



INSTITUTO ESTATAL ELECTORAL
DE BAJA CALIFORNIA SUR

INFORME CORRESPONDIENTE AL MES DE ENERO RESPECTO DEL DESARROLLO DEL SISTEMA DE CÓMPUTOS DISTRITALES Y MUNICIPALES (SISCOM) DEL PROCESO LOCAL ELECTORAL 2017-2018

ieebcs.org.mx



UCSI
UNIDAD DE CÓMPUTO
Y SERVICIOS INFORMÁTICOS

Contenido

Introducción.....	2
Estructura General de los Proyectos de Desarrollo del SISCOM.	3
Estado del Cronograma.....	4
Descripción General del Motor de Base de Datos.....	5
Elementos de Auditoria.....	5
Descripción General del Proyecto para los Servicios RestFul en Java.	7
Método Base.....	8
Clases de Seguridad.	8
Esquema General de implementación del SISCOM.	10
Repositorio de Código y Documentación del SISCOM.....	12



Introducción

El presente informe que presenta la Unidad de Cómputo y Servicios Informáticos (UCSI) da cuenta de los avances que presenta la Unidad respecto de la actual etapa de desarrollo del Sistema de Cómputos Distritales y Municipales (SISCOM) a utilizarse en el Proceso Local Electoral 2017-2018.

El documento consta de diversas secciones en las cuales se mencionan los avances en el desarrollo del SISCOM, principalmente destacar que el avance presentado está enfocado en la codificación de servicios que utilizará el sistema a nivel de base de datos y de seguridad, en específico lo relativo a la bitácora de operaciones del sistema. Dichos elementos de carácter interno se consideran medulares en el correcto funcionamiento y ejecución del SISCOM.



Estructura General de los Proyectos de Desarrollo del SISCOM.

Recordemos que dentro de la etapa de Análisis del SISCOM, se estableció la metodología de trabajo así como la arquitectura de desarrollo y los elementos que la contemplan, recordando que es importante puntualizar que se optó por un esquema híbrido, en el cual se tiene una aplicación de escritorio en C# (utilizada por los órganos desconcentrados) y un contenedor de servicios RestFul en Java JEE. Aspecto importante a mencionar, es que la parte de Administración y Control del SISCOM será mediante una interfaz Web, la cual de igual forma está contemplada en la actual etapa de desarrollo. Con esto en mente, a continuación se presentan los siguientes elementos que contemplan la solución tecnológica completa.

Base de datos: Motor de Base de Datos en MySQL, derivado que se considera que cuenta con la capacidad suficiente para poder soportar las operaciones y transacciones que se realicen al más bajo nivel del Sistema, a la par de la experiencia en el uso del mismo por parte del personal de la UCSI. Aspecto muy importante también, es que MySQL da la posibilidad de crear Vistas, Procedimientos Almacenados y Esquemas de Replicaciones, elementos cruciales para establecer el nivel esperado de disponibilidad de la información que contempla el SISCOM.

Servicios: Del tipo RestFul en Java JEE y como contenedor de servicios WildFly 10. Las herramientas de seguridad, administración y la documentación que existe para este modelo son muy extensas, además que es una solución típica cuando se trata de seguridad. Se contempla el uso de librería en Jersey y Jetty para manejar temas relacionado con permisos y potenciales amenazas, por ejemplo, filtros Anti-DDoS. Además de capas dentro del mismo contenedor que permiten gestionar de manera eficiente las conexiones y clientes.

Capa Intermedia: Para el lado de la aplicación en C#, cabe destacar que la capa intermedia no existirá y será el cliente directamente quien lleva a cabo la serialización y manejo de objetos (datos), sin embargo, para los módulos en interfaz Web se está manejando una capa PHP únicamente para llamar los servicios, agregando una capa de seguridad a los datos adicional.

Cliente: el cliente que reciba y haga las peticiones será de dos lugares diferentes:

- Aplicación C#: En el día de la Jornada Electoral y los Cómputos será el **SISCOM Escritorio**, encargado de realizar las peticiones por medio de

librerías que manejen seguridad e integridad en llamadas de servicios RestFul.

- Aplicación Web PHP-AngularJS: la aplicación Web servirá para la administración, por lo tanto se conectará mediante la llamada de servicios RestFul por PHP y manejará la parte del FrontEnd con AngularJS para realizar la serialización y manejo de objetos así como interacción con el usuario.

Contemplando lo anterior señalado, a continuación se menciona el avance de desarrollo en cada uno de dichos aspectos.

Estado del Cronograma. *

Dentro del Cronograma del SISCO, se establecieron Hitos (módulos) como productos entregables definidos en periodos de entre 6 a 18 días, dependiendo de su complejidad. De acuerdo a dicho cronograma, a la fecha de la sesión ordinaria del COTASISCO, nos encontramos actualmente en la semana 4 del Desarrollo del sistema, por lo que deberíamos estar situados en el término del módulo “Administrador del Sistema” e iniciando el hito “Jornada Electoral”. (Rojo)

Tareas	Fecha Inicio	Fecha Fin	Duración	Días Completados	Días por completar	Porcentaje de Avance
Módulo del Sistema	02/01/2018	27/02/2018	56	26.00	30.00	46%
Administrador del Sistema	02/01/2018	02/01/2018	0	0.00	0.00	0%
Control de Usuarios	02/01/2018	08/01/2018	6	5.00	1.00	0%
Reportes de Bitácoras	02/01/2018	08/01/2018	6	5.00	1.00	0%
Matriz de Seguridad	08/01/2018	14/01/2018	6	5.00	1.00	0%
Gestión de Insumos	08/01/2018	14/01/2018	6	6.00	0.00	0%
Generación de tokens	14/01/2018	20/01/2018	6	5.00	1.00	0%
Jornada Electoral	20/01/2018	20/01/2018	0	0.00	0.00	0%
Estado de Paquetes Electorales	20/01/2018	28/01/2018	8	0.00	8.00	0%
Resultados Preliminares	20/01/2018	28/01/2018	8	0.00	8.00	0%
Elementos Generales de las AFC	28/01/2018	03/02/2018	6	0.00	6.00	0%
Grupos de Trabajo	03/02/2018	03/02/2018	0	0.00	0.00	0%
Cálculo de GT y PR	03/02/2018	09/02/2018	6	0.00	6.00	0%
Captura de Cómputos	09/02/2018	09/02/2018	0	0.00	0.00	0%
Cómputo Distrital/Municipal	09/02/2018	27/02/2018	18	0.00	18.00	0%
Cómputo en GT	09/02/2018	15/02/2018	6	0.00	6.00	0%
Reportes	15/02/2018	27/02/2018	12	0.00	12.00	0%
Fin o Cierre	27/02/2018	27/02/2018	0	0.00	0.00	0%

Sin embargo, es preciso señalar que, derivado del proyecto de modificación del presupuesto del Instituto, la contratación del personal Analista Programador agendada para el mes de enero de 2018 se tuvo que aplazar para el mes de febrero, por lo tanto el avance real (Azul) es el que se muestra en la imagen anterior.

Respecto de las actividades pendientes de acuerdo al cronograma, estaríamos situados al término del módulo de Administrador del Sistema, sin embargo por lo señalado en el párrafo anterior, no se han podido cerrar dichas tareas derivado a que actualmente se tienen los servicios y base de datos así como las pruebas unitarias pero, falta la vista de los mismos.

Descripción General del Motor de Base de Datos.

El motor de base de datos es un elemento primordial para almacenar de manera segura y eficiente la información que maneja el SISCOP. Por ende es importante establecer ciertos mecanismos y estructuras para que dicha información pueda ser confiable y rápida a la hora de mostrarse.

Elementos de Auditoría.

En una base de datos debe existir la certeza de quién, cuándo y qué modificó cuando se trata de la información. Por ello las tablas se dividieron en dos grandes grupos para los elementos de la Auditoría:

- **Tablas con actualizaciones:** las tablas que contemplan actualizaciones en el sistema tienen elementos a nivel motor de base de datos para conocer la hora de modificación y a nivel aplicación de quien realizó dicha modificación. Por ende tienen los siguientes campos (sin excepción):

Campo	Descripción
Usuario de Creación	Campo con el Id del usuario que crea el registro, si es una tabla previamente cargada se asigna el id "1".
Fecha Creación	Fecha con TimeStamp que contiene la hora de creación del registro
Usuario Modificación	Id del usuario que actualiza el registro.
Fecha Modificación	Auto actualización cuando se actualiza el registro en la base de datos. Fecha con hora.

- **Tablas sin actualizaciones:** Tablas que por su naturaleza no debe ser posible modificarse, por ejemplo, auditorias. En estas tablas únicamente se pueden realizar inserciones.

Campo	Descripción
Usuario de Creación	Campo con el Id del usuario que crea el registro, si es una tabla previamente cargada se asigna el id "1".
Fecha Creación	Fecha con TimeStamp que contiene la hora de creación del registro

Los campos de usuario son obligatorios y no deben quedar en NULL (vacío) en ningún caso. Mientras que las fechas deben tener la hora, minutos y segundos de modificación.

Las tablas de la base de datos que ya se encuentran implementadas del SISCOM, se muestran en el siguiente diagrama. Es importante recalcar que las tablas contempladas son las que ya han sido utilizadas para crear la clase correspondiente que las manipula dentro de los Servicios, por lo que el procedimiento es, crear la tabla, luego su clase y finalmente los métodos para ser accedidos y manipulados desde su Servicio correspondiente.

siscom bitadmin	
Id	: int(11)
fecha_creacion	: timestamp
usuario_creacion	: int(11)
id_usuario	: int(11)
id_equipo	: int(11)
id_operacion	: int(11)
datos	: varchar(500)
estatus	: int(1)

siscom tipos_usuario	
Id	: int(11)
fecha_creacion	: timestamp
usuario_creacion	: int(11)
fecha_modificacion	: timestamp
usuario_modificacion	: int(11)
clave	: varchar(2)
descripcion	: varchar(50)

siscom equipos_computo	
Id	: int(11)
fecha_creacion	: timestamp
usuario_creacion	: int(11)
fecha_modificacion	: timestamp
usuario_modificacion	: int(11)
nombre	: varchar(100)
mac	: varchar(12)
ip	: varchar(15)
estatus	: int(1)

siscom ambitos	
Id	: int(11)
valor	: varchar(5)
fecha_creacion	: timestamp
usuario_creacion	: int(11)
fecha_modificacion	: timestamp
usuario_modificacion	: int(11)

siscom usuarios	
Id	: int(11)
fecha_creacion	: timestamp
usuario_creacion	: int(11)
fecha_modificacion	: timestamp
usuario_modificacion	: int(11)
usuario	: varchar(20)
id_empleado	: int(11)
id_equipo	: int(11)
password	: varchar(150)
idambito	: int(11)
id_tipo	: int(11)
estatus	: int(1)

siscom operaciones	
Id	: int(11)
fecha_creacion	: timestamp
usuario_creacion	: int(11)
fecha_modificacion	: timestamp
usuario_modificacion	: int(11)
operacion	: varchar(100)

siscom matriz_seguridad	
Id	: int(11)
fecha_creacion	: timestamp
usuario_creacion	: int(11)
fecha_modificacion	: timestamp
usuario_modificacion	: int(11)
ip	: varchar(15)
mac	: varchar(16)
id_equipo	: int(11)
id_operacion	: int(11)
id_usuario	: int(11)
estatus	: int(1)

siscom tokens	
id	: int(11)
fecha_creacion	: timestamp
usuario_creacion	: int(11)
fecha_modificacion	: timestamp
usuario_modificacion	: int(11)
token	: varchar(500)
comentarios	: varchar(100)
estatus	: int(1)
observaciones	: varchar(150)
estado_token	: int(1)

Descripción General del Proyecto para los Servicios RestFul en Java.

El avance más significativo que se señala en el presente informe, es en respecto de la capa de negocios, y es donde efectivamente se invierte más tiempo para realizar validaciones, así como pruebas unitarias por función y elementos que permitan localizar bugs o fallos.

En este rubro se establecieron funciones, clases y librerías que permitan poner cimientos para soportar todo el proyecto del SISCOP. Para ello como primer actividad se creó la estructura del proyecto en Eclipse, con el objetivo de tener un orden, control y conocer perfectamente el objetivo que se pretende lograr con cada uno de los componentes.

Método Base.

El proyecto contiene la estructura de archivos estándar de Java, siendo este por paquetes. Cada paquete contiene un grupo de clases relacionadas entre sí y están perfectamente separadas con el objetivo de ser más legibles, organizadas y seguras a la hora de modificar o crear tanto nuevas implementaciones como clases en Java.

Toda petición que se realiza al contenedor de servicios forzosamente pasa por las capas de las clases de seguridad, así mismo queda registrada dicha actividad dentro de las bitácoras correspondientes, tanto a de nivel de código como de nivel de servidor.

Clases de Seguridad.

Se establecieron clases de seguridad que se describieron en los documentos de análisis y se detallan a continuación:

HeaderIEE.java: es un encabezado único por usuario que permite tener datos de control y administración dentro del Sistema. Sin excepción alguna, este encabezado debe existir en toda solicitud que se realice. La importancia del mismo es que contiene datos relevantes y de seguridad que no pueden faltar para interactuar en el sistema como son los siguientes:

```
private String idUsuario;  
private String usuario;  
private String ambito;  
private String tipo;  
private String referencia;  
private String ip;  
private String equipoAcceso;  
private String mac;
```

Destacando que dicho encabezado se envía de manera cifrada utilizando una adecuación al algoritmo Base64, con lo que se fortalece el nivel de seguridad en el SISCO, y si bien es cierto puede ser susceptible de suplantación, es preciso señalar que no será el único mecanismo de seguridad, los cuales han sido señalados en el documento de Análisis del SISCO.

Autorizacion.java: Es una clase lógica de muy alto impacto. No sólo se encarga de leer los encabezados; sino también de validar su origen. Contiene los métodos de: validar origen de la petición, autenticación básica por encabezado y autoriza matriz de seguridad.

ResponseWS.java: se diseñó una clase genérica para manejar los mensajes (como bien se establece en los principios de los servicios REST) para poder normalizar la comunicación tanto de la aplicación de escritorio como de la administración web.

Dicha clase contiene la posibilidad de almacenar mensajes dentro de la petición, así como código de errores para transportarlos a la interfaz final.

Data.properties: archivo de configuración que permite manejar datos relevantes a nivel de contenedor de aplicaciones, como conexiones de origen, datos de conexión con la base de datos, entre otros.

El avance del desarrollo de los servicios también contempla métodos que permiten interactuar con la base de datos y así mismo registrar interacciones con los mismos en la bitácora de operaciones.

Los servicios que ya se implementaron bajo este esquema son los siguientes:

- **ServicioAuditoria.**
- **ServicioTokens.**
- **ServicioUsuarios.**

Cabe señalar que por ser Servicios, éstos tienen una interfaz pública, es decir, una ruta para llamarse desde cualquier cliente. Estos a su vez tienen clases que permiten realizar sus funciones descritas en el documento de Análisis, así como en el Anexo de Casos de Uso descritos con anterioridad.

Dentro de la funcionalidad relacionada, se contemplan los siguientes avances:

- **AuditoriaDAO.**
- **MatrizSeguridadDAO.**
- **TokenDAO.**
- **UsuarioDAO.**

Los DAO (Data Access Object, o los objetos de acceso a datos) están ligados a sus clases correspondientes, y tienen su interfaz de acceso, en específico los BO (Business Object o los objetos de negocio), tal como lo dictan los estándares de JAVA.

Éstos a su vez tienen las clases necesarias para interactuar con todos los elementos que ocupa, por ejemplo, UsuarioDAO necesita forzosamente a EquipoComputo, TipoUsuario, Ambitos y Empleados para efectos de estar completa en su funcionalidad y relaciones.

Con estas clases, funciones y servicios se logran los hitos correspondientes de la semana 3 del cronograma de desarrollo, faltando únicamente las vistas para presentarse ante el usuario.

Esquema General de implementación del SISCOM.

Contextualizando el avance, ya se realizó programáticamente el funcionamiento medular de los servicios y trazabilidad de los servicios. Por medio del siguiente ejemplo se ilustrará cómo funcionan dichos elementos.

Se tiene por el lado del cliente, tres elementos que son forzosamente necesarios para aceptar una petición como válida en el contenedor de servicios del SISCOM:

- Un encabezado (definido por la variable **Headers**). Otra es la
- Entrada o datos de la operación (definidos por **Input**) y por último,
- La ruta de acceso del servicio (definida por **Url**).

Si estos elementos no están presentes en la petición, no será posible realizarse. Tras realizar la petición vía POST (y es el único medio admitido), se envía al servicio para validar todos sus elementos.

Dentro del servicio a nivel del contenedor se realiza la validación de CORS y de DoSFilter:

CORS: El Intercambio de Recursos de Origen Cruzado (CORS) es un mecanismo que utiliza encabezados adicionales HTTP para permitir que un user agent obtenga permiso para acceder a recursos seleccionados desde un servidor, en un origen distinto (dominio), al que pertenece.

DoSFilter: El filtro DoS realiza un seguimiento del número de solicitudes de una conexión por segundo. Si las solicitudes superan el límite, Jetty rechaza, retrasa o acelera la solicitud y envía un mensaje de advertencia.

Una vez validado, se realiza lo siguiente:

- Atrapar el código correspondiente a encabezado y petición.
- Enviar a autorización los datos.
- Atrapar el resultado de autorización.
- Llamar al servicio correspondiente de la petición.
- Regresar en ResponseWS al cliente.

Donde se reciben los datos del encabezado y de petición. Así mismo se llama a la clase antes mencionada de Autorización, donde le envía los datos de la petición, autenticación, encabezado y el ID de la operación a realizar para validarla con la matriz de seguridad.

Puede suceder que se rechace la solicitud, al realizar la petición en dos niveles:

- Si es a nivel de encabezado de la petición, se registra el intento en el Log del servidor.
- Si se rechaza a nivel de matriz de seguridad, se registra en la base de datos.

Ejemplo de log en el servidor:

```
2018-01-20 12:15:23,212 INFO [stdout] (default task-54) Authorization:Basic aXh4Ynh4OnNIY3VyZQ==
2018-01-20 12:15:23,212 INFO [stdout] (default task-54) Accept:application/json
2018-01-20 12:15:23,212 INFO [stdout] (default task-54) Connection:Keep-Alive
2018-01-20 12:15:23,212 INFO [stdout] (default task-54) X-Forwarded-Host:eleccionesbcs.org.mx
2018-01-20 12:15:23,213 INFO [stdout] (default task-54) X-Forwarded-For:187.191.78.145
2018-01-20 12:15:23,213 INFO [stdout] (default task-54) Host:127.0.0.1:8080
2018-01-20 12:15:23,213 INFO [stdout] (default task-54) Content-Length:379
2018-01-20 12:15:23,213 INFO [stdout] (default task-54) X-Forwarded-Server:187.191.78.145
2018-01-20 12:15:23,213 INFO [stdout] (default task-54) Content-Type:application/json
2018-01-20 12:15:23,213 INFO [stdout] (default task-54) ixxbxx:secure127.0.0.1
2018-01-20 12:15:23,216 INFO [stdout] (default task-54) Conexión a base de datos usuarios OK
```

Ejemplo de log en la matriz:

```
Negación por MatrizSeguridad : HeaderIEE [idUserio=1, usuario=Chuthulhu, ambito=IEE, tipo=AA,
referencia=356a192b7913b04c54574d18c28d46e6395428ab, ip=192.168.11.1, equipoAcceso=1,
mac=XX:XX:XX:XX] : Operación solicitada = 12
```

Si la operación se realiza correctamente o incorrectamente, se crea un registro de la operación, con un estatus de "1" sí se realizó correctamente o un "0" sí esta inconclusa. De igual forma se registra el error en el Log del servidor.

Columna	Tipo	Función	Nulo	Valor
id	int(11)	<input type="text"/>		19
fecha_creacion	timestamp	<input type="text"/>		2018-01-15 19:03:15
usuario_creacion	int(11)	<input type="text"/>		1
id_usuario	int(11)	<input type="text"/>		1
id_equipo	int(11)	<input type="text"/>		1
id_operacion	int(11)	<input type="text"/>		2
datos	varchar(500)	<input type="text"/>		<pre> Usuario [id=2, usuarioCreacion=0, fechaCreacion=null, usuarioModificacion=0, fechaModificacion=null, usuario= ESTE ES NUEVO , idEmpleado=1, idEquipo=1, password=null, idAmbito=1, idTipo=1, estatus=0] </pre>
estatus	int(1)	<input type="text"/>		0

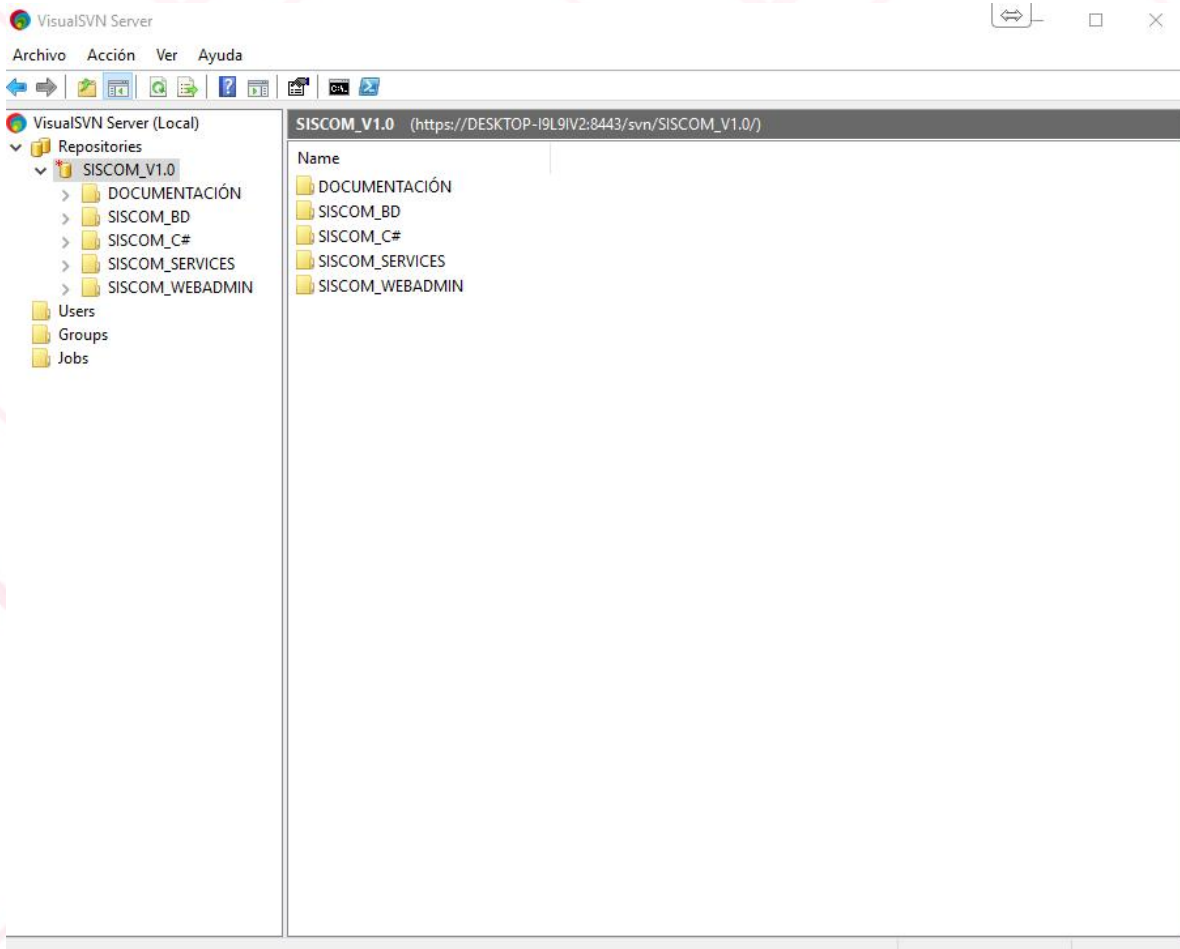
Interfaz de Phpmyadmin, se muestra los datos registrados en bitácora.

Una vez terminada la solicitud, se regresa en ResponseWS los datos de la operación concluida con los objetos en formato JSON de lo solicitado.

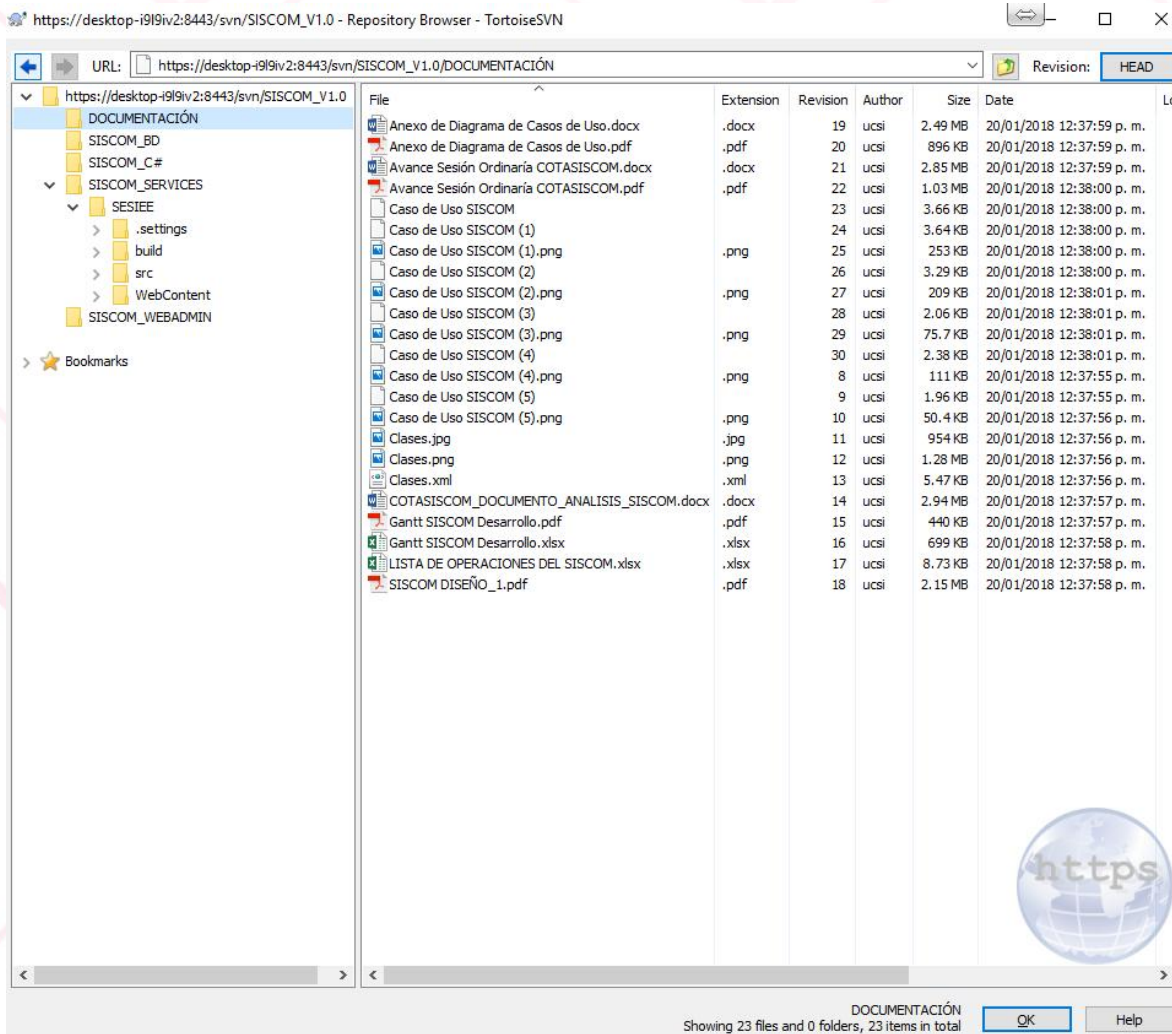
Repositorio de Código y Documentación del SISCOM.

Como mecanismo de resguardo de código y de documentación del SISCOM, se utilizará un servidor SVN donde se crearán carpetas por rubro y además por versión, con la posibilidad de hacer comentarios de los mismos y generar ramas para contar una mejor administración y control. Por el momento el repositorio será local en una máquina instalada en la Unidad, mientras no exista la necesidad de compartir con otro desarrollador de manera remota el código fuente o proyectos de avances.

Para lo anterior, se utilizará **VisualSVN** en su versión gratuita como servidor y como cliente **TortoiseSVN**, visualizándolo de la siguiente manera en su conjunto:



Interfaz de VisualSVN Server, como repositorio de código y control de versiones.



Interfaz de TortoiseSVN, cliente de acceso al repositorio.

De acuerdo al avance en el proyecto, se podrá ir agregando nuevas carpetas y ramas para llevar correctamente el código y documentación del SISCOM.