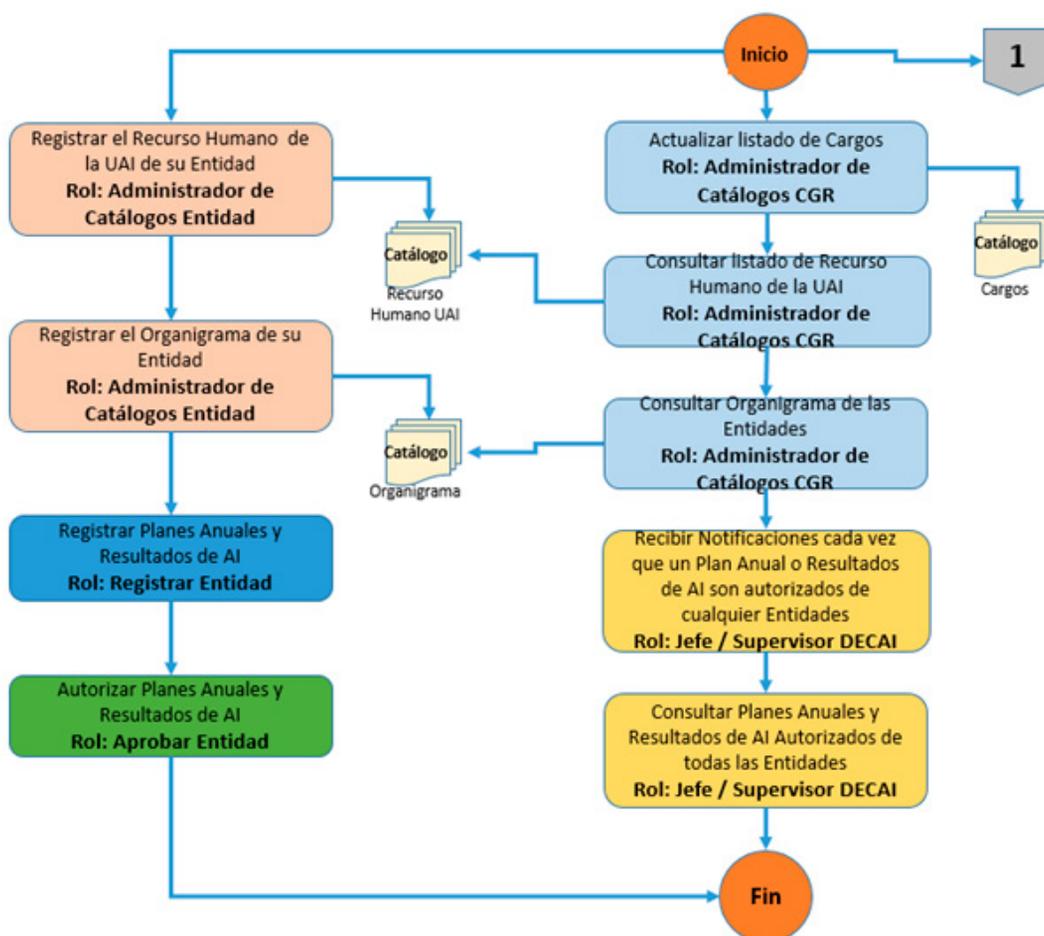
Esbozos de las pantallas y componentes visuales.



Interfaces intuitivas y fáciles de usar.

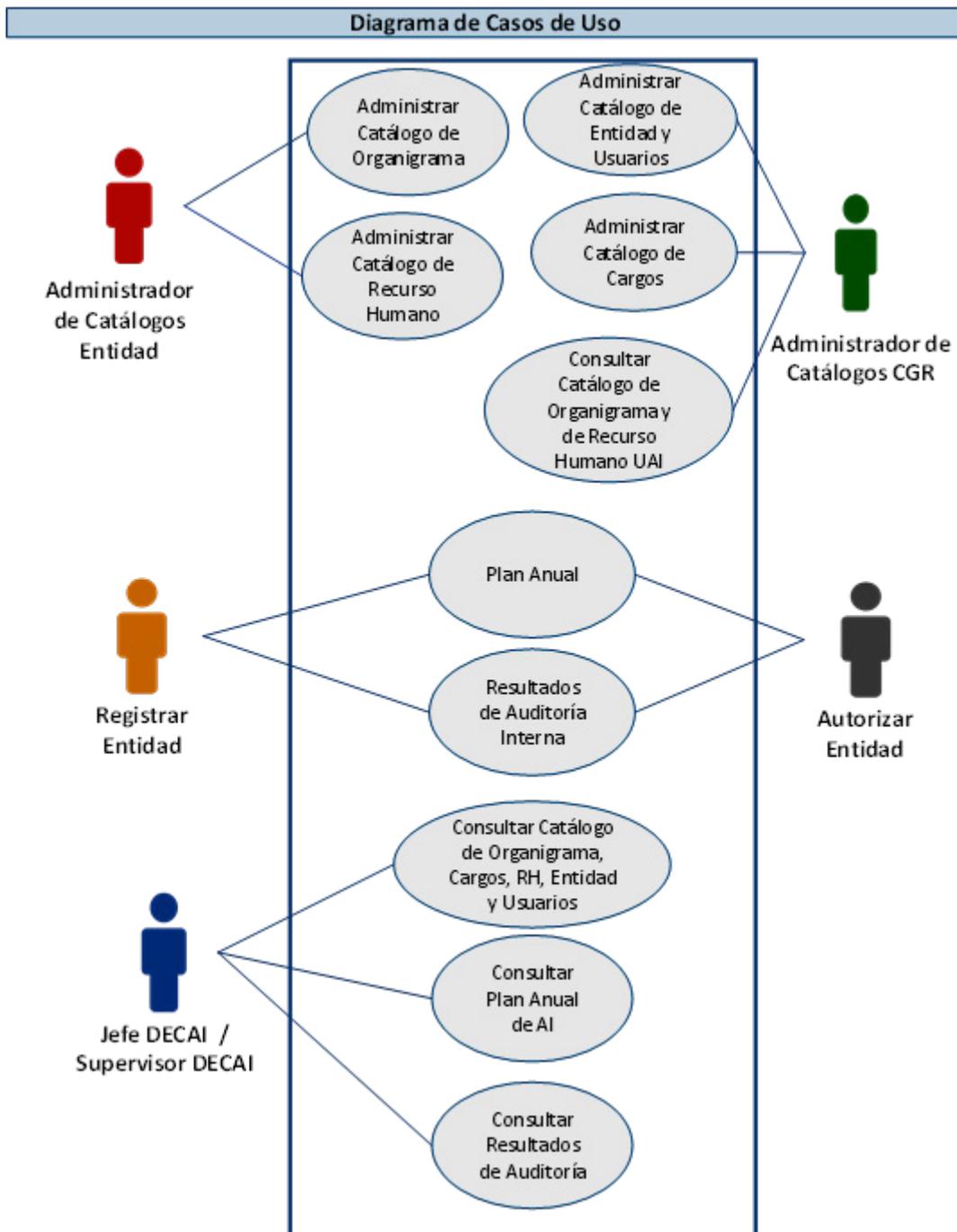
A screenshot of a web form titled 'Recurso Humano UAI' under the heading 'Acciones de Personal'. The form contains several input fields: 'Nombre' (160/160 caracteres), 'Cédula' (with dropdown arrows), 'Correo Electrónico', 'Fecha de Nacimiento' (with a calendar icon), 'Género' (radio buttons for 'Masculino' and 'Femenino'), 'Universidad' (dropdown), 'Nivel Académico' (dropdown), and 'Cargo que Ocupa' (dropdown). There is an 'Agregar' button next to the 'Nivel Académico' field. Below the form, there is a table with columns 'Universidad', 'Nivel Académico', and 'Titulo'. At the bottom, there are 'Regresar' and 'Guardar' buttons.

Diagramas que describe la navegación del usuario a través del sistema de la Etapa 1 del proyecto el cual incluye etapas posteriores identificadas a través del número 1 que se desarrollarán posteriormente donde intervienen otros roles de la CGR para la administración y atención posterior a los Resultados de las Auditoría Internas.



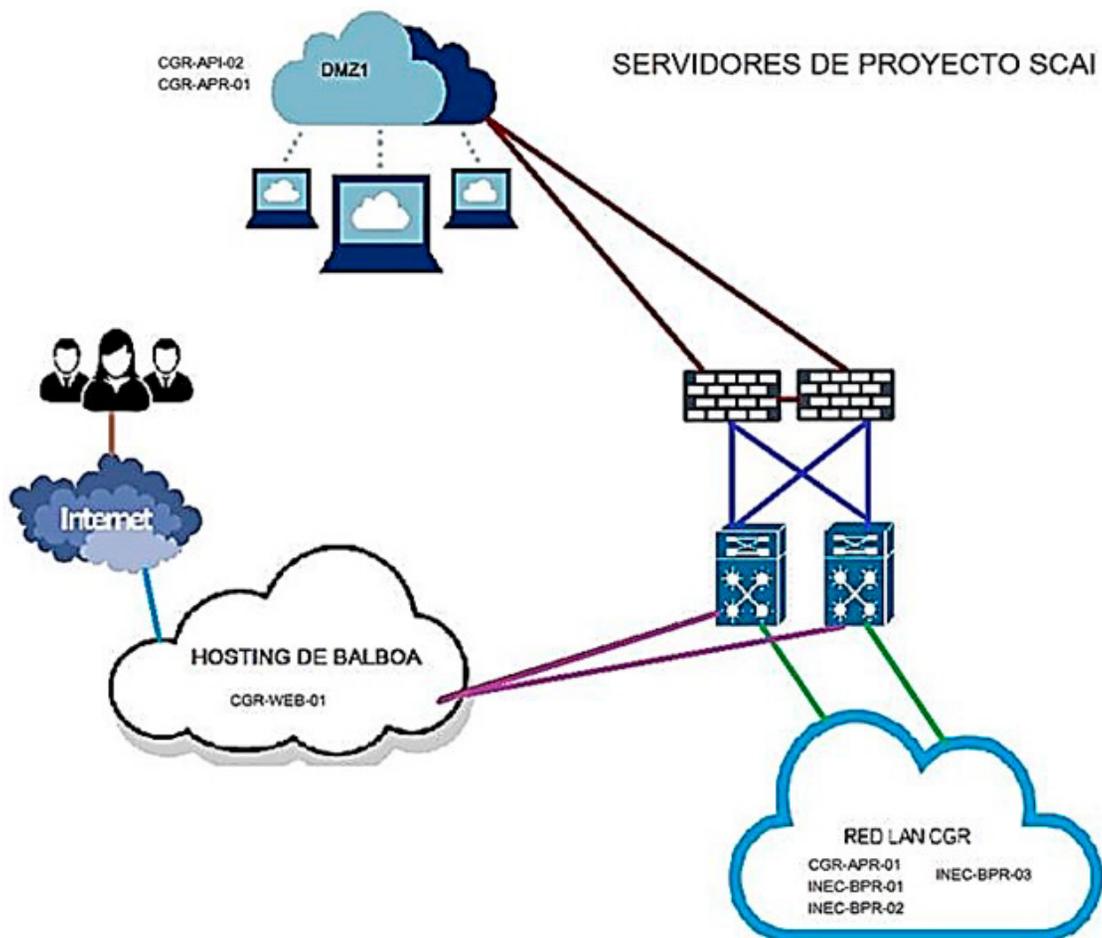
3. Diseño del Componente de Lógica de Negocio

Escenarios que describen cómo interactúan los usuarios con el sistema.



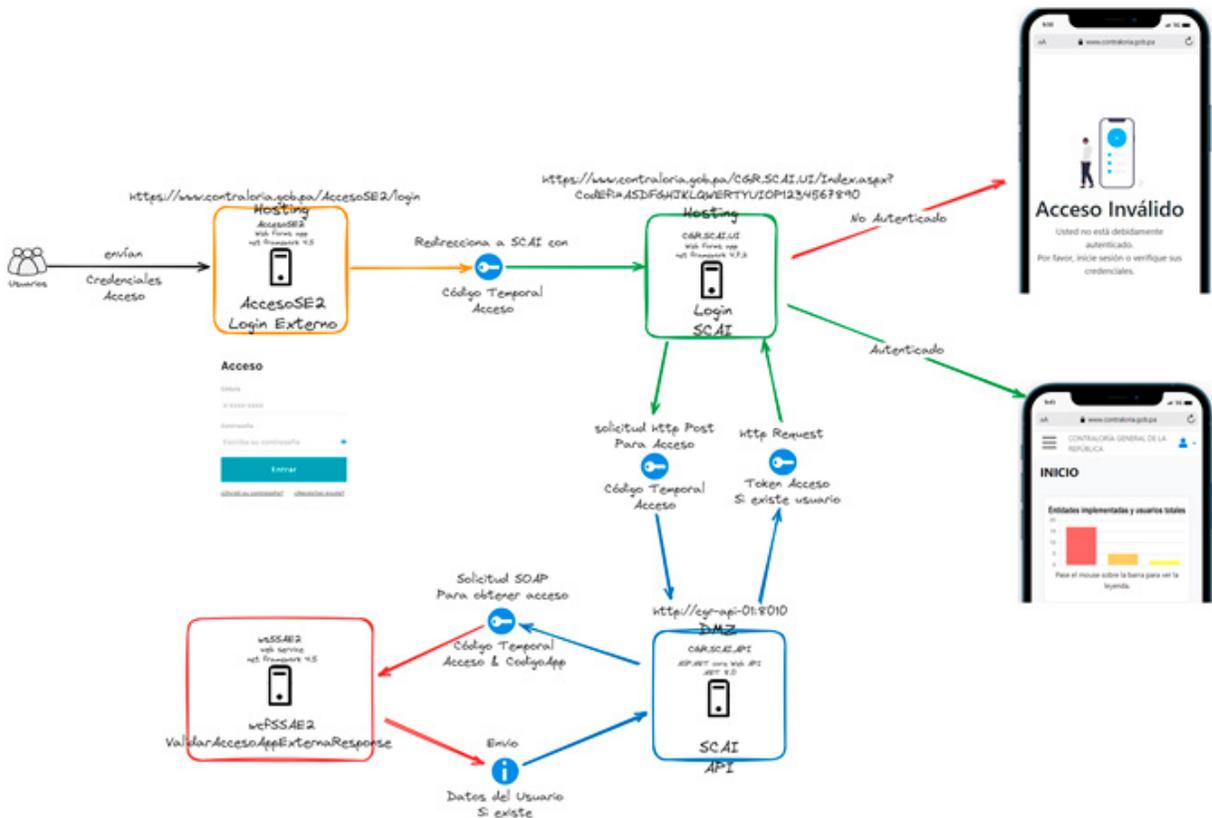
4. Diseño del Componente de Comunicación y Redes

Asegura que los componentes se comuniquen efectivamente a través de una red.



5. Diseño del Componente de Seguridad

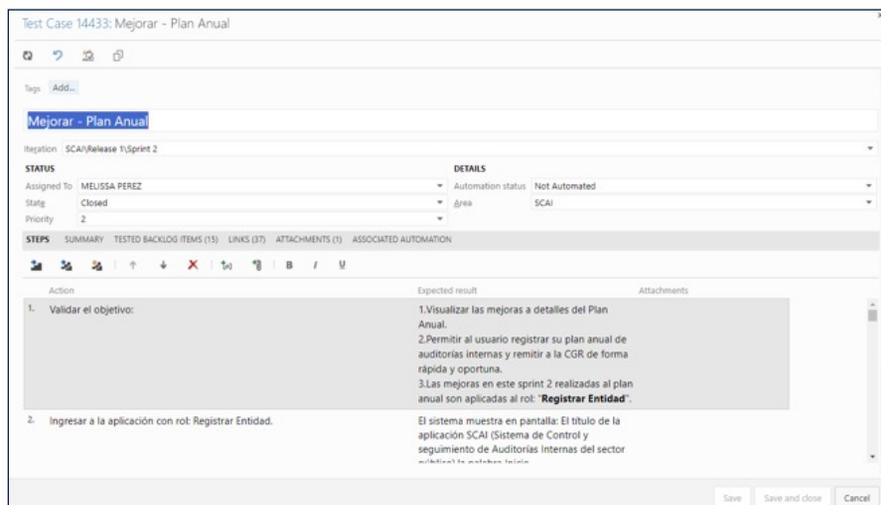
Componentes que protegen el sistema y los datos contra accesos no autorizados y otras amenazas. Mecanismos de autenticación y autorización.



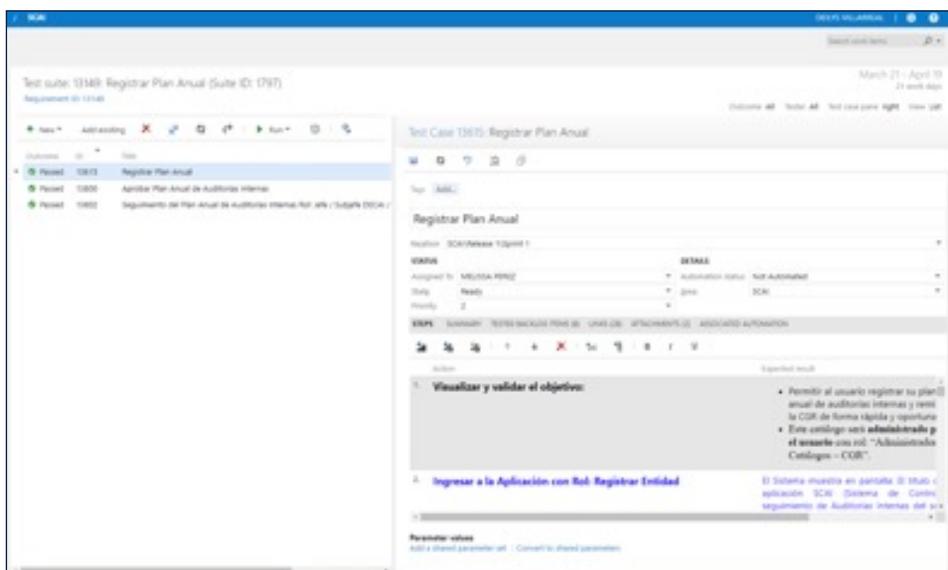
6. Diseño del Componente de Pruebas

Para asegurar que el software funcione correctamente bajo diversas condiciones:

- Se crean escenarios de pruebas en el TFS donde se registran los criterios de aceptación de cada requerimiento.



- A través de la herramienta Test Manager del TFS se configuran los entonos específicos para realizar pruebas para la verificación de las pruebas.



1. Capa de Presentación

- Framework: .NET Framework 4.7.2
- Tecnología: WebForms
- Lenguaje: VB.NET
- Funcionalidad: Manejo de la interfaz de usuario, validaciones y lógica de presentación.
- Comunicación: Las peticiones desde el cliente al backend se realizan utilizando JavaScript y el formato de datos JSON.

2. Capa de Lógica de Negocio

2.1 API Rest

- Framework: .NET 8
- Lenguaje: C#
- ORM: Entity Framework (EF)
- Funcionalidad: Implementación de servicios web eficientes y escalables. Realiza consultas a la base de datos utilizando LINQ.

2.2 WebService WCF

- Framework: .NET Framework 4.7.2
- Lenguaje: VB.NET
- Funcionalidad: Realiza la misma funcionalidad que la API Rest, pero ejecuta procedimientos almacenados directamente en la base de datos SQL Server.

3. Capa de Datos

- Sistema de Gestión de Base de Datos: SQL Server 2016
 - Funcionalidad: Almacenamiento y gestión de datos del sistema.
 - API Rest: Accede a la base de datos utilizando Entity Framework.
 - WebService WCF: Accede a la base de datos ejecutando procedimientos almacenados de SQL.
-

4. Capa de Comunicación

- Formato de Datos: JSON
- Protocolo: HTTP/HTTPS
- Interoperabilidad: Asegura la interoperabilidad y eficiencia en el intercambio de información entre el frontend y el backend.

5. Herramientas de Desarrollo

- Entorno de Desarrollo Principal: Microsoft Visual Studio Professional 2022 (64-bit)
 - IDE del Consultor: Visual Studio Code
-

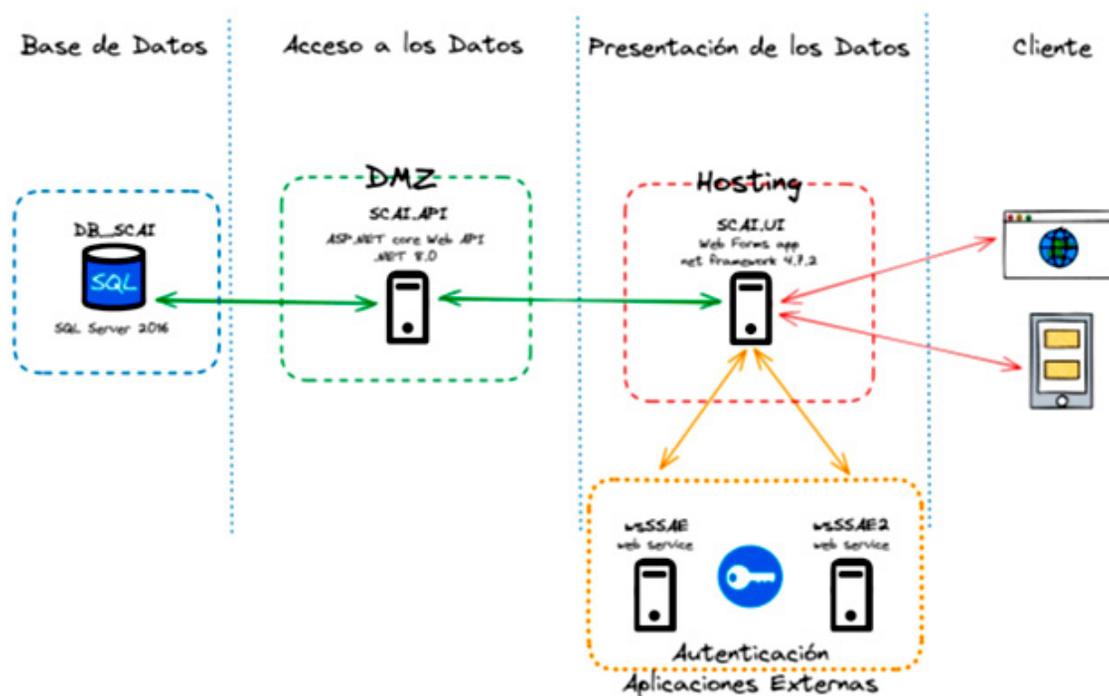
1. Enfoque de Desarrollo utilizando ORM (Object Relational Mapping)

1.1 Requisitos para esta Infraestructura:

- 1 Servidor Windows Server destinado para alojar la Interfaz de Usuario (UI).
- 1 Servidor multiplataforma para alojar el API REST desarrollado en .NET Core.
- 1 Base de datos cualquiera compatible con el ORM utilizado (como SQL Server, PostgreSQL, MySQL, etc.).

1.2 Flujo de Funcionamiento utilizando ORM:

- El cliente inicia solicitudes AJAX desde la Interfaz de Usuario (UI).
- La UI realiza una solicitud a un endpoint del API REST desarrollado en .NET Core.
- Utilizando un ORM, el API REST efectúa la consulta correspondiente a la base de datos.
- La base de datos procesa la solicitud y devuelve los datos al API REST.
- El API REST devuelve los datos recuperados a la UI.
- La UI procesa y presenta los datos al cliente.



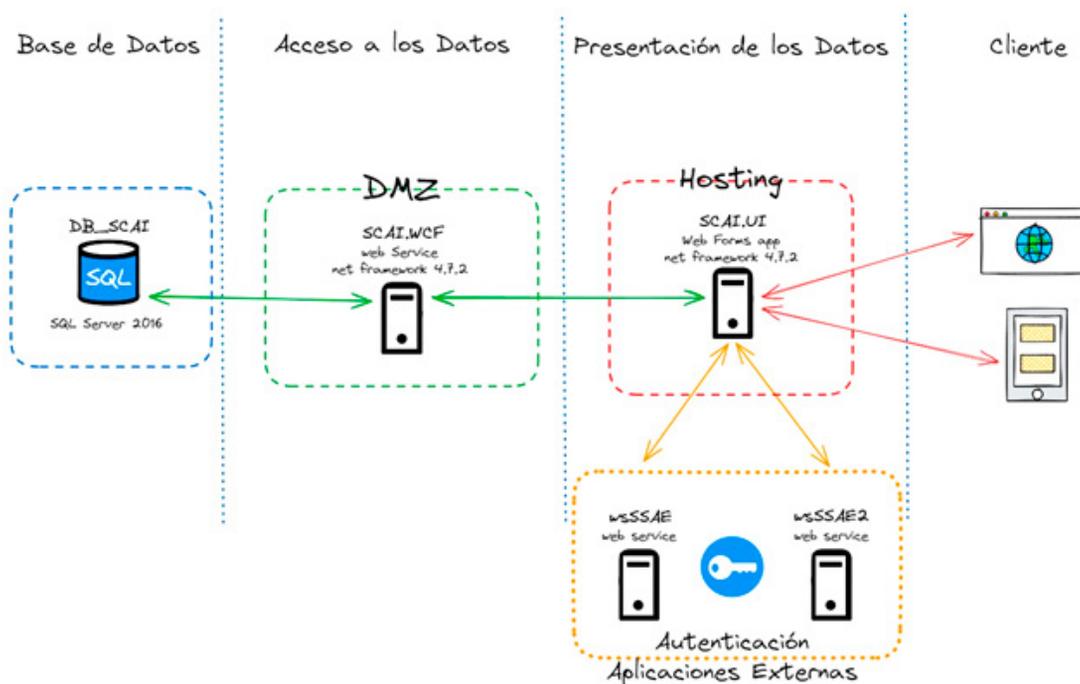
2. Enfoque de Desarrollo Utilizando WebServices y Procedimientos Almacenados

2.1 Requisitos de Infraestructura:

- 2 servidores Windows Server para alojar la aplicación (uno para la UI y otro para el Webservice).
- 1 servidor dedicado para la base de datos SQL Server.
- Se requiere SQL Server para un funcionamiento óptimo de esta infraestructura.

2.2 Flujo de Funcionamiento:

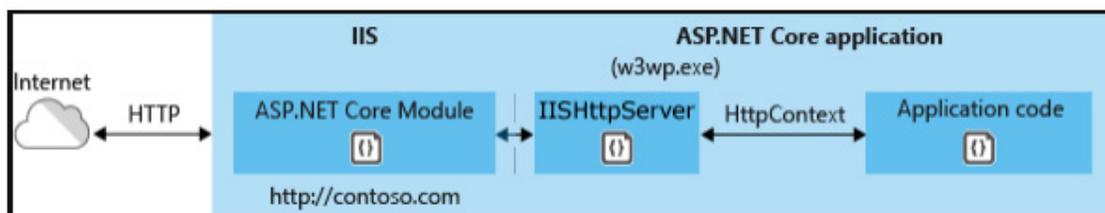
- El cliente genera solicitudes AJAX desde la Interfaz de Usuario (UI).
- La UI envía la petición a un Webservice.
- El Webservice ejecuta un procedimiento almacenado en la base de datos.
- La base de datos procesa la solicitud y devuelve un conjunto de resultados al Webservice.
- El Webservice retorna la respuesta en formato DataTable a la UI.
- La UI procesa y muestra los datos al cliente.



1. Introducción

Las aplicaciones hechas con ASPNET Core vienen con un servidor web por defecto llamado Kestrel que se encarga del tráfico http de la aplicación, este servidor termina compitiendo con IIS para el manejo del tráfico y es necesario acordar cómo será el funcionamiento entre ellos para que la aplicación se ejecute correctamente en IIS.

Existen 2 arquitecturas para acordar el manejo del tráfico que son: In-Process hosting en el que el builder llama directamente a la interfaz de IIS para hacer las llamadas y Outof-Process hosting donde kestrel mantiene el control y IIS funciona como un reverse proxy. La manera por defecto de desplegar una aplicación web de ASPNET Core en IIS es a través de la arquitectura In-Process hosting para evitar el tráfico extra que requiere hacer el reenvío del proxy.

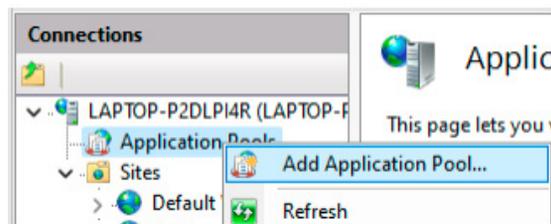


2. Instalación para el servidor de aplicación

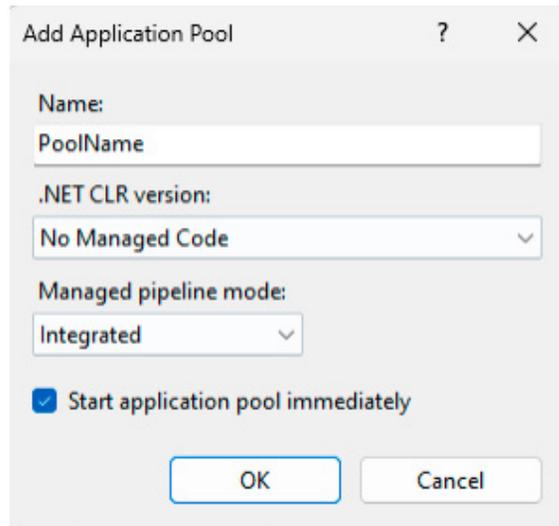
- IIS habilitado en el servidor
- Descargar e instalar el ASP.NET Core 8.0 Runtime (v8.0.1)
- Windows Hosting Bundle Installer en el servidor
- (<https://dotnet.microsoft.com/en-us/download/dotnet/thank-you/runtime-aspnetcore-8.0.1-windows-hosting-bundle-installer>)
- Dotnet 8 SDK instalado en la máquina de desarrollo
- Variable de ambiente asignada ASPNETCORE_ENVIRONMENT=<Local/Development/Production/CApacificacion/

3. Configuración de arquitectura

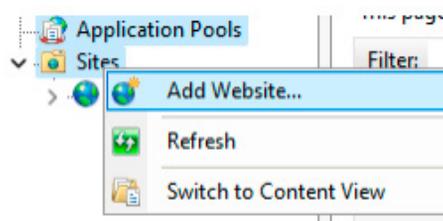
Dentro de la ruta C:\inetpub crearemos una carpeta donde estarán alojados los archivos compilados de la aplicación. En el menú izquierdo de IIS buscamos Application pools y le damos clic derecho y luego Add Application pools.



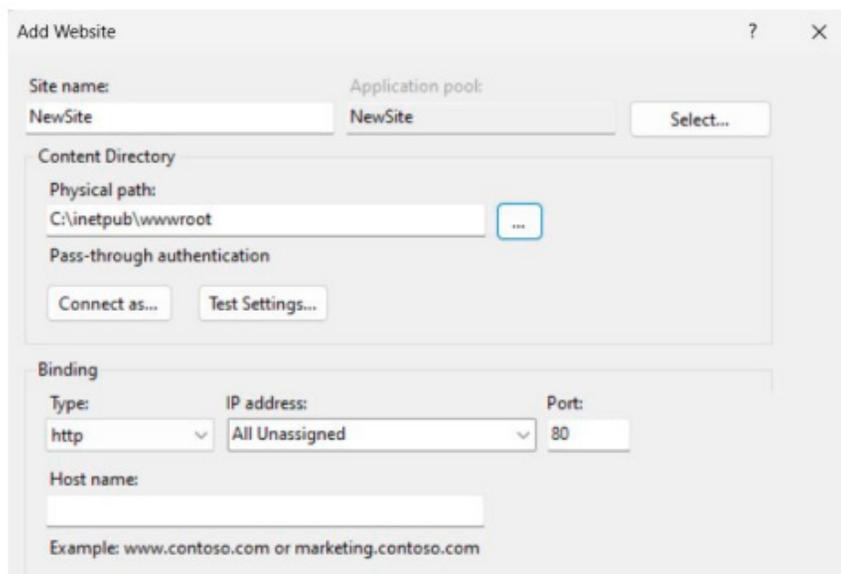
Debe darse un nombre a este nuevo pool y configurarse con el .net CLR versión como No Managed Code y Managed Pipeline Mode como Integrated.



En el menú izquierdo de IIS damos clic derecho a Sites y damos en Add website (también podemos usar el default website si no está ya en uso).



Definimos el nombre del sitio, seleccionamos el Application pool que creamos, en physical path elegimos la ruta a la carpeta que creamos y seleccionamos el protocolo y puerto que utilizaremos, en este caso puede ser https y 443 siempre y cuando se tenga un certificado para utilizar.



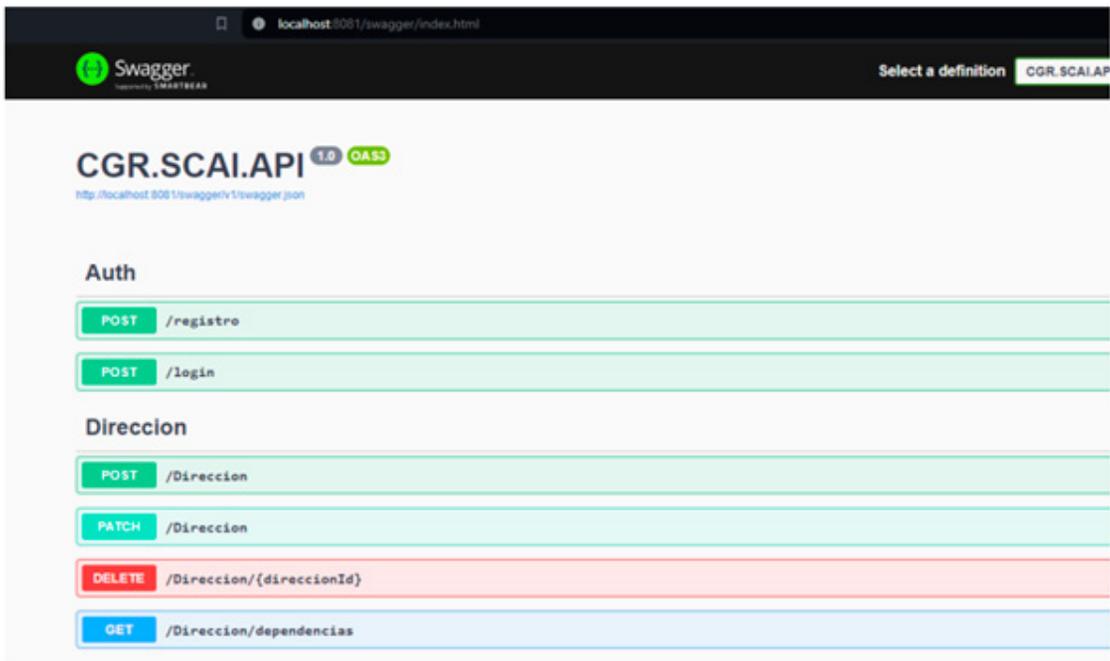
Dentro de la carpeta de la solución, abrimos una consola de CMD o powershell y ejecutamos el siguiente comando `dotnet publish -c Release` este comando va a compilar el código y a generar los DLL necesarios para que funcione el servicio, estos archivos se pueden encontrar en la ruta `<Dirección del proyecto principal>\CGR.SCAI.API\bin\Release\net8.0\publish`, debemos copiar todo el contenido de la carpeta publish y colocarlo en la carpeta creada anteriormente para el sitio en IIS.

Recordemos que el código compilado incluirá todos los archivos de configuración `appsettings.json`, es recomendable que solo quede el que le corresponde al ambiente que están creando para evitar confusión.

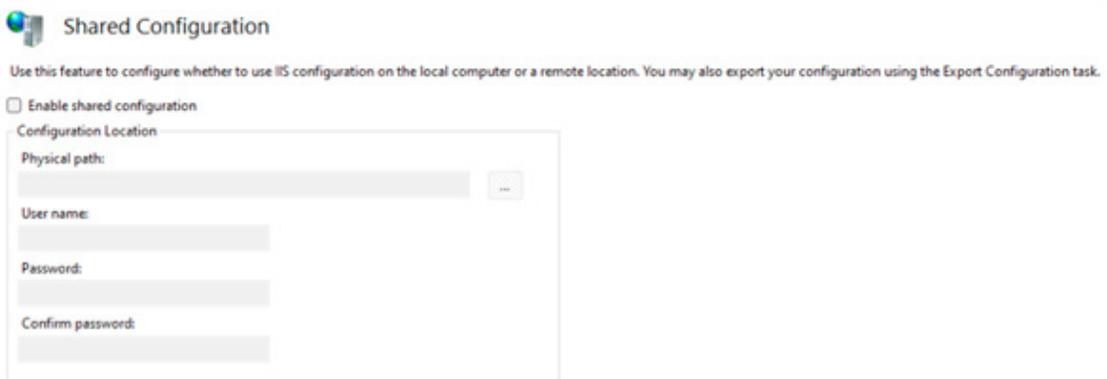
Luego de estos pasos, se debe reiniciar el sitio en IIS y esperar a que inicie con el nuevo código.

Si usamos un ambiente que no sea producción podemos entrar a la ruta `<server>/swagger/index.html` para comprobar que esté funcionando.

Sitio de prueba del API con Swagger UI



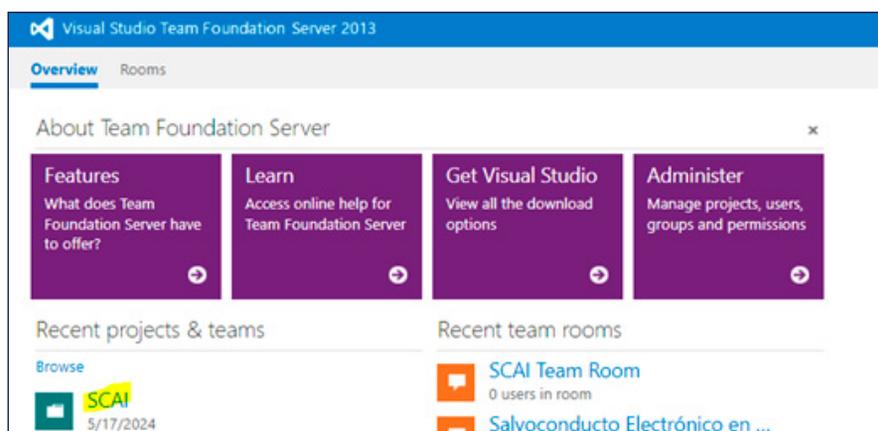
NOTA: Recordar que para poder instalar el dotnet host bundle es necesario que IIS tenga deshabilitada la opción de shared configurations, la cual puede ser habilitada de nuevo luego de la instalación.



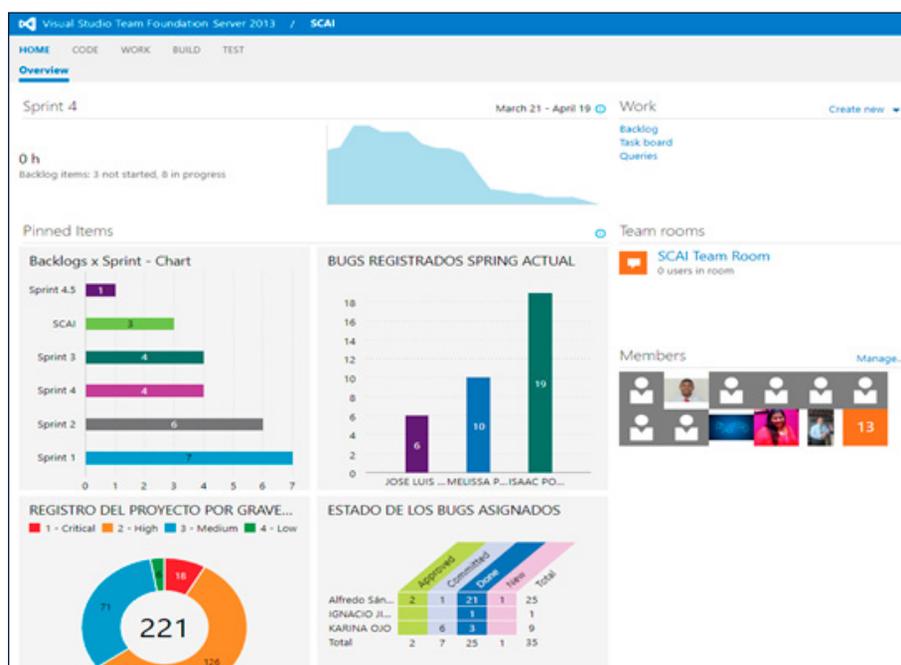
1. Visual Studio Team Foundation Server (TFS)

Plataforma de colaboración para la gestión del ciclo de vida de las aplicaciones (ALM) de Microsoft. Donde también se lleva el versionamiento de la aplicación:

- Creación del Team Project: Se asigna nombre y descripción del proyecto.



- Asignación de los miembros del Equipo al Team Project: Se da acceso a los miembros del equipo de desarrollo.



- Se registran los requerimientos del proyecto (Product Backlog Item): Se debe incluir título del requerimiento, esfuerzo, estado (New, Committed, Approved, Done, Removed)
- Los Product Backlogs recogen las historias de usuarios las cuales deben seguir el siguiente estándar:

Como (Rol): Tipo de usuario
Quiero: (Funcionalidad): Hacer
Para: (Resultado): Para Lograr

- Se adjunta a las Historias de usuario un esquema gráfico o StoryBaords.
- También se detallan los Criterios de Aceptación.
- Se adjunta cualquier otro documento que apoye el desarrollo del requerimiento.
- Se reunión de planificación se aclaran los requerimientos con los miembros del equipo.
- Se establece la capacidad de atención de los miembros del equipo.
- Otorgar puntuación a las historias de usuarios.
- Se establece la duración del sprint.
- Creación y asignación de tareas.

ID	Title	State	Assigned To
13142	Configuración Inicial	Done	Alfredo Sánchez
13366	Creación del proyecto en VS	Done	Alfredo Sánchez
13367	Crear Aplicación y Roles en SSAE en desarrollo	Done	Alfredo Sánchez
13368	Creación de la BD_SCAI en INEC-BPR-01(Ambiente de Desarrollo)	Done	Alfredo Sánchez
13358	CREACIÓN DE ESCENARIOS CONFIGURACIÓN INICIAL	Done	ISAAC POMARES
13143	Interfaz gráfica	Done	IGNACIO JIMENEZ
13392	Programación de la interfaz , diseño del webform	Done	IGNACIO JIMENEZ
13393	Programación del backend, procedimientos almacenados	Done	IGNACIO JIMENEZ
13360	Validar login de Usuario	Done	IGNACIO JIMENEZ
13363	CREACIÓN DE ESCENARIOS INTERFAZ GRÁFICA Y DASHBOARD	Done	ISAAC POMARES
13146	Catálogo de Recurso Humano	Done	IGNACIO JIMENEZ
13372	Análisis de las tareas solicitadas por Desarrollar	Done	IGNACIO JIMENEZ
13377	Programación de la interfaz , diseño del webform	Done	IGNACIO JIMENEZ
13382	Programación del backend, procedimientos almacenados	Done	IGNACIO JIMENEZ
13386	Programación del frontend, validaciones, métodos, funciones	Done	IGNACIO JIMENEZ
13591	Excel - Recurso Humano - Entidad - UsuarioCGR	Done	IGNACIO JIMENEZ
13651	CASO DE PRUEBA CATÁLOGO DE RECURSO HUMANO Administrador de Catálogos - ...	Ready	ISAAC POMARES
13779	CASO DE PRUEBA Administrador de Catálogo-CGR, Recurso Humano Jefe DECAI / Su...	Ready	ISAAC POMARES

2. Estándares de Base de Datos

Tablas

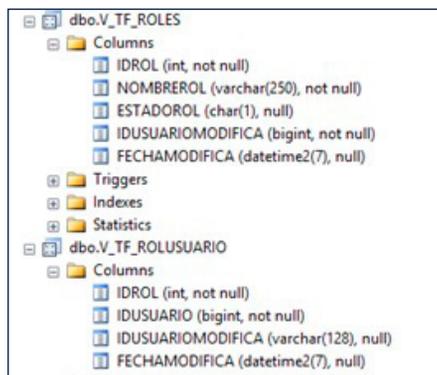
Tipo de Tabla	Estándar	Descripción
tablas maestras	dbo.TM_[NOMBRE]	por ejemplo dbo.TM_USUARIOS, en plural o singular
catálogos	dbo.TF_[NOMBRE]	por ejemplo dbo.TF_PROVINCIAS, en plural o singular
tablas cabeceras	dbo.TC_[NOMBRE]	por ejemplo dbo.TC_ORDENCOMPRA, en plural o singular
tablas detalle	dbo.TD_[NOMBRE]	por ejemplo dbo.TD_ORDENCOMPRA, en plural o singular

1. Todas las tablas deben contar con los siguientes 4 campos:
 - Usuario Registra , varchar(128) / Fecha Registra, datetime
 - UsuarioModifica , varchar(128) / Fecha Modifica, datetime
2. Analizar la composición de las llaves primarias (PK) ya sea utilizando un campo de tipo Identity u otros campos en la tabla.
3. Crear los índices requeridos Non Clustered.

3. Autenticación

Como regla general, la autenticación de los usuarios se realiza a través del Sistema de Seguridad de Aplicaciones Externas. Para toda nueva aplicación el DBA asignado al proyecto debe se debe realizar los siguientes pasos:

1. Abrir la aplicación “Sistema de Seguridad para Aplicaciones Externas”
2. Registrar la aplicación
3. Crear los roles identificados
4. Crear los usuarios y asignarles el o los roles
5. Crear las vista de roles y roles por usuario de requerirse



El mantenimiento de aplicaciones en la Contraloría General de la República (CGR) se realiza tanto para la atención de incidentes, como por solicitudes de los usuarios rectores de las aplicaciones. Estas actividades pueden incluir incidentes presentados durante el uso del SCAI o por nuevos requerimientos.

1. Proceso de Mantenimiento

- **Correcciones y Mejoras:** Derivadas de incidentes reportados, se corrige la aplicación o la Base de Datos. Antes de liberar a producción las correcciones o modificaciones realizadas se realizan las pruebas correspondientes.
- **Nuevos Requerimientos:** Solicitados por los usuarios rectores o por sugerencia de usuarios de las Unidades de Auditoría Interna en las Entidades y autorizado por los usuarios rectores.

2. Soporte y Atención a Incidentes

- **Grupo de Monitoreo:** Los usuarios rectores escogen un grupo de especialistas en el negocio, este grupo es el primer punto de contacto para atender cualquier consulta o duda de los auditores internos. Si el incidente es de tecnología, se reporta al Departamento de Apoyo Técnico al Usuario de la DNTIC para su atención.
- **Departamento de Apoyo Técnico al Usuario (DATU):** Este departamento se encarga de orientar a los usuarios del SCAI, principalmente en aspectos técnicos relacionados con el uso del sistema. Debe identificar y remitir aquellos incidentes que requieren de atención del Departamento de Desarrollo de Sistemas.
- **Sistema de Mesa de Ayuda:** Los incidentes se registran en un Help Desk que permite su asignación y seguimiento oportuno por parte del Departamento de Desarrollo de Sistemas.
- **Identificación y Gestión Técnica:** Una vez asignado el incidente al área técnica, se identifican las acciones necesarias para su resolución. Esto puede incluir actualizaciones en los diferentes ambientes, tanto de aplicación como de base de datos, para pruebas y liberación, si es necesario.

Este enfoque garantiza una respuesta eficiente y efectiva a las necesidades de mantenimiento y soporte de las aplicaciones, asegurando que los sistemas de la CGR operen de manera óptima y continua.

