



MINAGRICULTURA



Ministerio de Agricultura y Desarrollo Rural

## **ARQUITECTURA DE SISTEMAS DE INFORMACIÓN**

Agosto 18 de 2017

Bogotá, Colombia

## HISTORIAL DE VERSIONES

<b>Versión</b>	<b>Fecha</b>	<b>Descripción</b>	<b>Autor</b>	<b>Cargo</b>
1.0	22/08/2016	Versión inicial.	Eder Meneses	Arquitecto empresarial
1.1	18/08/2017	Actualización: <ul style="list-style-type: none"><li>• Lineamientos de Arquitectura</li><li>• Arquitectura General MinAgricultura.</li><li>• Catálogo de Sistemas de Información.</li><li>• Interoperabilidad MinAgricultura.</li></ul>	Carlos Garcia.	Arquitecto de Soluciones

## TABLA DE CONTENIDO

1. PROPÓSITO .....	5
2. ESTÁNDARES DE CODIFICACIÓN.....	6
3. LINEAMIENTOS DE ARQUITECTURA DE SISTEMAS DE INFORMACIÓN.....	7
3.1. Políticas generales .....	8
3.2. Lineamientos para aplicaciones en Microsoft Sharepoint.....	9
3.2.1. Audiencia.....	9
3.2.2. Estándares y mejores prácticas .....	10
3.2.3. Líneas de desarrollo en SharePoint.....	10
3.2.4. Apps Solutions.....	11
3.2.5. Soluciones y proyectos en Visual Studio .....	12
3.2.6. Características (Features).....	14
3.2.7. Manejo de la marca (Branding).....	14
3.2.8. Plantillas de Sitio .....	15
3.2.9. Despliegue de una Solución .....	15
3.2.10. Herramientas .....	18
3.2.11. Infraestructura Tecnológica.....	18
3.3. Aplicaciones desarrolladas en .NET Framework.....	19
3.3.1. Audiencia.....	19
3.3.2. Principios de arquitectura y diseño .....	19
3.3.3. Herramientas .....	20
4. VISTA GENERAL DE LOS SISTEMAS DE INFORMACIÓN.....	22
4.1. Catálogo de sistemas de información .....	22
5. DIAGRAMA DE INTEROPERABILIDAD.....	23
5.1. Catálogo de servicios de interoperabilidad .....	23
5.1.1. DANE .....	23
5.1.2. IDEAM.....	25
5.1.3. FINAGRO.....	25
5.1.4. ICA .....	25
5.1.5. MinAgricultura.....	25
5.1.6. URT .....	26

## TABLA DE ILUSTRACIONES

Ilustración 1 Lineamientos Auditoria y Trazabilidad MINTIC .....	7
Ilustración 2 Granja de Servidores Sharepoint .....	18
Ilustración 3 Vista de sistemas de información MinAgricultura .....	22
Ilustración 4 Diagrama de interoperabilidad .....	23

## **1. PROPÓSITO**

Documentar la arquitectura de los sistemas de información del Ministerio de agricultura y desarrollo rural. El documento incluye una vista general de los sistemas de información, con sus principales relaciones, el catálogo de sistemas de información, el diagrama de interoperabilidad con otras instituciones del estado colombiano y el catálogo de servicios del Ministerio.

## 2. ESTÁNDARES DE CODIFICACIÓN

En esta sección se describen los estándares de codificación adoptados en el ministerio de agricultura y desarrollo rural para aplicaciones desarrolladas en cualquier tipo de lenguaje orientado a objetos.

- Estándar de Nomenclatura: Uso del estándar CamelCase.
- Estándares de Diseño:
  - Uso de los Patrones Generales de Asignación de Responsabilidades (GRASP):
    - Creador: Permite identificar quien es el creador o instanciador de objetos en el código.
    - Controlador: Indica que la lógica de negocio debe estar separada de la capa de presentación, lo anterior debido a que permite una mejor reutilización y mantenimiento del código.
    - Alta cohesión: Indica que la información que contiene una clase debe ser coherente y relacionada con la clase.
    - Bajo acoplamiento: Indica el poder de la reutilización de código haciendo que las clases sean lo menos dependientes entre sí.
    - Principios básicos de la programación orientada a objetos (herencia, encapsulamiento, polimorfismo, sobrecarga).
- Estándares de Documentación (Comentarios de código fuente):
  - Clases: Debe contener mínimo la siguiente información en el encabezado.
    - Autor: Nombre completo del desarrollador que construyó la clase.
    - Fecha: Fecha de creación de la clase
    - Descripción: Objeto para lo cual fue creada la clase.
  - Métodos:
    - Autor: Desarrollador que construyó el método.
    - Fecha: Fecha de creación.
    - Funcionalidad: Breve descripción de lo que hace el método.
    - Parámetros: Listado de parámetros y breve descripción del uso de los parámetros en el método.
- Manejo de Excepciones: Uso de las sentencias Try – Catch en los casos requeridos por la funcionalidad en el código fuente.
- Programación a la defensiva: Programar pensando en que el usuario puede tomar caminos diferentes a los esperados en la funcionalidad implementada; por ejemplo, escritura de caracteres extraños en formularios, inyección SQL, cargue de archivos shell entre otros.

### 3. LINEAMIENTOS DE ARQUITECTURA DE SISTEMAS DE INFORMACIÓN

En esta sección se describen los lineamientos de arquitectura de sistemas de información adoptados por el Ministerio de Agricultura y Desarrollo Rural.

A continuación, se encuentra la descripción del lineamiento sobre auditoría y trazabilidad tomado de MINTIC por la oficina de Tecnología.

*“En el diseño de sus sistemas de información, la dirección de Tecnologías y Sistemas de la Información o quien haga sus veces debe tener en cuenta mecanismos que aseguren el registro histórico para poder mantener la trazabilidad de las acciones realizadas por los usuarios”*

En la guía de implementación de MINTIC se encuentran los siguientes artefactos relacionados al lineamiento sobre auditoría y trazabilidad:

<b>Atributo</b>	<b>Descripción</b>
Modelo de datos para auditoría y trazabilidad	Se debe diseñar e implementar un modelo de datos de auditoría y trazabilidad transversal a todos los sistemas de información.  Una guía de referencia para la definición del modelo de auditoría y trazabilidad se puede encontrar en el documento <a href="#">E.SI.01.Especificación Técnica-Sistemas de Información-Trazabilidad</a> .
Componente de auditoría y trazabilidad	Se debe diseñar e implementar un componente transversal para realizar la auditoría y trazabilidad en todas las aplicaciones. Este componente debe usar el modelo de auditoría y trazabilidad definido.
Estrategia de auditoría y trazabilidad basada en logs	Se debe diseñar una estrategia transversal basada en logs para realizar la auditoría y trazabilidad en todas las aplicaciones por medio de los logs generados.
Lista de chequeo de auditoría y trazabilidad	Lista de chequeo que deben cumplir los sistemas de información para cumplir con las estrategias y modelos de trazabilidad y auditoría, definidos.

<b>Atributo</b>	<b>Descripción</b>
Característica	Característica del sistema para cumplir con las estrategias y modelos de trazabilidad y auditoría, definidos.
Cumplimiento	Indica si el sistema cumple con la característica.

*Ilustración 1 Lineamientos Auditoría y Trazabilidad MINTIC*

Documentos de referencia auditoría y trazabilidad consultados.

- Guía del Dominio de Sistemas de Información. (Arquitectura TI COLOMBIA Marco de Referencia, 2014).
- Guía Técnica Trazabilidad. (Arquitectura TI COLOMBIA Marco de Referencia, 2014).

### 3.1. Políticas generales

A continuación, se encuentra el compendio de los principios generales para el desarrollo de sistemas de información, en ellos se propone una serie de directrices básicas acerca del comportamiento esperado en la construcción de aplicativos.

Sitio WEB	Los sitios web ya sean portales web y/o aplicativos informáticos deben cumplir con el Manual para la Implementación de la Estrategia Gobierno en Línea –Manual GEL última versión, de acuerdo con un plan de acción definido por cada sitio web.
Usabilidad y Accesibilidad	Los sistemas de información deben cumplir con el Manual de Usabilidad y Accesibilidad definido en MInAgricultura.
Arquitectura	<p>Los sistemas de información del MinAgricultura deben estar desarrollados en arquitecturas multicapas y/o multinivel, es decir; tres (3) o más capas (como la de presentación, aplicación o negocio y datos), que se puedan alojar en uno, dos o tres servidores, permitiendo la segmentación lógica, y la distribución Física de los componentes de la solución, disminuyendo el acoplamiento y mejorando la cohesión de los componentes para facilitar su mantenimiento.</p> <p>Arquitectura debe ser orientada a servicios, SOA definiendo la utilización de servicios para dar soporte a los requerimientos de software del usuario, estructuradas en líneas de modelo de procesos de negocio del MinAgricultura, facilidad de cambio de acuerdo a estándares abiertos como XML, SOAP, WSDL, REST.</p>
Documentación	La documentación mínima que debe contener un sistema de información debe estar alineada con la Metodología de referencia para el desarrollo de sistemas de información.
Tecnologías	<p>Los sistemas de información desarrollados para el Ministerio de Agricultura y Desarrollo Rural deben utilizar tecnologías Microsoft.</p> <p>Aplicaciones:</p> <ul style="list-style-type: none"> <li>• .NET Framework 4.0 o superior.</li> <li>• Lenguajes de programación:             <ul style="list-style-type: none"> <li>○ C#</li> <li>○ Visual Basic</li> </ul> </li> <li>• MVC</li> <li>• HTML5</li> <li>• CSS3</li> <li>• Bootstrap</li> </ul>

	<ul style="list-style-type: none"> <li>• Microsoft Sharepoint 2013 o superior</li> </ul> <p>Servicios WEB:</p> <ul style="list-style-type: none"> <li>○ WCF</li> <li>○ SOAP, REST</li> <li>○ Estándar GEL-XML</li> </ul> <p>Datos:</p> <ul style="list-style-type: none"> <li>• SQL Server 2012 o Superior</li> <li>• Reporting Services</li> <li>• Integration Services</li> <li>• Power BI</li> </ul>
--	---

### 3.2. Lineamientos para aplicaciones en Microsoft Sharepoint

Esta sección del documento es un lineamiento que describe los estándares y mejores prácticas para la implementación de portales en el Ministerio de Agricultura y Desarrollo Rural para aplicaciones desarrolladas sobre SharePoint.

Temas a tratar:

- Estándares y mejores prácticas
- Desarrollo en SharePoint 2013
- Control de código fuente
- Soluciones y proyectos en Visual Studio
- Características
- Manejo de la marca (Branding).
- Plantillas de Sitio
- Herramientas

Esta sección del documento no contempla:

- Instalaciones o configuraciones de la Plataforma o elementos.
- Instalaciones o configuraciones de soluciones externas como:
  - SharePoint Designer
  - Microsoft Visual Studio
- Instalaciones o configuraciones de herramientas de terceros como:
  - CAML Builder

#### 3.2.1. Audiencia

Su terminología está elaborada de una forma sencilla para el usuario final con conocimiento básico en la plataforma de SharePoint.

Está orientado para los responsables de soluciones y desarrollo, ingenieros de desarrollo de SharePoint.

- Estándares y mejores prácticas
- Desarrollo en SharePoint 2013
- Control de código fuente
- Soluciones
- Características

- Manejo de la marca (Branding).
- WebParts
- Plantillas de Sitio
- Aseguramiento de la calidad
- Herramientas
- Principios generales para el buen desarrollo de estándares y mejores prácticas

### 3.2.2. Estándares y mejores prácticas

Una de los grandes pilares de SharePoint es ser una plataforma para desarrollo, esta brinda diferentes opciones para hacer lo mismo de diferentes formas; sin embargo, se deben tener en cuenta algunos principios generales, que permite que la plataforma siga funcionando en condiciones óptimas.

A continuación, se describen los principios base.

- Toda funcionalidad nueva debe ser documentada
- No se deben editar los archivos base compilados SharePoint, estos solo se deben modificar a través de los fixes, updates o service pack proporcionados por Microsoft
- No se deben editar los archivos base de configuración de SharePoint tales como web.config, docicon.xml, resources.resx, de ser necesario alguna modificación se debe hacer a través de características u opciones propias de la plataforma para poder reversar los cambios realizados
- Nunca modifique el esquema de las bases de datos de SharePoint, para utilizar otros datos personalizados, cree sus propias bases de datos, preferiblemente en otra instancia o servidor de SQL Server.
- Nunca acceda directamente a las bases de datos de SharePoint
  - SharePoint provee de funcionalidades propias para el manejo de datos, que permite un buen rendimiento en el acceso a datos para muchos usuarios, como el manejo de cache entre otros, cualquier acceso a las bases de datos podría bloquear las tablas y por ende bloquear la plataforma
  - Utilice siempre los componentes propios de la plataforma para el acceso de datos, como las clases e instancia de objetos o los webservice disponibles
- Considere siempre la versión de SharePoint antes de empezar los desarrollos
  - Muchas actualizaciones arreglan errores propios de la plataforma, pero también puede introducir nuevos errores, documéntese bien de la versión de SharePoint en la que va a quedar el componente a desarrollar
- Utilice los componentes propios de la plataforma como base para nuevos desarrollos, como webparts, webcontrols, etc; esto optimiza tiempos de desarrollo y rendimiento en la granja

### 3.2.3. Líneas de desarrollo en SharePoint

SharePoint 2013 o superior ofrece 3 formas diferentes de crear soluciones en la plataforma:

- SandBox Solutions
- App Solutions
- Farm Solutions

Sin embargo, SandBox Solutions es un método obsoleto para la construcción de soluciones dadas las limitantes de este componente, se sigue manteniendo en SharePoint 2013 para dar soporte a las soluciones de SharePoint 2010

Para considerar si se debe crear una App de SharePoint o una Farm solution tenga en cuenta la forma en que se despliegan ambas soluciones, a continuación, se detalla lo que debe tener en cuenta al desarrollar una App Solution.

### 3.2.4. Apps Solutions

Una App para SharePoint es una aplicación pequeña e independiente, fácil de usar que soluciona una necesidad del usuario final, esta puede ser desde una lista básica de SharePoint con unos datos específicos hasta una funcionalidad completa con muchos elementos involucrados como un administrador de llamadas.

Las Apps se ejecuta en diferentes lugares, dependiendo de donde se encuentra alojada la aplicación, no se se ejecutan en el contexto de SharePoint, pero si en el contexto del navegador en la que se está visualizando o en una plataforma hospedada (ej.:Yammer)

Al implementar una App de SharePoint alojados, SharePoint crea un nuevo sitio web llamado app web. Este es un espacio seguro creado de forma dinámica para su aplicación, que le permite almacenar páginas, listas y bibliotecas, la app web provee un punto final aislado que puede ser llamada desde el lado del cliente mediante el uso de JavaScript.

Antes de poder llamar a las API de SharePoint desde su aplicación, es necesario autenticarse en SharePoint, teniendo en cuenta lo siguiente que el mecanismo de autenticación depende de donde se está ejecutando el código de su aplicación.

- En SharePoint se usa HTML y JavaScript y la autenticación se hace con el mecanismo de SharePoint.
- En la nube se tienen dos opciones, al lado del cliente por medio de una librería de Cross-Domain o al lado del servidor por medio de OAuth.
- Por último, puede hacer uso por medio de la API de REST.

El uso de las apps es limitado por su arquitectura y su contexto, una app no puede:

- Llamar componentes fuera de la colección de sitios donde se utiliza, es decir no puede llamar objetos que estén en la granja o la aplicación web donde es llamado
- No puede llamar a otra App
- No se pueden hacer plantillas de sitio personalizado
- No se utilizan para hacer paquetes de diseño o temas personalizados.
- No se utilizan para crear grupos de acción y acciones personalizadas
- No se utilizan para crear controles de usuario (archivos .ascx)

### 3.2.5. Soluciones y proyectos en Visual Studio

Una solución de VS es un conjunto de proyectos que son creados para resolver un problema o requerimiento para la organización, sin embargo, un proyecto de SharePoint creado en VS puede ser una solución de SharePoint, en síntesis, una solución en Visual Studio no tiene el mismo concepto de solución en SharePoint.

Para poder aclarar esto, tomaremos como ejemplo un portal que contenga los siguientes componentes:

- Diseño personalizado
- Página Inicial y paginas personalizadas
- Un sitio de noticias
- Un sitio para las wikis
- Una biblioteca de videos
- Un webpart para mostrar vínculos más utilizados

Cada uno de estos componentes serán personalizados ya sea a nivel de diseño o de estructura.

En VS se podría crear una solución con los siguientes proyectos

- Portal. MADR
  - Branding
  - Noticias
  - Wiki
  - Videos
  - Links

Sin embargo al publicar los diferentes proyectos para que sean desplegados en la granja, se crearían las siguientes soluciones de SharePoint:

- Branding.wsp
- Noticias.wsp
- Wiki.wsp
- Videos.wsp
- Links.wsp

Para el buen manejo de las soluciones y proyectos en VS se recomiendan las siguientes prácticas:

#### 1. Realice un diseño de componentes y estructura antes de crear la solución de VS (Análisis y Diseño).

Esto se refiere a abstraer todos los requerimientos del usuario y esquematizar los componentes que sean comunes o que se comporten como una única solución.

Las soluciones de SharePoint pueden ser tan simples como una biblioteca personalizada o tan complejas que se requiere el uso de muchos componentes.

Siguiendo el ejemplo del portal anterior, consideremos que el requerimiento de videos es tener un lugar para el almacenamiento de videos, esto implica los siguientes componentes:

- Tipo de contenido para el video
- Biblioteca de Videos

Sin embargo, el sitio de Wiki puede implicar los siguientes componentes:

- Tipo de contenido de Wiki
- PageLayouts
- Paginas
- CSS y JS
- Biblioteca de Paginas
- Lista de Categorías
- Webparts de Visualización interna
- Webpart de visualización externa (Home)

## 2. Soluciones independientes

Cada solución de SharePoint debe ser independiente, para esto debe considerarse:

- La solución debe poderse empaquetar y desplegar en un único WSP
- La solución debe poder ser funcional sin la dependencia de otras soluciones, o de ser necesario las soluciones dependientes deben ser desplegadas con esta.
- Debe poderse desplegar en cualquier tipo de Sitio de SharePoint
- Debe hacer uso de recursos propios.

## 3. Numero de Soluciones y paquetes

Se debe mantener acorde el número de soluciones que se deben desplegar y mantener un número aceptable para los despliegues.

En proyectos grandes se puede crecer considerablemente el número de soluciones de SharePoint, debido a que cada solución permite un fácil mantenimiento del código, sin embargo, esta especialización puede dificultar la instalación y el paso entre ambientes.

Si una solución en VS tiene muchos proyectos (>10), considere la creación de proyectos para el empaquetamiento de varios proyectos en uno solo, esto permite mantener el código y facilita la instalación.

En el ejemplo del portal, los sitios de Noticias y Wiki podría empaquetarse en una única solución llamada sitios.comunes, esta solución solo sería un espejo del paquete de cada una de las soluciones, los desarrolladores seguirían trabajando en los proyectos originales, pero al momento de hacer la publicación del paquete, se haría por medio de sitios.comunes, quedando una solución de SharePoint como sitios.comunes.wsp

El nuevo esquema en VS quedaría:

- Portal.MADR
  - Branding
  - Noticias
  - Wiki
  - Sitios.Comunes
  - Videos
  - Links

Sin embargo, al publicar las soluciones de SharePoint serían:

- Branding.wsp

- Sitios.Comunes.wsp
- Videos.wsp
- Links.wsp

### 3.2.6. Características (Features)

Las características en SharePoint se tienen diferentes tipos de alcances (scopes) que permite establecer en donde se despliega los elementos que estan en estas. Los alcances son:

- Farm (Granja)
- Web Application
- Site Collection
- Site

A continuación, se listan el uso que se debe dar a las características:

- Defina y utilice correctamente el alcance de las características y cree las que considere necesarias para el despliegue, como ejemplos:
  - A nivel de Site: paginas, instancias de listas, elementos de listas
  - A nivel de Site Collection: Diseños de página, tipos de contenido, flujos de trabajo
  - A nivel de Farm: Registro de categorías de logs.
- Defina las dependencias de la característica frente a otras características. Como ejemplo no se puede crear paginas si no existe la biblioteca de páginas que es provisionada por la característica de publicación nativa de SharePoint.
- Defina si la característica es oculta (no la debe desactivar el usuario) o si se debe mostrar en la lista de características.
- Asegúrese de eliminar los elementos creados por la característica cuando desactive esta.
- Asegúrese de que la característica tenga icono, titulo, descripción y versión de actualización

### 3.2.7. Manejo de la marca (Branding).

Branding comprende los siguientes elementos de SharePoint

- MasterPages
- PageLayouts
- CSS
- JS
- Imágenes
- Archivos de fuentes .spfont
- Archivos de colores .spcolor

A continuación se lista el buen uso para la creación y manejo del Branding

- Utilice la herramienta de Administrador de diseños para la creación de los elementos del branding
- Mantenga siempre el archivo .html de la master para poder crear el archivo .master

- Nunca modifique el archivo .master
- Analice si es necesario la creación de otros page layouts o si las nativas de SharePoint son suficientes
- Haga uso de los archivos de fuentes y de colores propios para SharePoint
- Haga uso de framework conocidos como JQuery.
- En lo posible haga uso de HTML5 y CSS3
- Utilice las imágenes en el tamaño que se van a mostrar
- Cree un proyecto en VS para el manejo del Branding

Para el manejo de dispositivos móviles, SharePoint ofrece dos alternativas:

- **Adaptative Design:** es nativa de SharePoint y permite definir los tipos de canales que se van a utilizar (iPhone, Android, Ipad, etc), esto implica la creación de una Masterpage por cada uno de los canales, hace uso del user agent para establecer el dispositivo
- **Responsive Design:** es el estándar universal para el manejo de dispositivos móviles, se hace por medio de CSS y JS, se hace por medio de la resolución de la pantalla, sin importar el dispositivo que se utilice.

Aunque se puede utilizar cualquiera de los dos, se recomienda que se use Responsive Design por su flexibilidad y estandarización.

### **3.2.8. Plantillas de Sitio**

SharePoint 2013 se puede crear las siguientes plantillas de sitios

- Site Definitivos
- Site Templates
- Web Templates

A continuación, se lista el buen uso para la creación de plantillas de Sitio

- Defina si la plantilla va a ser utilizada para la creación de un site collection o para un site
- Verifique que en el archivo de definición (onet.xml) se incluyan todas las características necesarias para el correcto funcionamiento del sitio, tanto a nivel de SiteCollection como a nivel de Site
- Verifique que las características personalizadas estén incluidas en el archivo de definición
- Verifique el orden correcto de las características, estas se activan de acuerdo al orden dado en el archivo de definición
- Utilice archivos de recursos propios para el manejo del lenguaje

### **3.2.9. Despliegue de una Solución**

Como se indicó anteriormente, una solución de SharePoint puede ser tan simple o compleja de acuerdo a las necesidades del usuario y puede involucrar la utilización de servicios propios de SharePoint, como lo son el User Profile Service, Excel Service, Search Service, entre otros servicios, como también puede involucrar componentes de

terceros o propios instalados anteriormente, lo que supone que todos estos elementos deben estar creados o disponibles antes del despliegue de la solución.

Para entender esto, demos como ejemplo una solución sencilla de un event receiver para una lista que envía correo cuando uno de sus elementos es eliminado. Aunque la solución es simple, hace la utilización del servicio de SMTP de SharePoint y por consiguiente se debe revisar que esta configuración esté disponible; además de esto supone que debe existir una lista, que debe estar en un sitio dentro de una colección de sitios, que a su vez está en una aplicación web, esto supone que todo esto debe estar creado antes de poder utilizar la funcionalidad.

Teniendo claro que el despliegue de una solución de SharePoint, así esta sea sencilla, puede comprender de varios pasos o validaciones que estas relacionados con la solución a instalar y que además una solución puede ser cualquier implementación dentro de SharePoint, es necesario que cada una cuenta con un documento de **Plan de despliegue** que indique como se debe desplegar la solución dentro de la granja, indicando todo lo necesario que sea requerido para que la solución funcione.

El documento de plan de despliegue debe comprender de los siguientes elementos:

- **Introducción:** Indicando la solución que se va a instalar, describiendo de manera genérica su objetivo dentro de la organización, el enunciado de los prerequisites y los nuevos componentes después de la instalación
- **Prerrequisitos:** Conjunto de tareas que permiten la validación de los componentes necesarios antes de la instalación de las soluciones
- **Actividad:** Conjunto de tareas que al ejecutarlas generan un objetivo específico, la actividad debe poderse validar para indicar si fue correctamente ejecutada.
- **Tarea:** Enumeración de pasos ordenados que se deben seguir en la plataforma.

A continuación, se muestra un ejemplo de cómo debe quedar una Actividad dentro del documento de despliegue.

Actividad: Instalación de soluciones

### **Agregue las soluciones a la granja**

1. Ingresar al servidor donde está instalado SharePoint con un usuario administrador de la granja
2. De clic en Inicio de Windows
3. De clic en Todos los programas
4. De clic en Microsoft SharePoint 2013 Products
5. De clic derecho en SharePoint 2013 Management Shell
6. De clic en Ejecutar como administrador
7. Ejecutamos los siguientes comandos ejecute línea a línea.
  - Add-SPSolution C:\Deploy\MADR.Branding.wsp
  - Add-SPSolution C:\Deploy\MADR.Custom.Tools.wsp
  - Add-SPSolution C:\Deploy\MADR.SPUtilities.wsp
8. Verifique que los comandos se hallan ejecutado correctamente, sin ningún mensaje de error
9. Cierre SharePoint 2013 Management Shell

### **Verifique las soluciones agregadas**

1. Ingrese a la Administración Central con un usuario administrador de la granja
2. De clic en Configuración del sistema
3. De clic en Administrar soluciones del conjunto de servidores, en la sección Administración del conjunto de servidores
4. Verifique que todas las soluciones cargadas anteriormente estén en la lista
  - MADR.Branding.wsp
  - MADR.Custom.Tools.wsp
  - MADR.SPUtilities.wsp

**Nota:** Si alguna instalación de las soluciones falla debe repetirse el procedimiento, si continua con falla comuníquese con el desarrollador de la solución.

El documento de despliegue debe contemplar todas las verificaciones y tareas necesarias para que la solución pueda ser utilizada satisfactoriamente, y para que el administrador de la plataforma pueda ejecutarlas sin necesidad de asistencia del equipo de desarrollo.

La creación del documento de despliegue debe ser responsabilidad de quien desarrolla la solución de SharePoint.

### 3.2.10. Herramientas

A continuación, se mencionan algunas herramientas de terceros útiles para un buen desarrollo:

- ULS Viewer: Utilizado para ver los archivos de Logs de SharePoint, permite filtrar por diferentes campos para una mejor búsqueda de los errores
- SharePoint Manager: Permite la exploración de las aplicaciones web con todos sus sitios, utilizada para poder encontrar las definiciones en xml de listas, conten types, etc.
- CAML Buider: Utilizado para crear código CAML para la consulta de ítems de una lista o biblioteca
- Oxygen XML Editor: Editor de XML para el manejo de XSLT y XML, utilizado para la creación de webpart que utilizan xslt.
- FireFox Bug: Utilizada para la depuración de CSS, HTML y Javascript al lado de cliente.

### 3.2.11. Infraestructura Tecnológica

El Ministerio de Agricultura y Desarrollo Rural cuenta con una granja de servidores para el ambiente de producción de Sharepoint organizada de la siguiente forma:

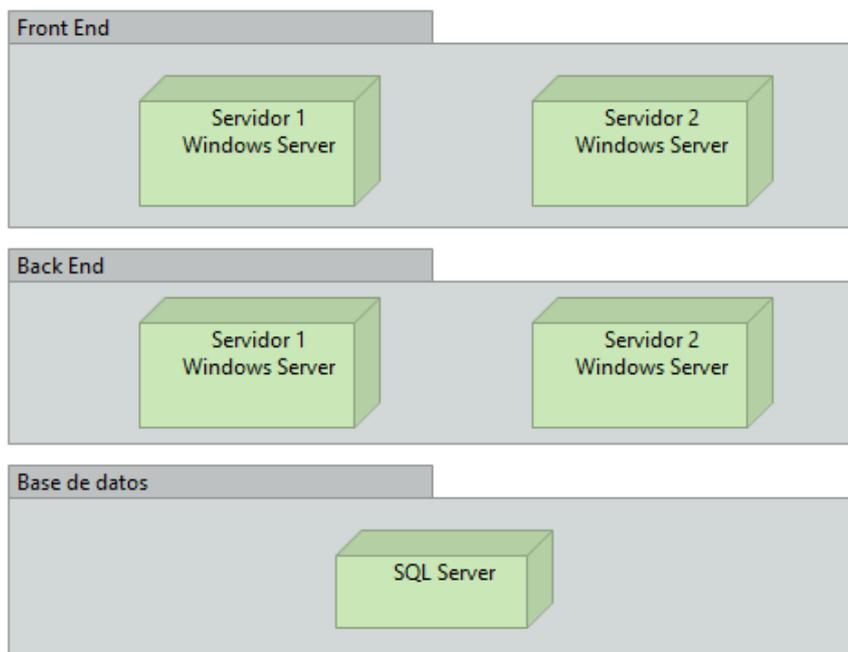


Ilustración 2 Granja de Servidores Sharepoint

### 3.3. Aplicaciones desarrolladas en .NET Framework

Esta sección del documento describe la arquitectura de referencia para las aplicaciones Web que se desarrollarán en el Ministerio de Agricultura y Desarrollo Rural. El centro de excelencia nace para articular la implementación de las necesidades identificadas en el ejercicio de arquitectura empresarial que se desarrolló en el ministerio durante 2014, donde surgió Productores 360 como una iniciativa fundamental que le dé al ministerio las herramientas que necesita para cumplir sus objetivos.

Productores 360 abarca aspectos de tecnología, procesos y con otras entidades. De los aspectos tecnológicos se identificaron un subconjunto de aplicaciones que se categorizan como aplicaciones Web y se refiere esencialmente a aplicaciones que se utilizarán a través de un navegador (*Browser*) y mediante una conexión a Internet. Este componente describe la arquitectura de referencia para esta clase de aplicaciones.

#### 3.3.1. Audiencia

Este documento está dirigido a:

- Arquitectos empresariales
- Arquitectos de soluciones
- Arquitectos de aplicaciones
- Desarrolladores

#### 3.3.2. Principios de arquitectura y diseño

- Arquitectura por capas

El estilo de arquitectura por capas separa la aplicación en diferentes capas de responsabilidad donde las capas inferiores desconocen las capas que las invocan. De esta forma una capa puede invocarse por cualquier otra capa y reutilizarse para distintos consumidores. Esta separación de responsabilidades promueve la mantenibilidad. En este diseño se utilizará un modelo clásico de capa de datos, capa de servicio y capa de presentación. Esta separación lógica no siempre es física; hacerlo a nivel físico promueve la escalabilidad, pero en el caso de aplicaciones Web frecuentemente agrega complejidad que hace cuestionable su conveniencia. Para el caso de las aplicaciones Web de Productores 360 no se prevé conveniente introducir separación física de capas y esto se refiere concretamente a no separar la aplicación de presentación de la aplicación de servicios.

- Separación de aspectos

Es la práctica de diseño mediante la cual cada aspecto está plenamente diferenciado y hay muy poco solapamiento entre funcionalidades. Esto se logra mediante modularidad, encapsulación y el principio de única responsabilidad. Con esta práctica se promueve la mantenibilidad y flexibilidad de la aplicación.

- Inversión de control

Es una forma de reducir el acoplamiento al no requerir objetos concretos en cada llamada sino interfaces de cómo es uno de esos objetos. Esta técnica además de reducir el acoplamiento permite fácilmente intercambiar componentes y aumenta la factibilidad de introducir pruebas unitarias. En la implementación se utilizará el inyector de dependencias provisto por el Framework que en tiempo de ejecución brindará los objetos concretos que se requieran.

- MVC – Modelo Vista Controlador

Este patrón divide la aplicación en tres áreas de responsabilidad:

- El modelo, dueño del modelo de negocio incluyendo sus datos y lógica
- La vista, que genera la interfaz gráfica y captura las entradas del usuario
- El controlador, acepta solicitudes del usuario, traduce las entradas del usuario a operaciones en el modelo y entrega datos a la vista para crear la interfaz de usuario

### **3.3.3. Herramientas**

- Estaciones de trabajo

Todo el desarrollo de estas aplicaciones se realizará utilizando Visual Studio en estaciones de trabajo de desarrollo o en máquinas virtuales de Windows Azure.

Cada estación de trabajo debe brindarle al desarrollador todo lo que necesita para crear y probar la aplicación de forma completamente aislada, sin depender de otros desarrolladores o de servicios compartidos.

Las estaciones de trabajo deben tener:

- Visual Studio 2013 Pro / Premium o Ultimate
- Acceso a internet
- SQL Server 2012 Developer Edition
- Últimas versiones de los navegadores Internet Explorer, Google Chrome y Mozilla Firefox
- Windows Azure SDK
- Dynamics CRM 2015 SDK
- Azure PowerShell Cmdlets
- Web Essentials (Plugin de Visual Studio)

- Repositorio, construcción y pruebas

El equipo de desarrollo requerirá además una serie de servicios compartidos:

- Herramientas de versionamiento; en el caso del Ministerio de Agricultura y Desarrollo Rural, Team Foundation Server (TFS).
- Servidor de compilación (Build Server) - TFS o VisualStudioOnline del ministerio
- Ambiente de pruebas – En este ambiente se desplegará de forma continua o al menos diaria el trabajo del equipo de desarrollo para correr pruebas de integración, pruebas automatizadas y pruebas manuales. Este ambiente estará en Windows Azure.

## 4. VISTA GENERAL DE LOS SISTEMAS DE INFORMACIÓN

A continuación se muestra el diagrama con los sistemas de información organizados de acuerdo a cuatro categorías (portales, misionales, direccionamiento y apoyo).

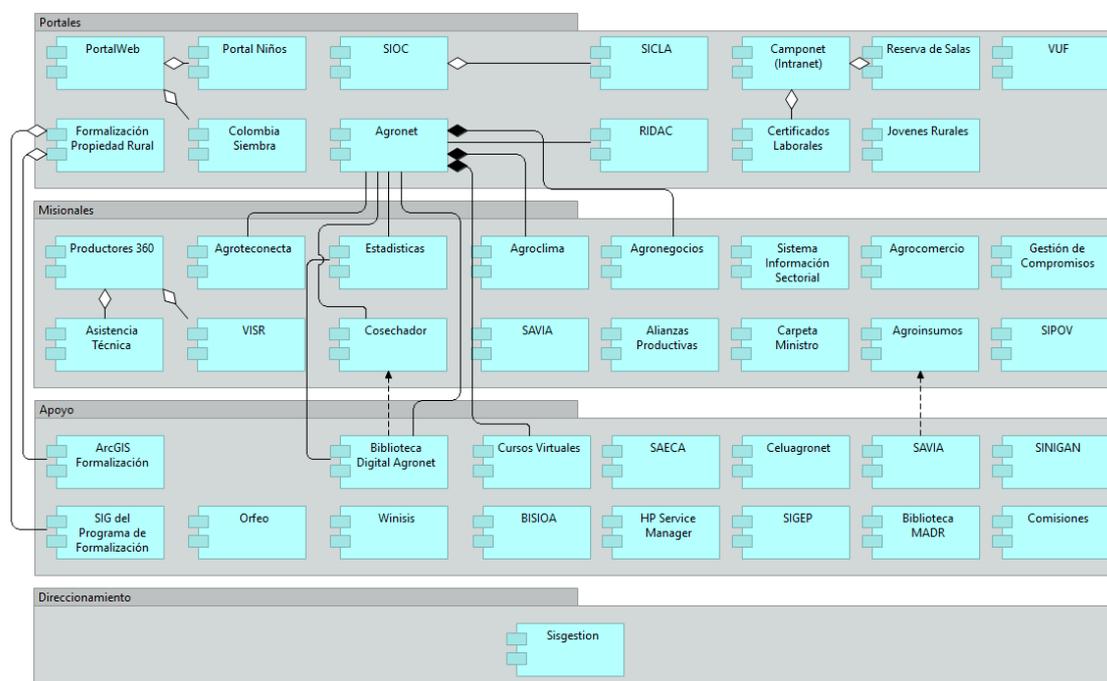


Ilustración 3 Vista de sistemas de información MinAgricultura

### 4.1. Catálogo de sistemas de información

Se consolida la información de los sistemas de información, teniendo en cuenta la siguiente estructura de campos:

- Nombre
- Descripción
- Categoría (Misional, Apoyo, Portales, Direccionamiento).
- Tipo (Web, Cliente servidor, Standalone, BD y scripts, Hoja de cálculo).
- Usuario funcional.
- Dependencia responsable.
- Soporte.
- Fecha de vencimiento del soporte.
- Estado (Desarrollo, Pruebas, Producción).
- Tipo de licenciamiento (Ilimitado, por usuario).
- Número de licencias.
- Servidor de aplicaciones producción.
- Base de datos producción.
- Servidor de aplicaciones pruebas.
- Base de datos pruebas.
- Tecnologías y/o productos de desarrollo.
- Requerimientos de servicios tecnológicos.
- Sitio.

Lo anterior se puede consultar en el documento: Catálogo de Sistemas de información.

## 5. DIAGRAMA DE INTEROPERABILIDAD

Esta vista permite identificar las diferentes entidades con las que el Ministerio de agricultura y desarrollo rural intercambia información, así como los sistemas de información entre los cuales se realiza dicho intercambio:

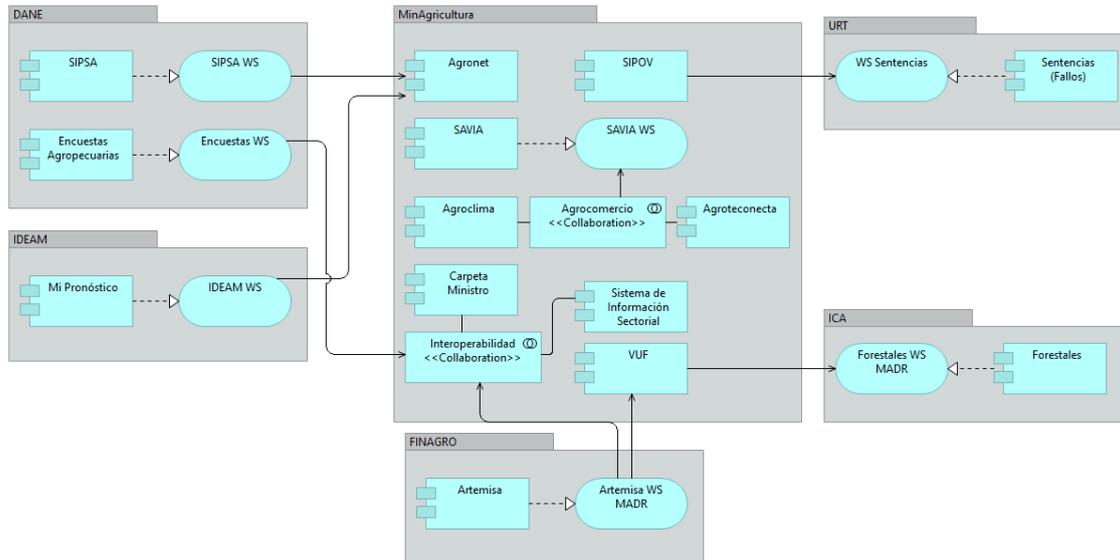


Ilustración 4 Diagrama de interoperabilidad

### 5.1. Catálogo de servicios de interoperabilidad

#### 5.1.1. DANE

Entidad proveedora	DANE
Sistema de información emisor	SIPSA
Servicio	Sipsa WS
Tipo de integración	WS - SOAP
URL	<a href="http://frmweb.dane.gov.co:8001/sipsaWS/SrvSipsaBeanService?WSDL">http://frmweb.dane.gov.co:8001/sipsaWS/SrvSipsaBeanService?WSDL</a>
Entidad receptora	MinAgricultura
Sistema de información receptor	Agronet

Entidad proveedora	DANE
Sistema de información emisor	Encuestas Agropecuarias
Servicio	DF_SAG_VGUN: Sacrificio de ganado vacuno, peso en pie y peso en canal por categorías según departamentos
Tipo de integración	WS – REST Protocolo SDMX
URL	<a href="http://apps.dane.gov.co/dane-jax-ws/rest/data/DF_SAG_VGUN">http://apps.dane.gov.co/dane-jax-ws/rest/data/DF_SAG_VGUN</a>
Entidad receptora	MinAgricultura
Sistema de información receptor	Staging para SI Sectorial y Carpeta Ministro

<b>Entidad proveedora</b>	<b>DANE</b>
Sistema de información emisor	Encuestas Agropecuarias
Servicio	DF_SAG_PORC: Sacrificio de ganado porcino, peso en pie y peso en canal por categorías según departamentos
Tipo de integración	WS – REST Protocolo SDMX
URL	<a href="http://apps.dane.gov.co/dane-jax-ws/rest/data/DF_SAG_PORC">http://apps.dane.gov.co/dane-jax-ws/rest/data/DF_SAG_PORC</a>
Entidad receptora	MinAgricultura
Sistema de información receptor	Staging para SI Sectorial y Carpeta Ministro

<b>Entidad proveedora</b>	<b>DANE</b>
Sistema de información emisor	Encuestas Agropecuarias
Servicio	DF_SAG_BUF Sacrificio de ganado bufalino, peso en pie y peso en canal por categorías. Total nacional
Tipo de integración	WS – REST Protocolo SDMX
URL	<a href="http://apps.dane.gov.co/dane-jax-ws/rest/data/DF_SAG_BUF">http://apps.dane.gov.co/dane-jax-ws/rest/data/DF_SAG_BUF</a>
Entidad receptora	MinAgricultura
Sistema de información receptor	Staging para SI Sectorial y Carpeta Ministro

<b>Entidad proveedora</b>	<b>DANE</b>
Sistema de información emisor	Encuestas Agropecuarias
Servicio	DF_SAG_OPR Sacrificio de ganado caprino, peso en pie y peso en canal por categorías. Total nacional
Tipo de integración	WS – REST Protocolo SDMX
URL	<a href="http://apps.dane.gov.co/dane-jax-ws/rest/data/DF_SAG_OPR">http://apps.dane.gov.co/dane-jax-ws/rest/data/DF_SAG_OPR</a>
Entidad receptora	MinAgricultura
Sistema de información receptor	Staging para SI Sectorial y Carpeta Ministro

<b>Entidad proveedora</b>	<b>DANE</b>
Sistema de información emisor	Encuestas Agropecuarias
Servicio	DF_SAG_OVN Sacrificio de ganado ovino, peso en pie y peso en canal por categorías. Total nacional
Tipo de integración	WS – REST Protocolo SDMX
URL	<a href="http://apps.dane.gov.co/dane-jax-ws/rest/data/DF_SAG_OVN">http://apps.dane.gov.co/dane-jax-ws/rest/data/DF_SAG_OVN</a>
Entidad receptora	MinAgricultura
Sistema de información receptor	Staging para SI Sectorial y Carpeta Ministro

### 5.1.2. IDEAM

Entidad proveedora	IDEAM
Sistema de información emisor	Mi Pronóstico
Servicio	IDEAM WS
Tipo de integración	WS
URL	<a href="http://mipronostico.ideam.gov.co:80/ideam-webservice/WebServicesIDEAM">http://mipronostico.ideam.gov.co:80/ideam-webservice/WebServicesIDEAM</a>
Entidad receptora	MinAgricultura
Sistema de información receptor	Agronet-Agroclima

### 5.1.3. FINAGRO

Entidad proveedora	Finagro
Sistema de información emisor	Artemisa
Servicio	Artemisa WS MADR
Tipo de integración	WS
URL	<a href="https://apl.finagro.com.co/ws_cif/Service1.asmx">https://apl.finagro.com.co/ws_cif/Service1.asmx</a>
Entidad receptora	MinAgricultura
Sistema de información receptor	VUF

### 5.1.4. ICA

Entidad proveedora	ICA
Sistema de información emisor	Forestales
Servicio	Forestales WS MADR
Tipo de integración	WS - REST
URL	<a href="http://misional.ica.gov.co/ws_forestales/">http://misional.ica.gov.co/ws_forestales/</a>
Entidad receptora	MinAgricultura
Sistema de información receptor	VUF

### 5.1.5. MinAgricultura

Entidad proveedora	MinAgricultura
Sistema de información emisor	SAVIA
Servicio	SAVIA WS MADR
Tipo de integración	WS - REST
URL	<a href="http://serviciosmadr.minagricultura.gov.co/SaviaWebApi/api/agro/test">http://serviciosmadr.minagricultura.gov.co/SaviaWebApi/api/agro/test</a>
Entidad receptora	MinAgricultura
Sistema de información receptor	Aplicaciones móviles: Agronegocios, Agroinsumos, Agroclima y Agroteconecta.

### 5.1.6. URT

<b>Entidad proveedora</b>	<b>URT</b>
Sistema de información emisor	SENTENCIAS (FALLOS)
Servicio	SERVICIO CONSULTA DE SENTENCIA
Tipo de integración	WS - REST
URL	http://172.30.46.3/wsURT/madrordenservice/getOrdenesByIdSentencia
Entidad receptora	MinAgricultura
Sistema de información receptor	SIPOV

<b>Entidad proveedora</b>	<b>URT</b>
Sistema de información emisor	SENTENCIAS (FALLOS)
Servicio	SERVICIO CONSULTA DE SENTENCIA
Tipo de integración	WS - REST
URL	http://172.30.46.3/wsURT/madrordenservice/getOrdenByIdOrden
Entidad receptora	MinAgricultura
Sistema de información receptor	SIPOV

<b>Entidad proveedora</b>	<b>URT</b>
Sistema de información emisor	SENTENCIAS (FALLOS)
Servicio	SERVICIO CONSULTA DE SENTENCIA
Tipo de integración	WS - REST
URL	http://172.30.46.3/wsURT/madrpersonaservice/getPersonaByDocumento
Entidad receptora	MinAgricultura
Sistema de información receptor	SIPOV

<b>Entidad proveedora</b>	<b>URT</b>
Sistema de información emisor	SENTENCIAS (FALLOS)
Servicio	SERVICIO CONSULTA DE SENTENCIA
Tipo de integración	WS - REST
URL	http://172.30.46.3/wsURT/madrpredioservice/getPredioByIdSentencia
Entidad receptora	MinAgricultura
Sistema de información receptor	SIPOV

<b>Entidad proveedora</b>	<b>URT</b>
Sistema de información emisor	SENTENCIAS (FALLOS)
Servicio	SERVICIO CONSULTA DE SENTENCIA
Tipo de integración	WS - REST
URL	http://172.30.46.3/wsURT/madrpredioservice/getPredioByCedulaCatastral
Entidad receptora	MinAgricultura
Sistema de información receptor	SIPOV

<b>Entidad proveedora</b>	<b>URT</b>
Sistema de información emisor	SENTENCIAS (FALLOS)
Servicio	SERVICIO CONSULTA DE SENTENCIA
Tipo de integración	WS - REST
URL	http://172.30.46.3/wsURT/madrpredioservice/getPredioByFMI
Entidad receptora	MinAgricultura
Sistema de información receptor	SIPOV

<b>Entidad proveedora</b>	<b>URT</b>
Sistema de información emisor	SENTENCIAS (FALLOS)
Servicio	SERVICIO CONSULTA DE SENTENCIA
Tipo de integración	WS - REST
URL	http://172.30.46.3/wsURT/madrsentenciaservice/getSentenciaByRadicado
Entidad receptora	MinAgricultura
Sistema de información receptor	SIPOV

<b>Entidad proveedora</b>	<b>URT</b>
Sistema de información emisor	SENTENCIAS (FALLOS)
Servicio	SERVICIO CONSULTA DE SENTENCIA
Tipo de integración	WS - REST
URL	http://172.30.46.3/wsURT/madrsentenciaservice/getSentenciaByFecha
Entidad receptora	MinAgricultura
Sistema de información receptor	SIPOV