DOCUMENTO ARQUITECTURA DE SOFTWARE

Agencia Nacional de Tierras

DIRECCIÓN DE GESTIÓN DEL ORDENAMIENTO SOCIAL DE LA PROPIEDAD

SUBDIRECCIÓN DE SISTEMAS DE INFORMACIÓN DE TIERRAS

TABLA DE CONTENIDO.

1 INTRODUCCIÓN 5
1.1. alcance 5
1.2. objetivo 5
1.3. definiciones 5
2 RESUMEN ARQUITECTÓNICO 7
2.1. ESTILOS ARQUITECTÓNICOS 8
2.1.1. Estilo basado en capas 8
2.1.2. Estilo MVC (Modelo Vista Controlador) 10
3 COMPONENTES SIGNIFICATIVOS DE LA ARQUITECTURA DEL SISTEMA 11
3.1 PRESENTACIÓN 12
3.2 COMPONENTES DE ALMACENAMIENTO Y EXTRACCIÓN DE DATOS 12
3.3 ESTÁNDARES 12
4 VISTAS 12
4.1 VISTA DE COMPONENTES 12
4.1.1. Sistema Orientado al Proceso 19
4.1.2. Consolidación de Insumos 21
4.1.3. Crear Análisis Predial Integrado - API 21
4.1.4. Gestión Análisis Predial Integrado - API 23
4.1.5. Validación 23
4.1.6. Gestión NUPRE (En Diseño) 23
4.1.7. SIT – Desconectado 23
4.1.8. Aseguramiento de la calidad 24
4.1.9. SIT – Solicitudes FISO 24
4.1.10. SIT – RESO – Valoración 24
4.1.11. SIT – Implementación POSPR 24
4.1.12. SIT – Procedimiento Único 25
4.1.13. Repositorio Operadores 25
4.1.14. ESB – ANT 25
4.1.15. Componente Geográfico 25
4.1.16. Seguridad 25
4.1.17. Servicios 26

4.1.18. Diagnostico Técnico Jurídico- DTJ 28

Diagrama de componentes diagnostico Técnico Jurídico (DTJ) 29

4.2 VISTA DE DESPLIEGUE SIT 30

4.3 VISTA LÓGICA. 32

5 INTEGRACIÓN 34

5.1 INTEGRACIÓN DE LOS COMPONENTES 34

6 ATRIBUTOS DE LA CALIDAD 34

6.1 ESENCIALES 35

6.2 ESPERADOS 35

6.3 DESEADOS 36

TABLAS DE ILUSTRACIONES.

Ilustración 1: Vista general de la Suite ArcGis (Esri) 6
Ilustración 2: Diagrama de contexto general del SIT 8
Ilustración 3: Capas del sistema de Información 9
Ilustración 4: Arquitectura de Servicios WebAPI 11
Ilustración 5: Diagrama de componentes (Comunicación de los componentes) 13
Ilustración 6: Imagen del ASP.Net MVC. 15
Ilustración 7: Arquitectura SIT Windows WPF 16
Ilustración 8: Arquitectura SIT con Cliente ArcGi 16
Ilustración 9: Barrido Predial Masivo. 18
Ilustración 10: Sistema Orientado al Proceso 20
Ilustración 11: Modelo de creación de la API. 22
Ilustración 12 Diagrama de Arquitectura por Capas OSPR 27
Ilustración 13 Diagrama de Componentes DTJ 29
Ilustración 14: Vista de despliegue 30
Ilustración 15 Vista Lógica Sistema Integrado de Tierras 33

1 INTRODUCCIÓN

La Agencia Nacional de Tierras – ANT es una entidad estatal de naturaleza especial, del sector descentralizado de la Rama Ejecutiva del Orden Nacional, con personería jurídica, patrimonio propio y autonomía administrativa, técnica y financiera adscrita al Ministerio de agricultura y Desarrollo Rural - MADR, creada bajo el Decreto 2363 del 7 de diciembre de 2015.

De acuerdo al artículo 3 del Decreto 2363 de 2015, determino como objeto: "(..) máxima autoridad de las tierras de la Nación, tendrá por objeto ejecutar la política de ordenamiento social de la propiedad rural formulada por el Ministerio de Agricultura y Desarrollo Rural, para lo cual deberá gestionar el acceso a la tierras como factor productivo, lograr la seguridad jurídica sobre esta, promover su uso en cumplimiento de la función social de la propiedad y administrar y disponer de los predios rurales de la propiedad de la Nación".

El artículo 18 del Decreto 2363 de 2015 asigna las funciones a la Subdirección de Sistemas de Información de Tierras que le permite soportar los procesos que hacen parte de la Gestión de Ordenamiento Social de la Propiedad, cabe destacar entre otras funciones, el diseñar e implementar las bases de datos y sistemas información la Agencia y garantizar su articulación e interoperabilidad con Sistema Nacional Gestión de Tierras, mantener actualizada la información que produzca Agencia que permita su disponibilidad, visualización, acceso y uso para la toma de decisiones, integrar evaluar y hacer seguimiento a los sistemas de información que implementa y administra la ANT, implementar y administrar el Registro Único de Usuarios de la Agencia Nacional de Tierras, diseñar, implementar y administrar el Sistema de información los Fondos Tierras.

La ANT requiere contar con una Arquitectura de Sistemas de Información para ayudar a orientar las funciones de la SSIT enfocadas hacia la transformación de la entidad en la búsqueda de una óptima gestión de los procesos, ofrecido los lineamientos que le permiten integrar y alinear la arquitectura de TI con la arquitectura misional.

1.1. ALCANCE

Este documento describe la Arquitectura de Software del Sistema de Integrado de Tierras - SIT haciendo uso de una serie de vistas de alto nivel que explican los diferentes componentes del sistema y sus relaciones, muestra las decisiones arquitectónicamente significativas que ayudaron a diseñar la estructura del SIT.

1.2. OBJETIVO

Ofrecer al interesado, la información del Sistema Integrado de Tierras desde distintos puntos de vista, de tal forma que identifique el estilo arquitectónico implementado, los componentes y módulos que lo conforman, sus relaciones y flujo de información y la infraestructura que lo soporta.

1.3. DEFINICIONES

EntityFramework[1]**:** es un conjunto de tecnologías de ADO.NET que permiten el desarrollo de aplicaciones de software orientadas a datos. Los arquitectos y programadores de aplicaciones orientadas

[1] https://msdn.microsoft.com/es-es/library/bb399567(v=vs.100).aspx

a datos se han enfrentado a la necesidad de lograr dos objetivos muy diferentes. Deben modelar las entidades, las relaciones y la lógica de los problemas empresariales que resuelven, y también deben trabajar con los motores de datos que se usan para almacenar y recuperar los datos. Los datos pueden abarcar varios sistemas de almacenamiento, cada uno con sus propios protocolos; incluso las aplicaciones que funcionan con un único sistema de almacenamiento deben equilibrar los requisitos del sistema de almacenamiento con respecto a los requisitos de escribir un código de aplicación eficaz y fácil de mantener.

Entity Framework permite a los programadores trabajar con datos en forma de objetos y propiedades específicos del dominio, por ejemplo, con clientes y direcciones, sin tener que pensar en las tablas de las bases de datos subyacentes y en las columnas en las que se almacenan estos datos. Con Entity Framework , los desarrolladores de software pueden trabajar en un nivel más alto de abstracción cuando tratan con datos, y puede crear y mantener aplicaciones orientadas a datos con menos código que en las aplicaciones tradicionales. Dado que Entity Framework es un componente de .NET Framework, las aplicaciones de Entity Framework se pueden ejecutar en cualquier equipo en el que esté instalado .NET Framework a partir de la versión 3.5 SP1.

ArcGis[2]: ArcGIS es un completo sistema que permite recopilar, organizar, administrar, analizar, compartir y distribuir información geográfica. Como la plataforma líder mundial para crear y utilizar sistemas de información geográfica (SIG), ArcGIS es utilizada por personas de todo el mundo para poner el conocimiento geográfico al servicio de los sectores del gobierno, la empresa, la ciencia, la educación y los medios. ArcGIS permite publicar la información geográfica para que esté accesible para cualquier usuario. El sistema está disponible en cualquier lugar a través de navegadores Web, dispositivos móviles como smartphones y equipos de escritorio.

Ilustración 1: Vista general de la Suite ArcGis (Esri)

[2] http://resources.arcgis.com/es/help/getting-started/articles/026n00000014000000.htm

API[3]**:**Una API (siglas de 'Application Programming Interface') es un conjunto de reglas (código) y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas: sirviendo de interfaz entre programas diferentes de la misma manera en que la interfaz de usuario facilita la interacción humano-software.

2 RESUMEN ARQUITECTÓNICO

El Sistema de Integrado de Tierras (SIT) es un sistema de información conformado por diferentes módulos algunos desarrollados internamente y otros heredados. El SIT está soportado por cuatro (4) dominios trasversales que apoyan los de negocio y la interfaz de presentación del sistema: el dominio de seguridad, interoperabilidad, servicios geográficos y gestión documental.

Para soportar los módulos que se vienen desarrollando, implementando e integrando al SIT bajo uno de los proyectos más importantes de la ANT, el Catastro Multipropósito, se amplían las capacidades de interoperabilidad para disponer y consumir servicios web, proporcionar segmentos de capas de presentación e información geográfica de los predios y territorios rurales, productos de los procesos de negocios junto con el visor geográfico.

Las nuevas necesidades de la ANT han hecho necesario trasladar algunas operaciones del SIT a la nube, para lo cual se implementó una estrategia de aprovisionamiento del trámite de recepción de información del sujeto de ordenamiento y su posterior tratamiento de datos para la validación, verificación e ingreso del Registro de Sujeto de Ordenamiento a las bases de la ANT, dicho proceso se encuentra distribuido en el módulo FISO como herramienta de captura de información en campo, posterior a este proceso y gracias a la madurez de los servicios de interoperabilidad de la ANT se procede a realizar las respectivas validaciones y verificaciones de la información diligenciada en el FISO con las fuentes de información de acuerdo con los convenios establecidos por la ANT con otras entidades como: DNP, IGAC y Registraduría Nacional; además de las validaciones que se realicen por la ANT.

En la siguiente ilustración se ejemplifica la arquitectura de alto nivel del Sistema Integrado de Tierras – SIT con los componentes base que soporta, las vistas de componentes a nivel de detalle se presentan más adelante.

[3] http://www.ticbeat.com/tecnologias/que-es-una-api-para-que-sirve/

Ilustración 2: Diagrama de contexto general del SIT

Cada uno de los componentes del sistema de información se comunica entre sí por medio de Servicios lo que le permite que los pilares del sistema se mantengan ya que se propende porque cada uno de los componentes presten los servicios de negocio por los que cada uno es construido.

2.1. ESTILOS ARQUITECTÓNICOS

Los estilos de arquitectura permiten mostrar la forma en la que el sistema de información está construido y se pueda contar con la estructura del sistema de tal forma que permite que tanto los desarrolladores y/o arquitectos puedan conocer la aplicación sin tener que adentrarse en el código.

2.1.1. Estilo basado en capas

El sistema de información utiliza este estilo de arquitectura para el desarrollo de los componentes que se construyen en funcionalidades específicas, conexión con datos externos y exposición de servicios, las principales capas que se pueden encontrar en la aplicación son: Datos, Entidades, Negocio, Servicios, Presentación, Seguridad y Auditoria, cada una con su responsabilidad especifica.

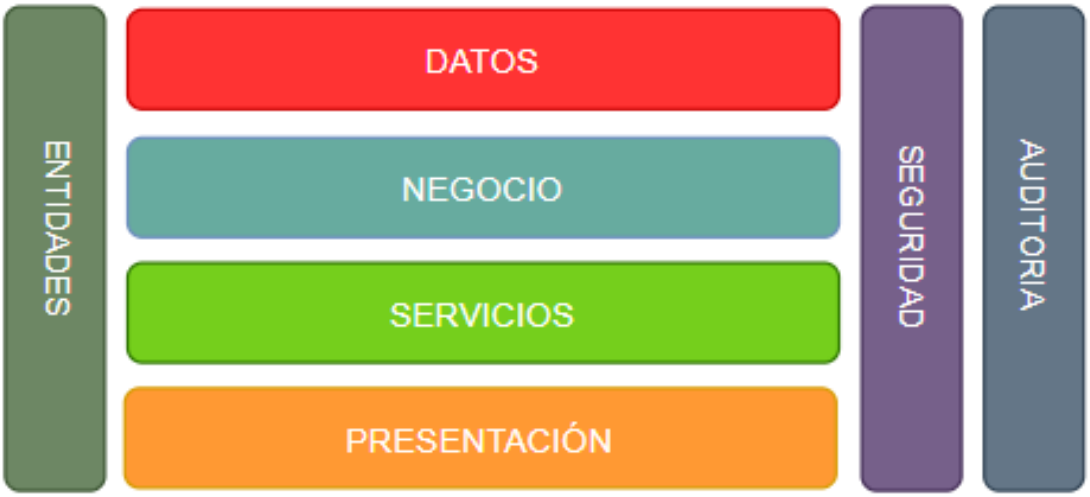

Ilustración 3: Capas del sistema de Información

En la ilustración se muestran las capas que contiene la aplicación, de acuerdo con los diferentes escenarios se pueden presentar todas o algunas no son necesarias.

Servicios Web o API para uso en Backend: En estos escenarios la capa de presentación no es obligatoria, dado que no se requiere que un usuario final dentro de la interfaz gráfica del sistema de información visualice o ingrese los parámetros para alguna operación.

En los escenarios donde se requiere que los componentes sean presentados por medio de una interfaz web, para estos escenarios la capa de presentación si está presente y por medio de esta se realiza el consumo de los servicios web de los componentes de negocio.

- Descripción de las capas:

Capa de Datos: Esta capa se encarga de realizar la lógica de conexión con fuentes externas de datos, como fuentes externas para el escenario del sistema de información se conoce como cualquier dato que no está en el entorno del sistema o en memoria de la aplicación, estos pueden ser (Archivos Planos, Archivos de Excel, Bases de Datos sin importar el motor, Servicios Web sean dentro de otro componente de la misma aplicación como de proveedores externos).

Para el acceso a las bases de datos en SQL Server se realiza haciendo uso de Entity Framework Database First, lo que implica que partimos del modelo de base de datos diseñado previamente para luego realizar la codificación.

Capa de Negocio: Esta capa contiene la lógica de negocio determinado para las operaciones que se están realizando, así como la comunicación con la capa de datos para la obtención de datos externos y procesamiento en el sistema de estos. El uso de esta capa se realiza a través de la capa de servicios.

Capa de Servicios: Esta capa contiene la implementación de los servicios web de la aplicación, esta se compone de la implementación de la interfaz del servicio y el cliente .Net para el consumo por parte de los demás componentes del sistema de información.

Entre las acciones que se realizan en esta capa es la validación de cada una de las operaciones contra el componente de seguridad y una vez finalizada la operación se registrará la auditoria.

Capa de Presentación: Esta capa se encarga de presentar al usuario final la accesibilidad para la interacción de este con los datos de la aplicación y las funciones de negocio, para esta capa el estilo principal utilizado es el Modelo Vista Controlador - MVC, que se veremos más en detalle en otra de las secciones que componen este documento.

Capa de entidades: En esta capa se colocan las clases que se utilizan para el trasporte de información entre las demás capas. Estas entidades deben utilizar el estándar establecido por MinTic para el intercambio de información, que en resumen indica que los nombres de las operaciones es en notación Pascal y los atributos de estos en Camel.

Capa de Seguridad: Esta es una capa trasversal utilizada para la gestión y el control de la seguridad de toda la aplicación llevada a nivel de las operaciones de los servicios, sobre la capa de presentación donde el un usuario puede realizar las acciones pertinentes.

Capa de Auditoria: La implementación de esta capa, permite registrar el uso de cualquier operación que se desee plasmar para una posterior revisión de auditoria en la cual se podrá gestionar tanto los parámetros de entrada como los resultados de cualquier operación del sistema.

2.1.2. Estilo MVC (Modelo Vista Controlador)

El estilo MVC se utiliza en el sistema de información para la representación y el desarrollo de la capa de presentación, dado que está capa utiliza el Framework de MVC de la plataforma .Net y este se compone del modelo, la vista y el controlador, estos existen en todos proyectos de Sitios Web de la aplicación y cada uno tiene su respectiva carpeta.

Como forma de desacoplar la aplicación, los modelos que son los encargados del manejo y acceso a los datos y la lógica del sistema se representan en cada uno de los servicios de los componentes que se exponen a través de servicios con las operaciones de la lógica de negocio del sistema de información. Por lo tanto, en la carpeta de los proyectos de Sitios Web (Models) se encontrará lo que para el proyecto serán los ViewModels los cuales se encargarán de la representación en donde sea necesario de las Vistas para la comunicación con los controladores.

Las demás vistas y controladores tendrán el comportamiento general de la implementación del framework.

Con el objetivo de dar soporte a nuevas implementaciones de la tecnología este componente MVC, a partir del año 2020 se utilizará para la generación de los nuevos servicios Web del sistema de información utilizando el framework de Web API.

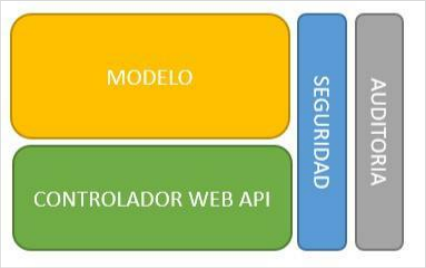

Ilustración 4: Arquitectura de Servicios WebAPI

Modelo: El modelo o las clases que soporta esta función se encargan de realizar la lógica de negocio necesaria de la aplicación, adicional presentan los modelos de transporte que viajaran en los Servicios.

Al realizar la lógica de negocio también se encarga de realizar la lógica de conexión con fuentes externas de datos, como fuentes externas para el escenario del sistema de información se conoce como cualquier dato que no está en el entorno del sistema o en memoria de la aplicación, estos pueden ser (Archivos Planos, Archivos de Excel, Bases de Datos sin importar el motor, Servicios Web sean dentro de otro componente de la misma aplicación como de proveedores externos).

Para el acceso a las bases de datos en SQL Server se realiza haciendo uso de Entity Framework Database First, lo que implica que partimos del modelo de base de datos diseñado previamente para luego realizar la codificación.

Las entidades de entity framework no se exponen directamente como modelos de transporte, debido que son creados desde la base de datos, por lo tanto, los modelos de transporte deben diseñarse de acuerdo con el diseño realizado en la interfaz de servicios Swagger.

Controlador Web API: El controlador Web API se encarga de encapsular los comportamientos de Seguridad y Auditoria por medio de Decoradores y este será capaz de exponer el API del servicio en RestFull utilizando los métodos (POST, PUT, GET y DELETE), adicional este controlador implementa las librerías de swagger para la generación de la documentación requerida.

Para las capas de ***Seguridad y Auditoria,*** en el apartado anterior se definió cada una de ellas por esta razón no se traen a colación en este ítem.

3 COMPONENTES SIGNIFICATIVOS DE LA ARQUITECTURA DEL SISTEMA

En la definición de los componentes arquitectónicamente significativos del sistema de información, se presentan tanto la implementación de la capa de presentación, componentes de la interfaz gráfica y los componentes de almacenamiento de datos, los cuales veremos a continuación:

3.1 PRESENTACIÓN

La capa de presentación es el componente que permitirá al usuario a interactuar con la aplicación, este tipo de interacción se puede realizar de forma conectada o desconectada dependiendo del tipo de interacción que el usuario requiere para el sistema de información.

3.2 COMPONENTES DE ALMACENAMIENTO Y EXTRACCIÓN DE DATOS

Los componentes de almacenamiento de datos son el Back End de la aplicación en estos componentes es donde se ven las capas más detalladas de la aplicación, tal como se describieron en el punto ***2.1 (Estilo basado en capas)***.

Estos componentes exponen servicios haciendo uso de RestFull para su consumo y utiliza una autenticación basada en token.

3.3 ESTÁNDARES

Se definieron los siguientes estándares (Lineamientos) para realizar la unificación de la codificación tanto para las bases de datos, como para las aplicaciones, con el fin de que el equipo o equipos involucrados en el desarrollo del sistema hagan uso de ellos y codifiquen siguiendo las pautas definidas.

- Estándares (lineamientos) de desarrollo.
  Estos estándares están definidos en el documento de Lineamientos para el desarrollo de productos de software de la ANT, para acceder a dicho documentos se puede hacer desde el siguiente enlace:
  https://agenciadetierras.sharepoint.com/sites/antintranet/sistema-integrado-de-gestion/procesos-misionales/Poltica%20%20Gestin%20de%20la%20informacin/GINFO-Pol%C3%ADtica-002-LINEAMIENTOS%20PARA%20DESARROLLO%20DE%20PRODUCTOS%20DE%20SOFTWARE%20EN%20LA%20ANT.pdf

- Estándares (lineamientos) de Bases de Datos.
  Estos estándares están definidos en el documento de Lineamientos y buenas prácticas de Bases de Datos, para acceder a dicho documentos se puede hacer desde el siguiente enlace:
  https://agenciadetierras.sharepoint.com/sites/antintranet/sistema-integrado-de-gestion/procesos-misionales/Poltica%20%20Gestin%20de%20la%20informacin/GINFO-Pol%C3%ADtica-003-LINEAMIENTOS%20Y%20BUENAS%20PRACTICAS%20DE%20BASE%20DE%20DATOS.pdf

4. VISTAS

4.1. VISTA DE COMPONENTES

La vista de componentes presenta una visión de alto nivel de los componentes (Módulos, aplicaciones o subsistemas) que forman parte del sistema de información y la forma en la que se realiza la comunicación entre los ellos.

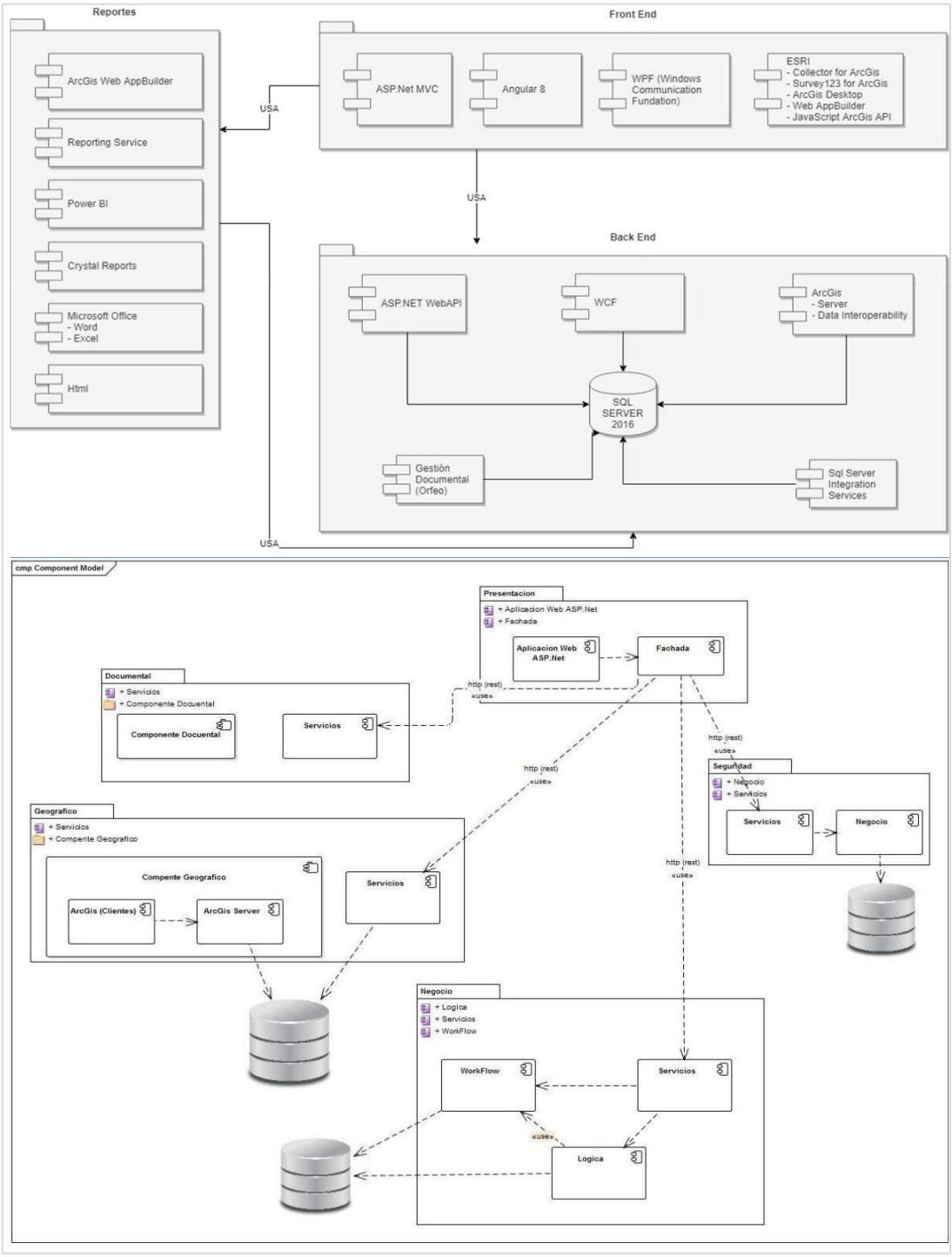

Ilustración 5: Diagrama de componentes (Comunicación de los componentes)

En el Diagrama de componentes que se muestra en la Ilustración 5, se presentan los componentes generales y la forman de comunicación con la capa de presentación por medio de la fachada.

Front End: El componente de presentación o Front End del sistema de información es el que se encarga de exponer al usuario final la posibilidad de navegar e interactuar con todos los procesos de negocio que el usuario requiera, esto por medio de los servicios que se expone por los componentes de negocio.

Este componente dentro del sistema de información puede tener dos variantes dependiendo de la necesidad del sistema Clientes Conectados y Desconectados.

Clientes Conectados: Este tipo de clientes son aplicaciones web construidas para estar constantemente conectadas al Backend de la aplicación, tienen la ventaja que el usuario podrá ver tan pronto ocurre un cambio en la información del sistema en su pantalla, al igual en caso de ingresar información a través de formularios web estos ingresaran inmediatamente a la base de datos de la aplicación.

Este tipo de clientes se construyen en el sistema de información con las siguientes tecnologías:

ASP.Net MVC: Este componente es una aplicación Web ASP.Net MVC la cual se desacopla por medio de los servicios en el que se exponen los componentes, y el consumo de estos servicios se centraliza en una fachada que cumple con el patrón de diseño fachada.

El componente comprende un proyecto de sitio Web principal, el cual es expuesto por medio de un Nuget SIT.Web y es implementado por los sitios Web de cada uno de los componentes de la aplicación que expongan una interfaz gráfica hacia el usuario.

A continuación, una descripción de la composición de los proyectos de capa de presentación:

- Models: En este directorio es donde se ubicarán las clases que estructurarán las Vistas del sistema de esta forma, estas solo contendrán propiedades que servirán para enviar información desde una vista al controlador y viceversa (ViewModels).

- Controllers: Los controladores se encargarán de ser el punto intermedio entre las vistas y los modelos que son los API Rest / Back End, por medio de la Fachadas que se exponen para cada uno de los componentes, además de recibir la información de las vistas por medio de los ViewModels.

- Vistas: Las vistas son los archivos cshtml que utilizan el motor de renderización Razor para modelar usando los ViewModels, enviar la información y recibirla de los controladores.

- Fachadas: Son el único punto de acceso de las Vistas, Controladores y demás clases que requieran acceso a las API Rest / Back End, de los componentes de la aplicación, los accesos a las fachadas se realizan mediante un Sigleton llamado “Sistema.Instance”, que construye una única instancia para el acceso a estas.

- Identity: Es el mecanismo encargado de controlar la sesión de usuario dentro de la instancia de la aplicación, haciendo uso de este mecanismo permite que este sitio Web sea el servidor de Autenticación dentro de la aplicación cumpliendo el protocolo oAuth 2.0.

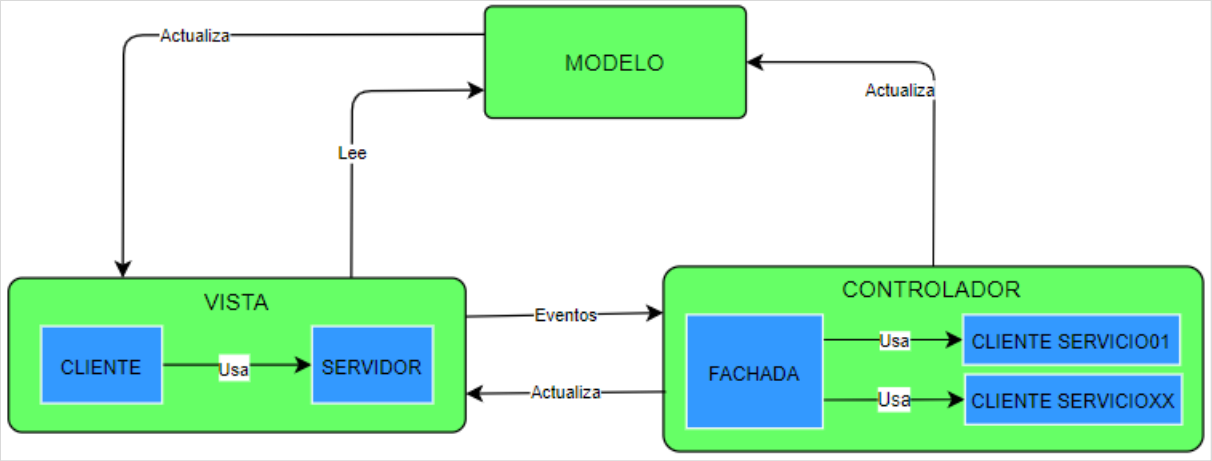

Ilustración 6: Imagen del ASP.Net MVC.

Angular 8: Algunos clientes no utilizarán ASP.Net MVC sino serán clientes cuyo procesamiento no tendrá comportamiento en servidor, estos clientes serán desarrollados utilizando el framework de Angular 8, el cual se encargará de consumir los servicios por medio de TypeScript implementando una fachada. Para este tipo de clientes es necesario tener en cuenta que tanto los servicios como los clientes deben tener registrado y habilitado de forma confiable las URL desde donde se realizaran las peticiones al backend dado que podrían detectarse como ataques de Cross-Site Scripting (XSS).

Clientes Desconectados: Este tipo de clientes pueden ejecutar sus funcionalidades de cara a los usuarios sin necesidad de estar conectados al BackEnd de la aplicación, por lo tanto, su comunicación con el BackEnd se realiza cuando el aplicativo encuentre conexión, para el sistema de información se trabajan los siguientes tipos de clientes:

WPF (Windows Presentation Fundation): Esta aplicación se construye utilizando el patrón de diseño MVVM el cual permite realizar la lógica de presentación de las interfaces de consulta, basados en las variables que se están construyendo, su mecanismo de sincronización utiliza los servicios Web expuestos por el BackEnd en el que el manejo de la bandeja de trabajo desconectada es el principal servicio que le permite su sincronización por medio de una interfaz la cual hereda cada uno de los formularios a construir en WPF el cual en uno de sus atributos se almacenaran todos los datos propios de cada formulario de captura en formato Json que persistirá en las bases de datos.

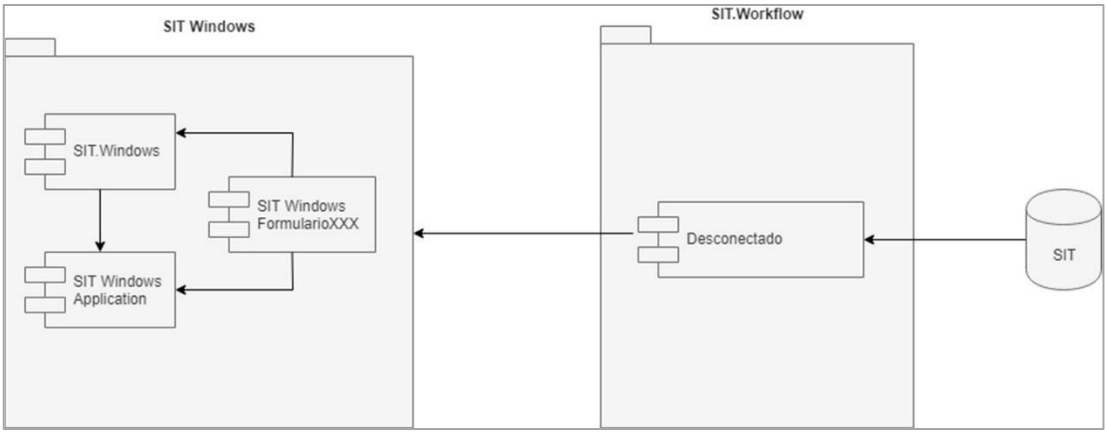

Ilustración 7: Arquitectura SIT Windows WPF

El BackEnd de sincronización hace uso en base de datos de las propiedades OPEN JSON de SQL Server 2016 por lo que hace indispensable que el uso del motor de base de datos sea este o superior.

Clientes ArcGis: Este tipo de clientes se utiliza cuando se desea que él o los formularios de captura estén disponibles en varias plataformas, la cantidad de preguntas sea variable con frecuencia, o requieran la captura de componentes geográficos, la construcción de esta se realiza de dos formas.

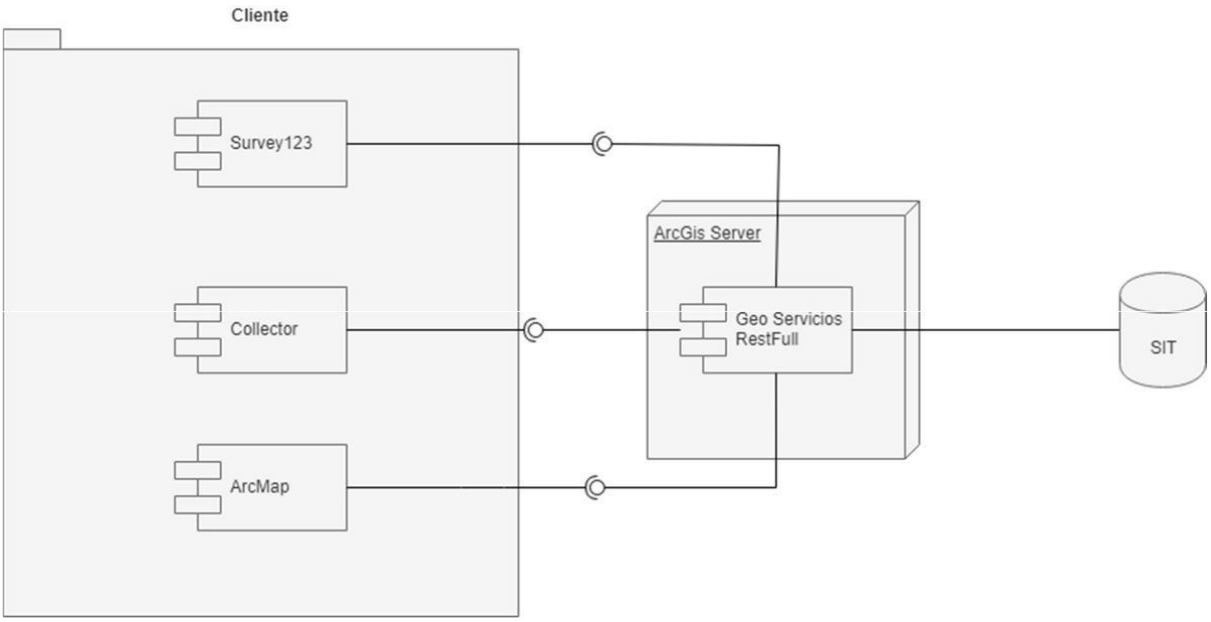

Ilustración 8: Arquitectura SIT con Cliente ArcGi

Construcción a partir de un modelo de base de datos: este proceso depende de que la base de datos este estructurada y dispuesta como una Enterprise Geodatabase ya que la comunicación y sincronización del cliente con el servidor es por medio de los servicios expuestos a través de ArcGis

Construcción a partir de un formulario: este proceso es en el cual por medio del diseñador de formularios en XSL Forms, se construye el formulario y a través de este se exponen los servicios que se almacenan en su propio esquema de base de datos, posterior a esto se realizan los procesos de transformación hacia el esquema de datos final deseado.

Barrido Predial Masivo → (OSPR) Ordenamiento Social de la Propiedad Rural.

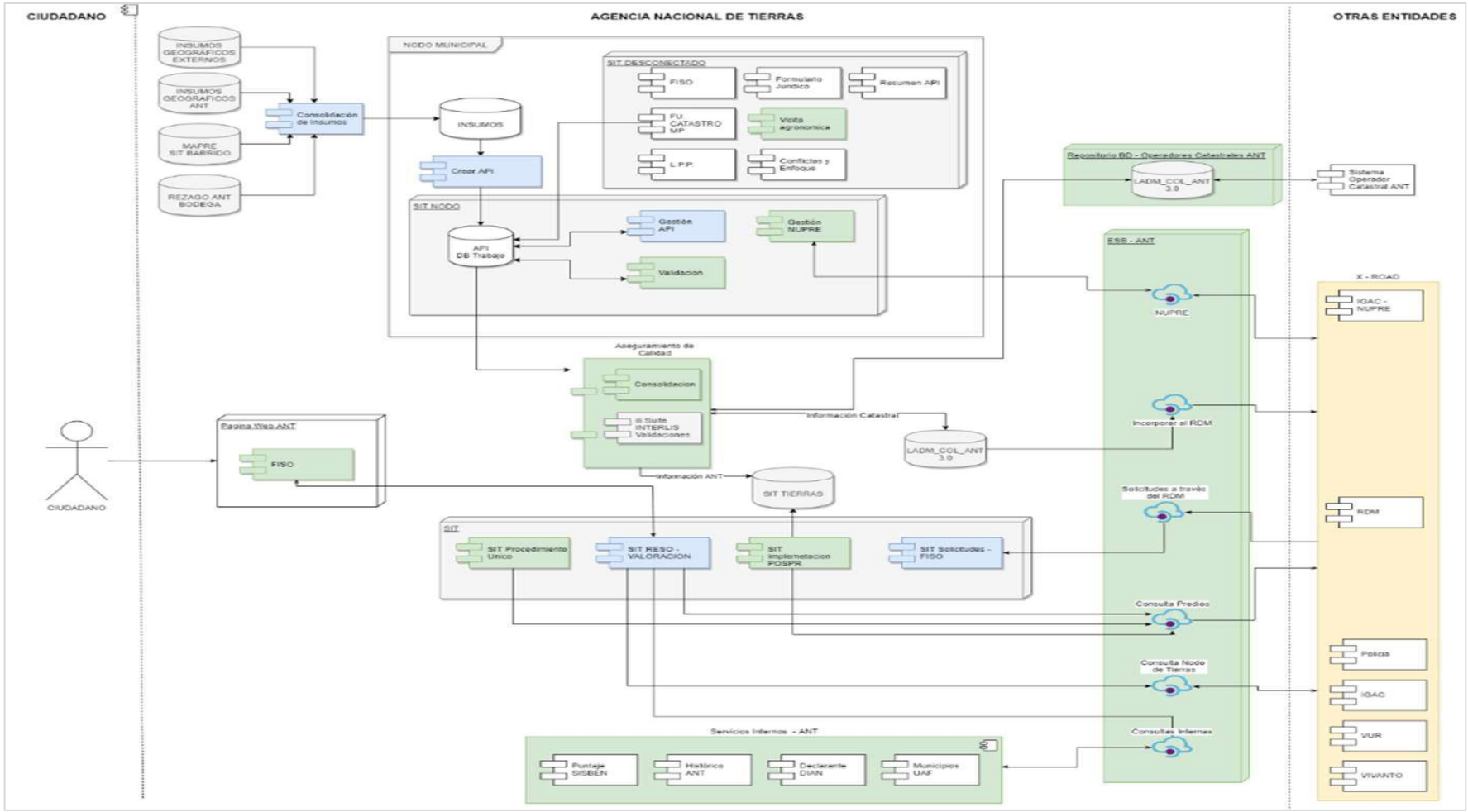

Ilustración 9: Barrido Predial Masivo.

4.1.1 Sistema Orientado al Proceso

A continuación, se presenta un diagrama con el proceso de OSPR apoyado desde el sistema de información en cada una de sus fases y componentes.

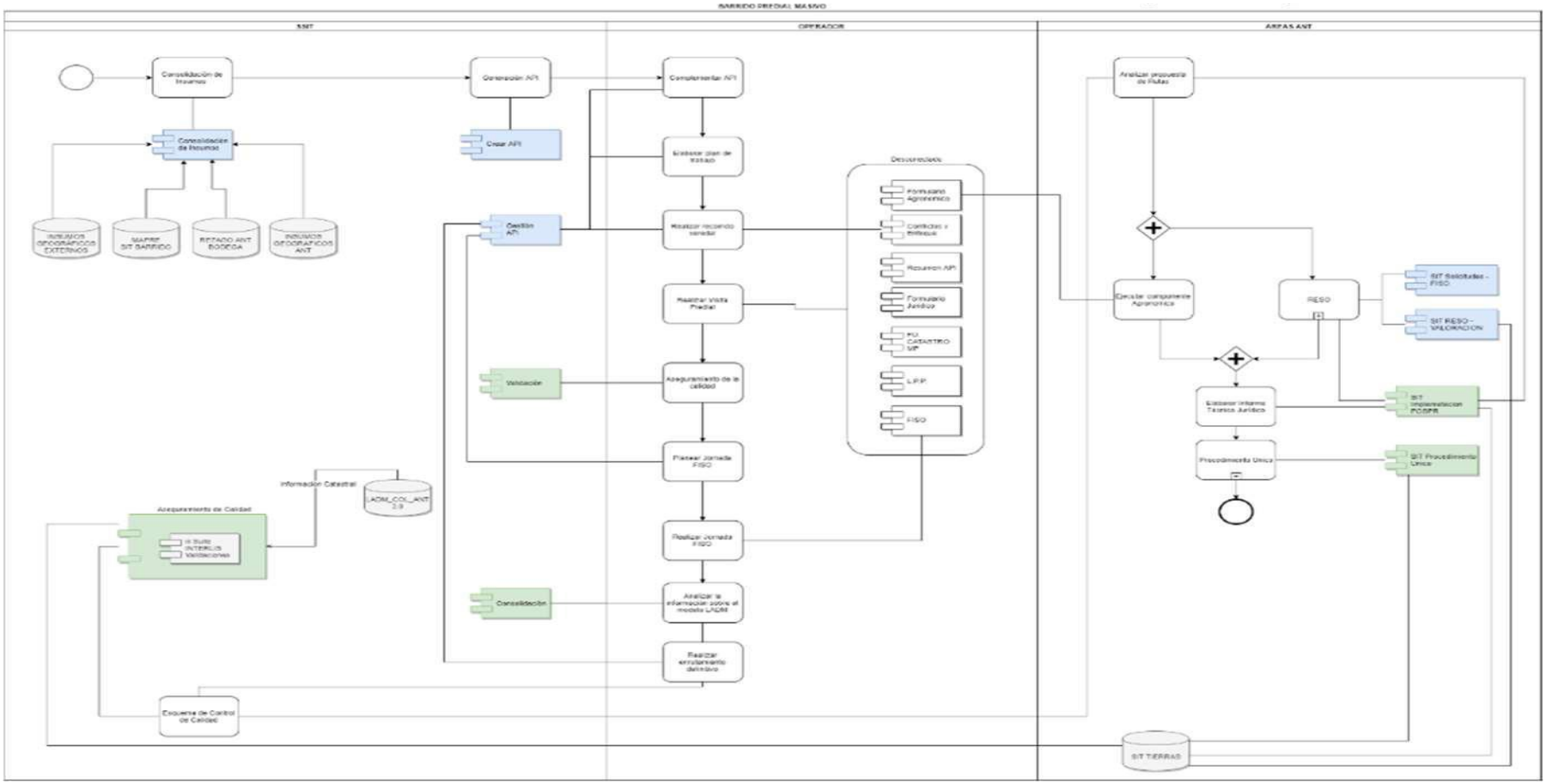

Ilustración 10: Sistema Orientado al Proceso

4.1.2 Consolidación de Insumos

Este es el componente al cual se le cargan las fuentes de información de los distintos insumos, por cada municipio, de las fuentes de información de la ANT, se encarga de realizar la extracción y cargue sobre la base de datos de INSUMOS a entregar al operador de la información correspondiente al municipio a intervenir.

La base de datos de INSUMOS es una Enterprese Geodatabase construida con la estructura de las fuentes origen filtradas por los municipios de intervención, es un componente desarrollado como servicio y ejecuta internamente las ETLs desarrolladas sobre la herramienta Data Interoperability de la Suite de ArcGis.

4.1.3 Crear Análisis Predial Integrado - API

Este es el componente del sistema de información que se encarga de extraer de los insumos de información cargados por municipio y consolida sobre el modelo de datos diseñado para el análisis predial integral.

Una vez consolidado sobre el modelo API, se ejecuta la lógica de negocio necesaria para realizar el enrutamiento o clasificación, sobre cada uno de los predios a intervenir, lo cual es insumo para realizar la planeación de las intervenciones.

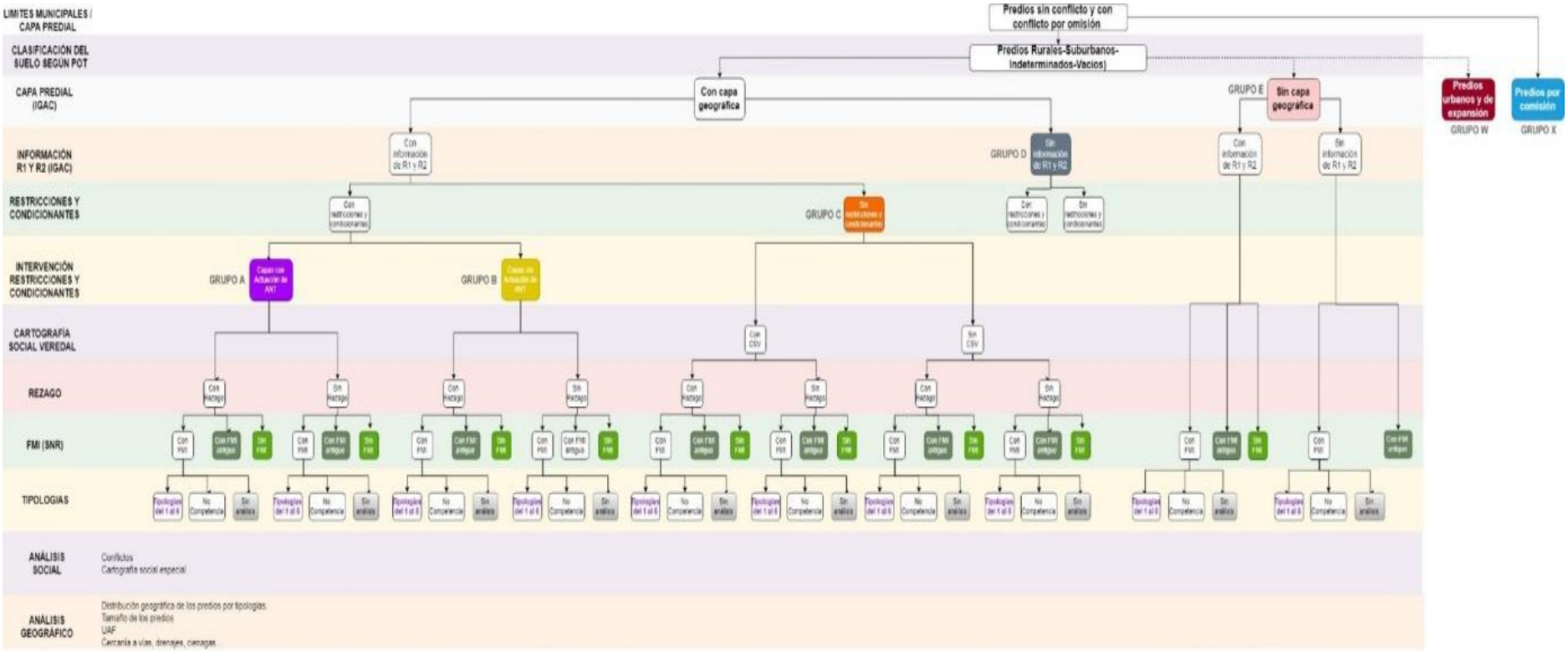

Ilustración 11: Modelo de creación de la API

4.1.4. Gestión Análisis Predial Integrado - API

Este componente se encarga de permitir los procesos de consulta, edición, asignación de la base de datos API, está compuesto de varias funcionalidades que le permiten al operador, socio estratégico y ANT realizar los procesos que hacen parte de las etapas de plan de trabajo, asignación de trabajo en campo y actualización del API.

Consulta: El sistema expone un visor geográfico construido utilizando WAB de ArcGis, en el cual se presentan las diferentes capas de información geográfica y alfanumérica dispuesta en la base de datos API, permite hacer filtrado de datos por las capas, y análisis de información capa a capa.

Consulta Avanzada y Edición: Cuando los procesos de consulta requieren un procesamiento más avanzado, realizar geoprocesamientos y cruces de capas avanzados, se dispusieron algunos servicios geográficos que permiten sobre cualquier herramienta cliente geográfica, para el propósito de los operadores de la ANT, se dispone de ArcGis Pro para esta tarea. Sobre estos mismos servicios se ofrece la capacidad de edición desconectada para realizar los procesos de actualización del API de acuerdo con las condiciones del territorio durante los recorridos veredales y prediales.

Asignación de Trabajo: Esta funcionalidad de la gestión de API permite al operador la asignación de carga de trabajo a los equipos de intervención, sobre esta funcionalidad se gestionan, unidades de intervención, cuadrillas y usuarios a trabajar, así como la asignación de las tareas sobre cada uno de los usuarios sobre los formularios de recolección de información en campo.

4.1.5. Validación

El componente de validación se presenta en varios momentos durante el flujo de información y es sobre este que el sistema permite al operador realizar la verificación de la calidad de la información recolectada por un equipo en un dispositivo desconectado, anterior a una integración y una consolidación de la información de la unidad de intervención. Se utiliza posterior a las visitas prediales de recolección de Formulario Único Catastral Multipropósito - FUCM, Conflictos y Levantamiento Planimétrico Predial - LPP y posterior a las jornadas FISO.

4.1.6. Gestión NUPRE (En Diseño)

Este componente se integra con las funcionalidades expuestas por el IGAC que proporcionara los números únicos prediales a los predios que se intervendrán durante el Barrido Predial Masivo.

4.1.7. SIT – Desconectado

Son los formularios desarrollados para la gestión de la operación en campo, se general los distintos formularios para recolección de información de:

- FUCM
- FISO
- LPP
- Conflicto y enfoques
- Formulario Jurídico

- Resumen API
- Visita agronómica

Estos formularios a excepción del LPP se desarrollan sobre la herramienta Survey 123 de ArcGis, el formulario de LPP se desarrolla sobre Collector For ArcGis.

4.1.8. Aseguramiento de la calidad

En este componente es donde el operador consolidará la información que será dispuesta a la ANT ya validada y estará listo para su consolidación y generación del modelo LADM nativo para interlist.

Luego se generarán la base de datos de conformidad total con el modelo LADM y permitirá a la ANT ejecutar las herramientas de validación de la integridad de la información con las herramientas de la suite de interlist (ili suite).

Una vez la calidad de la información sea conforme, este componente se encargará de realizar la incorporación de la información tanto al Repositorio de Datos Maestros RDM de catastro multipropósito, como generará las solicitudes para continuar con los procesos misionales de la ANT, incorporando la información a la base de datos misional del Sistema Integrado de Tierras SIT.

4.1.9. SIT – Solicitudes FISO

Es el componente del Sistema Integrado de Tierras que se encarga de gestionar las solicitudes que llegan a la ANT por concepto de FISO tanto para la demanda como la oferta, este es quien da inició a la gestión del proceso RESO.

4.1.10. SIT – RESO – Valoración

El RESO es el proceso que se encarga de evaluar la información de las personas que tienen intención de ser Sujetos de Ordenamiento por parte de la ANT, durante este proceso se realiza la valoración de los casos en la cual se realiza la consulta de diferentes fuentes de información externa a la ANT la cual se integrará utilizando la estrategia de ESB (Enterprise Service Bus) tanto para las funtes internas como externas a la ANT, entre ellas se encuentran las siguientes:

- Policía
- UARIV
- IGAC
- VUR
- AUF
- DIAN
- Histórico ANT
- SISBE

4.1.11. SIT – Implementación POSPR

Este es el proceso que se encarga dentro del Sistema Integrado de Tierras de orquestar toda la ruta de Ordenamiento Social de la Propiedad, es donde se puede determinar en qué estado se encuentra un

predio a lo largo de la RUTA. Integra las fases de los POSPR de Formulación, RESO y Procedimiento Único.

4.1.12. SIT – Procedimiento Único

Este es el proceso que permite a los equipos que atienden los procesos técnicos y jurídicos de la ANT, analizar la información de los Sujetos de Ordenamiento y permitir el trámite hacia un proceso de Adjudicación, Formalización o una pretensión agraria.

4.1.13. Repositorio Operadores

Se dispone de un repositorio de información no normalizado en donde los operadores que por alguna razón realizan barridos con propósito de Ordenamiento y no utilicen el sistema desarrollado por la ANT y esta base de datos sea consolidada y validada por el componente de Aseguramiento de Calidad anteriormente nombrado y continuar el proceso en la ANT.

4.1.14. ESB – ANT

Es la plataforma de interoperabilidad de la ANT que permite el consumo, exposición, orquestación de los servicios de la ANT, se implementa con el objetivo de consolidar una plataforma unificada de interoperabilidad para la entidad.

4.1.15. Componente Geográfico

Este componente del sistema es el encargado de interrelacionar con la plataforma ArcGis para el procesamiento y generación de productos cartográficos que permitan a la entidad tener la representación de los predios en una vista geográfica.

ArcGis en el sistema de información se utiliza para la generación de los siguientes componentes:

Geoprocesos: Estos se consumen desde la base de datos del Sistema Integrado de Tierras por medio de funciones y/o procedimientos almacenados que usar CLR (Common Language Runtime), directamente desde algunos de los servicios Web del Backend del sistema o desde el formulario Web que usa el API JavaScript de ArcGis.

Servicios WEB SIG: Son los servicios de edición o de capas que se utilizan para presentar y/o modificar capas geográficas desde una interfaz del sistema.

Visores o Formularios Geográficos de WAB: Estos formularios Web AppBuilder se construyen con el diseñador de ArcGis y se consumen desde la capa de presentación del SIT utilizando iFrames.

Visores o Formularios Geográficos Nativos hechos con API JavaScript ArcGis: Estos formularios son desarrollados directamente en la capa de presentación del sistema en un formulario HTML consumiendo los servicios web geográficos.

4.1.16. Seguridad

La seguridad de la aplicación hereda funcionalidades e interactúa con la fachada de los servicios de seguridad haciendo uso del Framework de Microsoft Identity para el manejo de la seguridad en capa de presentación.

Componente de Seguridad: El componente de seguridad se encarga de controlar y gestionar la seguridad de los componentes de negocio y la capa de presentación para el consumo de servicios, navegación y acceso a funcionalidades de negocio del sistema.

4.1.17. Servicios

Se hace énfasis en esta capa dado que es utilizada para la comunicación entre los componentes esta capa es desarrollada con tecnología Windows Comunication Fundation (WCF) y expone un End Point RestFull, para la exposición de servicios de aplicación para consumo de los componentes de la aplicación.

Sin embargo, WCF no será la única forma en la que se construirán los servicios Web, los servicios serán reemplazados por el ASP.Net WEB API, que es la nueva generación de servicios RestFull de Microsoft implementando la integración con Swagger con el fin de generar las interfaces de servicios documentadas.

Para el consumo de este servicio se crea una capa implícita en los servicios en la cual se desarrolla un cliente que permite un acceso a los servicios por referencia de dll para los casos en los que el componente sea desarrollado plataforma .Net y se comparte por medio de paquetes Nuget.

Para los servicios desarrollados en ASP.Net Web API, los clientes de servicios serán construidos utilizando los clientes generados por la interfaz de swagger expuesta y estos clientes serán utilizados por los clientes Web de diferentes tecnologías como JavaScript, asp.net core, asp. Net MVC.

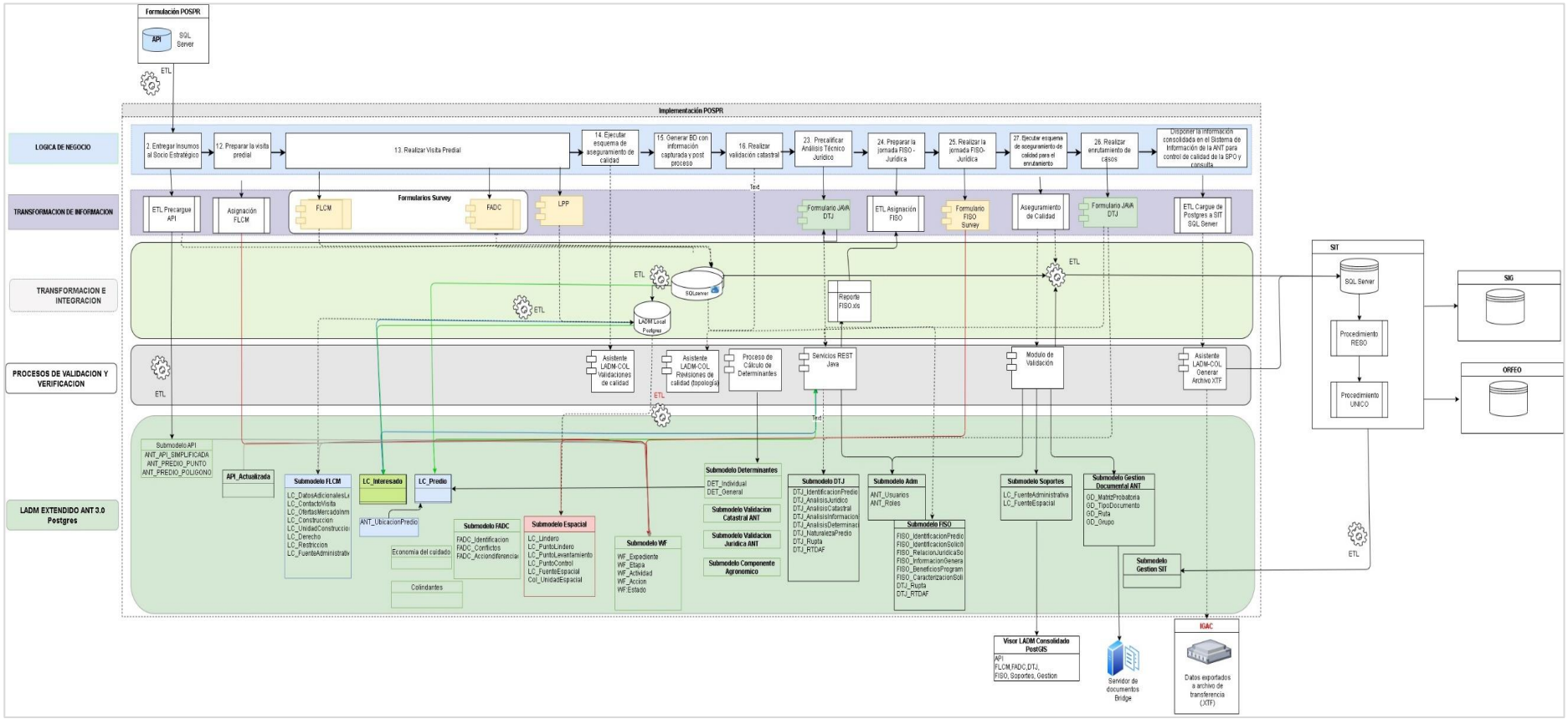

Ilustración 12 Diagrama de Arquitectura por Capas OSPR

4.1.18. Diagnostico Técnico Jurídico- DTJ

Con el fin de analizar la historia jurídica y catastral de los predios objeto de los barridos prediales, surge la necesidad de diseñar, desarrollar e implementar un Sistema de Información para el manejo del Diagnóstico Técnico Jurídico, que contenga la información base que permita realizar el análisis tanto jurídico como catastral de cada uno de los predios intervenidos en los diferentes municipios y de esta manera determinar cuáles hacen parte de Ordenamiento Social de la Propiedad Rural y cuales son objeto de formalización o actualización catastral.

Este sistema se integra con el modelo LADM_COL, sistema para la administración de la tierra y gestión de la información de catastro multipropósito en Colombia. Se implementa con la adopción de la norma ISO 19152 de 2012, modelo para el ámbito de administración del territorio (LADM - Land Administration Domain Model). Todo el modelo del Diagnóstico Técnico Jurídico del sistema está soportado en la estructura de LADM_COL.

El DTJ se integrará directamente con los desarrollos que se encuentran en el modelo LADM, se debe realizar una abstracción completa del modelo de base de datos y se deben mapear todas las clases que intervienen en dicha implementación. Adicionalmente al ser un desarrollo desacoplado, la información entre los niveles del MVC se comunica a través de solicitudes entre capas, en algunas de ellas se hace un consumo de servicios REST y en capas más cercanas a base de datos se realiza a través del ORM de Java. El consumo interno de servicios REST se implementa a través de los métodos HTTP (POST, PUT, GET Y DELETE). Toda la información constituida en el modelo de DTJ está disponible por base de datos y se puede implementar comunicación mediante servicios o en el mejor de los casos a través de una herramienta de mensajería como RabbitMQ.

Diagrama de componentes diagnostico Técnico Jurídico (DTJ)

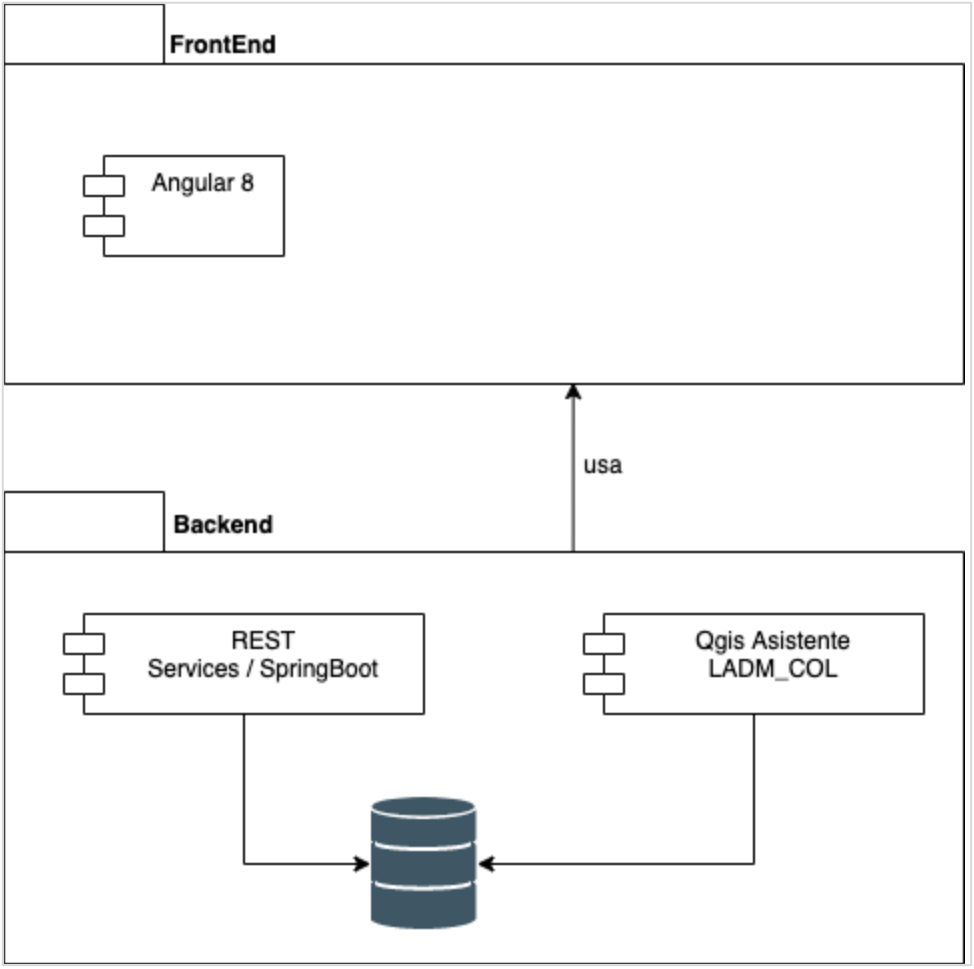

Ilustración 13 Diagrama de Componentes DTJ

El diagrama de componentes permite conocer cómo está construida la aplicación.

FrontEnd: se encarga de exponer a los socios, operadores y usuarios finales la forma de navegar e interactuar con todos los procesos de negocio que este requiere de acuerdo con la lógica de negocio implementada. Está construido en el framework de Google Angular, en su versión 8.

BackEnd: permite la interacción de la capa de presentación o FrontEnd con el modelo de base de datos Postgres, está implementado en SpringBoot, este framework agiliza el desarrollo en Java y presenta servicios de tipo REST que son consumidos por el FrontEnd, son usados a través de un token de seguridad. La comunicación entre componentes se realiza de una manera lineal desde la solicitud del usuario en la capa de presentación hasta la base de datos.

4.2 VISTA DE DESPLIEGUE SIT

La vista de despliegue representa desde un alto nivel la forma en la que el sistema se encuentra distribuido tecnológicamente a nivel de servidores y dispositivos.

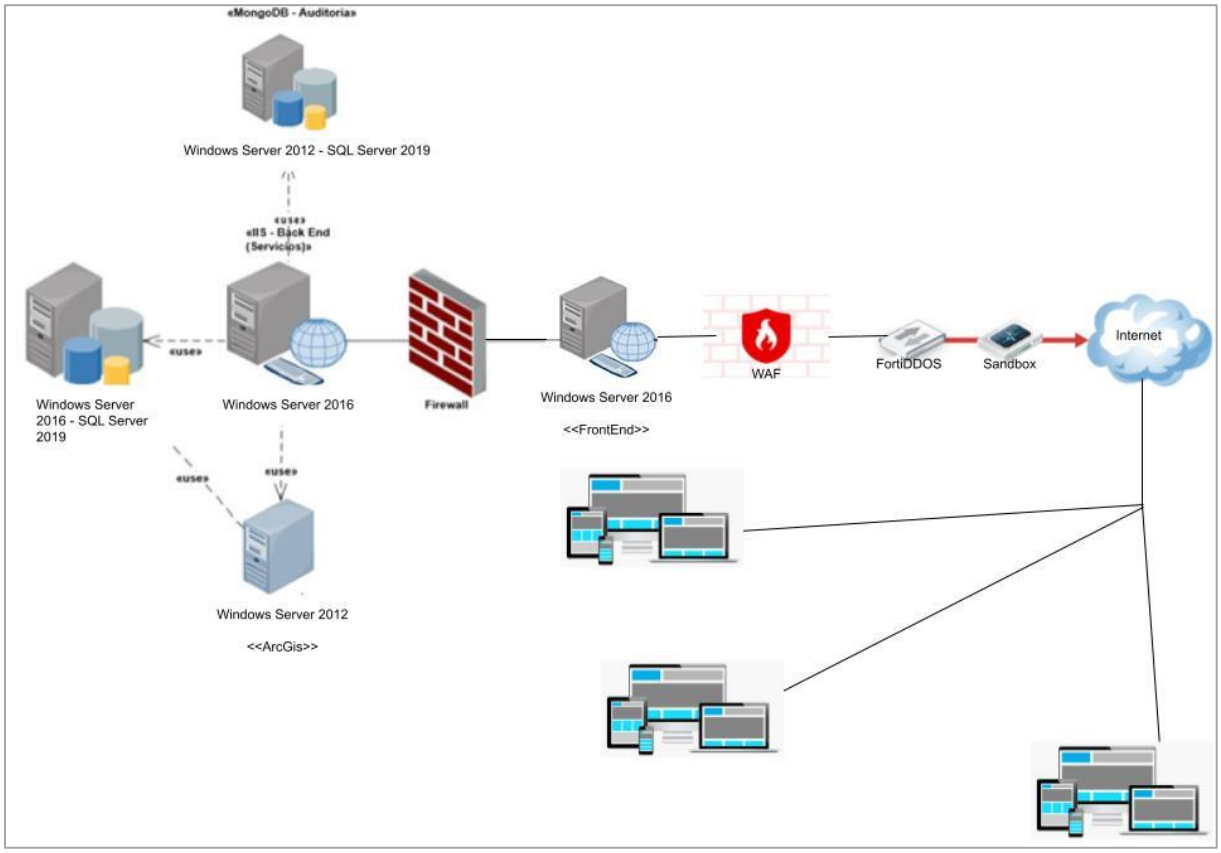

Ilustración 14: Vista de despliegue

En esta vista se presenta la distribución esperada de servidores, esto no significa que cada uno de estos servicios y servidores sea físicos o virtuales, esta decisión depende de la capacidad de infraestructura de la entidad y los componentes que se vallan implantando, sin embargo, en esta vista si se manifiesta las características mínimas para el funcionamiento del sistema. Características:

Servidor de Base de datos: Este servidor es el encargado de alojar el Servicio de Sql Server 2012 o superior, que es el motor de base de datos seleccionado para la construcción del sistema de información y almacenar la información de negocio del sistema.

Los siguiente son los requisitos mínimos para este servidor, esto con el fin de que la forma de escalar sea horizontal y se pueda disponer de más tamaño en disco duro, memoria RAM y procesador a medida de la disposición de nuevos componentes del sistema.

- Sistema Operativo: Windows Server 2016
- Versión del Motor: SQL Server 2012
- Memoria RAM: 8GB
- Disco duro: 500GB
- Procesador: Intel Xeon 2.67 Ghz

Servidor de Auditoria: Este servidor es el encargado de almacenar la auditoria de las transacciones del sistema de información, para este se seleccionó un motor en una base de datos NoSQL para almacenar las transacciones del sistema.

- Sistema Operativo: Windows Server 2016
- Versión del Motor: SQL Server 2012
- Memoria RAM: 8GB
- Disco duro: 500GB.
- Procesador: Intel Xeon 2.67 Ghz

Servidor de Aplicaciones (Back End): Este es el servidor que se encarga de servir de Hosting para los servicios en Back End de la aplicación, los cuales se desarrollan haciendo uso de la tecnología WCF y requieren para su funcionamiento un servicio de Internet Information Services (IIS).

Los siguiente son los requisitos mínimos para este servidor, esto con el fin de que la forma de escalar sea horizontal y se pueda disponer de más tamaño en disco duro, memoria RAM y procesador a medida de la disposición de nuevos componentes del sistema.

- Sistema Operativo: Windows Server 2012R2
- Versión del Motor: SQL Server 2012
- Memoria RAM: 8GB
- Disco duro: 500GB
- Procesador: Intel Xeon 2.67 Ghz

En los casos de que la capa de presentación del sistema de información se halla construido basado en el framework Angular 8, estos clientes serán hosteados directamente en este servidor dado que este tipo de clientes no requieren procesamiento en servidor y realizan los llamados directamente al backend de la aplicación desde el navegador de los usuarios.

Servidor de Aplicaciones (Front End): Este es el servidor que se encarga de servir de Hosting para los sitios Web que contienen la capa de presentación de la aplicación, este servidor debe permitir hospedar varios sitios simultáneamente dado que la arquitectura de la capa de presentación desacopla cada uno de los sitios web, con el fin de permitir que estos sean escalables y la mantenibilidad se vea favorecida.

Los siguiente son los requisitos mínimos para este servidor, esto con el fin de que la forma de escalar sea horizontal y se pueda disponer de más tamaño en disco duro, memoria RAM y procesador a medida de la disposición de nuevos componentes del sistema.

- Sistema Operativo: Windows Server 2008R2
- Versión del Motor: SQL Server 2012
- Memoria RAM: 8GB
- Disco duro: 500GB
- Procesador: Intel Xeon 2.67 Ghz

Firewall: Se espera que se disponga un firewall o algún mecanismo que permita que los servicios de Back End y Bases de datos, no sean expuestos directamente o se expongan de forma controlada y así el único servidor de cara a los usuarios sea el Servidor de Aplicaciones (Front End).

Clientes de Usuarios Finales: Los clientes de los usuarios finales serán clientes web que sea accederán a través de Navegadores, para el caso se diseña la aplicación con soporte para los siguientes navegadores.

- Microsoft Edge 38 o superior
- Chrome v55 o superior

4.3 VISTA LÓGICA.

El Sistema Integrado de Tierras - SIT - está compuesto por los siguientes módulos que dan cuenta de las competencias y alcance definido por las normas que condicionan el actuar de la Agencia.

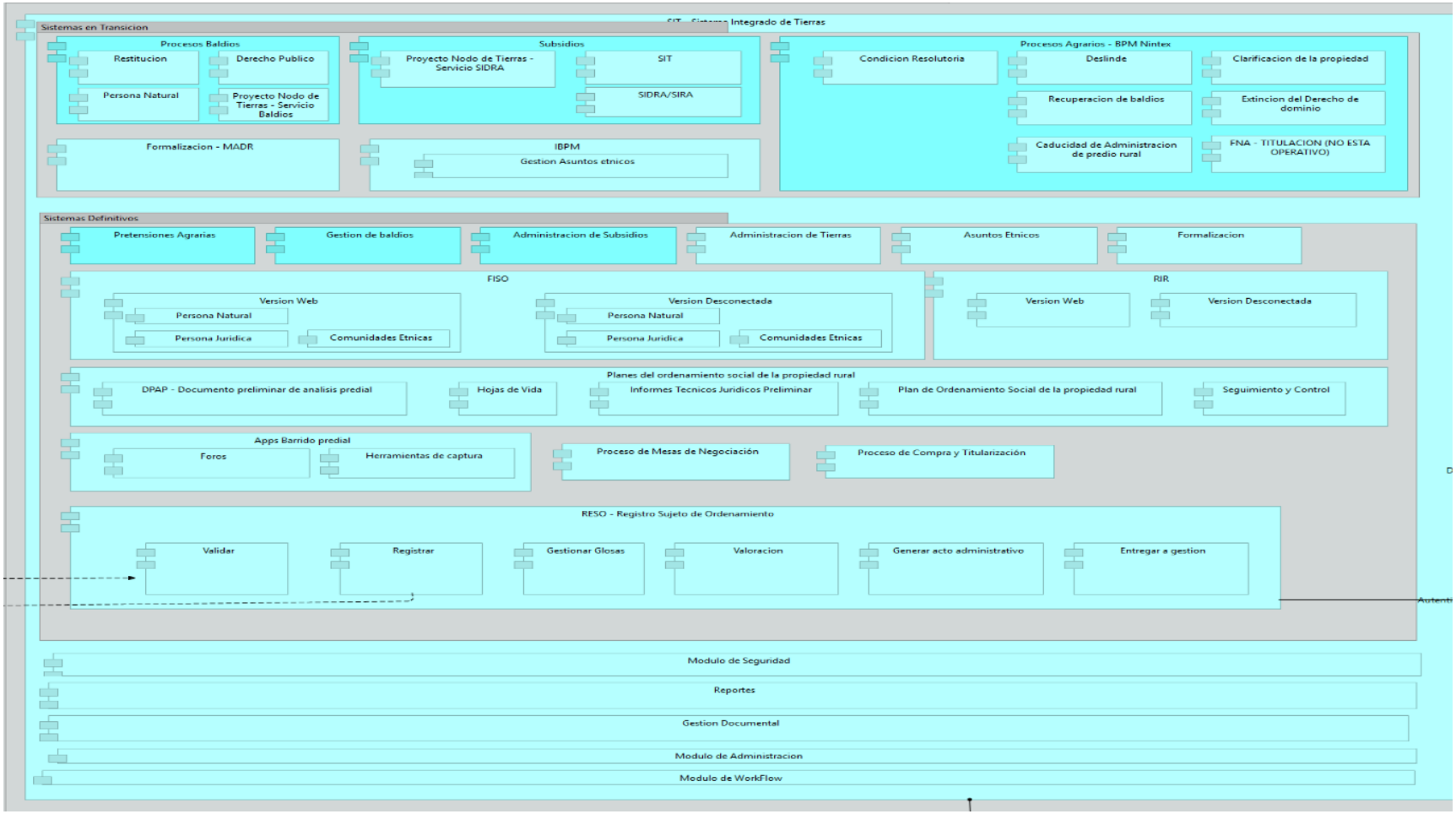

Ilustración 15 Vista Lógica Sistema Integrado de Tierras

5. INTEGRACIÓN

En este capítulo se espera hacer detalle en la forma en que el sistema de información está construido para interoperar entre los componentes y con terceros, entendiendo estos como necesidades de información de una entidad del sector o interesada.

5.1 INTEGRACIÓN DE LOS COMPONENTES

La integración que se busca en el sistema de información entre los componentes que lo conforman es a través de servicios Web que se desarrollaran cumpliendo con las especificaciones de RestFull, haciendo uso de la tecnología Windows Comunication Fundation (WCF).

Cada componente de integración que se existe en el sistema de información cuenta con las capas especificadas en el Titulo 2.1. donde se nombran las capas que tendrá cada componente del sistema de información.

La seguridad de los servicios se maneja haciendo uso de tokens, utilizando el protocolo de Autenticación y Autorización oAuth 2.0, el cual hace uso de este mecanismo para la Autenticación y el acceso a los recursos del sistema de información. En el siguiente enlace se puede detallar acerca de la implementación del protocolo utilizado https://oauth.net/getting-started/.

El consumo interno de los servicios se realiza a través de una biblioteca de clases creada para cada componente la cual se encarga de encapsular el consumo de los servicios web, este componente hace uso de HttpWebRequest para servir de cliente de los servicios web e implementa una interfaz de llamado para los métodos HTTP de Rest (POST, PUT, GET y DELETE), así mismo por medio de su constructor recibe el token y lo envía al servicio por medio de la cabecera del mensaje “Authorization”, para que cada operación del servicio valide los permisos de uso de este.

Los servicios construidos en WCF hacen uso de 3 comportamientos en su funcionamiento, el cual le permite: Validad autorización de uso de operaciones, Registrar auditoria y obtener las excepciones que se presenten en la ejecución de una operación.

Los servicios desarrollados en MVC utilizando el Framework de Web API, implementan la seguridad por medio del Cliente OWIN y este implementa la autenticación de oAuth 2.0 del servidor de autenticación y el registro de la auditoria en la base de datos MongoDB por medio de Decoradores en los controladores WEB API.

6. ATRIBUTOS DE LA CALIDAD

Los Requerimientos No Funcionales, como parte de los requerimientos generales de los Sistemas e Información, establecen características de calidad del producto de software relacionados con atributos

como eficiencia del desempeño, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad y portabilidad.

Los atributos de calidad que se presentan a continuación se definen de acuerdo con las necesidades del negocio y el estado actual de integración de este con los demás sistemas de información que son legados de la anterior institución (INCODRE).

6.1 ESENCIALES

Interoperabilidad: Una de las misiones de la Agencia Nacional de Tierras en el marco del catastro multipropósito y el barrido predial es el intercambio de información con otras entidades del sector y que hagan parte del catastro multipropósito, por lo que construir un sistema de información que el interoperar sea parte esencial de él es indispensable. Por lo que el sistema tiene una capa de servicios por cada componente de negocio del sistema en el cual se expone las funcionalidades necesarias del sistema, lo que permite que el interoperar para exponer o consumir servicios con otras entidades se puede hacer de forma nativa en él.

Mantenibilidad: Como uno de los requisitos del sistema de información es centralizar en un solo sistema de información los sistemas legados de la anterior institución y esto debido al difícil mantenimiento de diferentes sistemas de información en distintas plataformas, para lo que se requieren especialistas por cada sistema. Por esta razón cada uno de los componentes se construyó de tal forma que mantenga una misma arquitectura y plataforma para el mantenimiento, así que fácilmente un especialista en un componente del sistema pueda brindar soporte o desarrollar requerimientos sobre otro que no sea su especialidad.

Usabilidad: toda solución integrada en el SIT debe contemplar el diseño de interacción con el usuario, que permite definir el comportamiento y las características que debe tener la solución para ser fácil de usar y comprender. El análisis de usabilidad del sistema previo a su despliegue o paso a producción asegura el cumplimiento de los lineamientos establecidos en la política de Gobierno Digital con miras a ofrecer una adecuada experiencia de usuario y garantizar que la interacción con los usuarios, estén a la altura de sus necesidades y expectativas.

Integridad: Se desea que el grado de integridad de los datos se mantenga controlado por la criticidad de la información que maneja la entidad, por esto en el sistema de información se decidió que el acceso a los datos se realizará únicamente a través de la capa de servicios que este expone, y el control de esta está dado por cada operación y la autenticación por token de este.

6.2 ESPERADOS

Flexibilidad: Se espera que al modificar la aplicación que se encuentra en producción, por el agregar, modificar o corregir algún problema del sistema, no implique una caída de la totalidad del sistema y/o dejar sin operación a los usuarios por la corrección de los mismos. Para que el sistema tenga este tipo

de flexibilidad, cada uno de los componentes es independiente en su despliegue de los demás, teniendo como base algunas características trasversales como son (Seguridad y Auditoria).

Accesibilidad: este atributo de calidad en aplicaciones y formularios web es esencial en cumplimiento de la política de Ordenamiento Social de la Propiedad Rural en Colombia, entendiendo las características y cualidades de los ciudadanos, usuarios y grupos beneficiados o interesados, permite a la SSIT fortalecer la capacidad organizacional para atender a los ciudadanos o personas que tiene ciertas dificultades para usar herramientas tecnológicas, ya sea por su escaza educación digital y/o por alguna discapacidad física. Mejorar la accesibilidad de las herramientas digitales para atender las solicitudes y trámites de nuestros usuarios, permite avanzar en las acciones para atender las necesidades de la población rural en Colombia, en desarrollo del objeto de su misión.

6.3DESEADOS

Capacidad: el sistema debe operar dentro de los parámetros temporales definidos hasta con una cantidad de usuarios conectados concurrentemente definidos en los requerimientos de la solución de acuerdo con las necesidades de cada uno de los procesos que soporta, siendo la unidad de medida la cantidad de usuarios concurrentes o usuarios conectados o transacciones concurrentes u operaciones de BD concurrentes.

Compatibilidad: el SIT debe estar en capacidad de coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes sin detrimento, por medio tecnologías de contenedores o microservicios, es importante identificar en el servidor de aplicaciones que procesos están corriendo dentro de un sistema de contenedores y que los recursos de hardware asignados no entren en conflicto con los otros sistemas de información.

HISTORIAL DE CAMBIOS		
Fecha	**Versión**	**Descripción**
28/11/2022	01	Primera versión del documento.

Elaboró: María Argenis González Castellanos **Cargo:** Contratista Subdirección de Sistemas de Información de Tierras	**Revisó: Duberly Eduardo Murillo Barona**	**Aprobó: Jose Carlos Orozco Zequeda**
Firma: ORIGINAL FIRMADO **Elaboró: Carlos Eduardo Alfonso Pinilla** **Cargo:** Contratista Subdirección de Sistemas de Información de Tierras	**Cargo:** Subdirector de Sistemas de Información de Tierras	**Cargo**: Director de Gestión de Ordenamiento Social de la Propiedad (E)
Firma: ORIGINAL FIRMADO **Elaboró: Humberto Antonio Rosa Sarmiento** **Cargo:** Contratista Dirección de Gestión de Ordenamiento Social de la Propiedad **Firma:** ORIGINAL FIRMADO	**Firma:** ORIGINAL FIRMADO	**Firma:** ORIGINAL FIRMADO