



Presidencia de la República

OFICINA PRESIDENCIAL DE TECNOLOGÍAS
DE LA INFORMACIÓN Y COMUNICACIÓN



NORTIC A4 2015



GOBIERNO
ELECTRÓNICO
REPÚBLICA DOMINICANA

GUÍA DE IMPLEMENTACIÓN INTEROPERABILIDAD

Santo Domingo, República Dominicana
Mayo 2015



Presidencia de la República

OFICINA PRESIDENCIAL DE TECNOLOGÍAS
DE LA INFORMACIÓN Y COMUNICACIÓN

NORTIC A4 2015

Guía de Implementación de la Norma para la Interoperabilidad
entre los Organismos del Gobierno Dominicano

Santo Domingo, República Dominicana

7 de Mayo, 2015

NORTIC A4-2:2015

**Guía de Implementación de la Norma para la Interoperabilidad
entre los Organismos del Gobierno Dominicano**

Edición: 1era

Departamento de Estandarización, Normativas y Auditoría Técnica

Fecha de aprobación: 22 de enero de 2015

Fecha de lanzamiento: 07 de mayo de 2015

Categoría: A

Serie de documento: 2

Año de publicación: 2015

Versión 0.1.0

Diseñado y Diagramado por el Departamento Multimedia, OPTIC

Impreso en República Dominicana



CONTENIDO

PRÓLOGO.....	ix
MARCO LEGAL.....	xiii
INTRODUCCIÓN.....	xxi

CAPÍTULO I.

Guía de Implementación de la Norma para la Interoperabilidad entre los Organismos del Estado Dominicano.....	23
SECCIÓN 1.01 Alcance.....	23
SECCIÓN 1.02 ¿Cómo utilizar esta guía?.....	24
SECCIÓN 1.03 Términos y definiciones.....	27
SECCIÓN 1.04 Uso del catálogo de estándares.....	28
Sub-sección 1.04.1 Selección de estándares del catálogo.....	29
Sub-sección 1.04.2 Uso de estándares no presentes en la normativa.	29

CAPÍTULO II.

Implementación de la Interoperabilidad Organizacional.....	31
SECCIÓN 2.01 Acuerdos organizacionales.....	31
Sub-sección 2.01.1 Tipos de acuerdos.....	31
SECCIÓN 2.02 Gestión de procesos organizacionales.....	35
Sub-sección 2.02.1 Objetivos para la gestión de procesos.....	35
SECCIÓN 2.03 Gestión de proyectos de interoperabilidad.....	37
Sub-sección 2.03.1 Definición de roles.....	38
Sub-sección 2.03.2 Creación del acta de constitución del proyecto.....	39
Sub-sección 2.03.3 Elaboración del diagrama de planificación del proyecto.....	41
Sub-sección 2.03.4 Elaboración del documento de cierre del proyecto.....	42



CAPÍTULO III.

Implementación de la Interoperabilidad Semántica..... 45

SECCIÓN 3.01 Documentación de visualización de la información.45

Sub-sección 3.01.1 Especificación de formato de visualización de la información..... 46

Sub-sección 3.01.2 Diseño de presentación de datos para dispositivos..... 46

Sub-sección 3.01.3 Documentación con esquemas de páginas... 47

SECCIÓN 3.02 Metadatos..... 49

Sub-sección 3.02.1 Documentación interna de metadatos..... 49

Sub-sección 3.02.2 Documentación externa de metadatos..... 50

Sub-sección 3.02.3 Vocabularios externos a utilizar..... 51

CAPÍTULO IV.

Implementación de la Interoperabilidad Técnica..... 65

SECCIÓN 4.01 Protocolos para el intercambio de información e hipertextos..... 65

Sub-sección 4.01.1 Clases de protocolos seguros de tipo FTP..... 66

Apartado 4.01.1.1 Protocolo de transferencia de archivos mediante una capa de conexión segura 66

Apartado 4.01.1.2 Protocolo de Transferencia de Archivos Seguro..... 67

Apartado 4.01.1.3 Comparación entre el SFTP y el protocolo FTPS..... 68

Sub-sección 4.01.2 Implementación de un servidor SFTP en Linux..... 68

Sub-sección 4.01.3 Implementación de un servidor SFTP en Windows..... 71

Apartado 4.01.3.1 Posibles problemas durante la instalación..... 82

Sub-sección 4.01.4 Clases de protocolos de transferencia de hipertexto..... 84

Apartado 4.01.4.1 Protocolo de transferencia de hipertexto..... 84

Apartado 4.01.4.2 Protocolo de transferencia de hipertexto Seguro..... 85





Apartado 4.01.4.3 Recomendación de uso del protocolo HTTP. 85

Apartado 4.01.4.3 Implementación de Protocolo HTTPS..... 86

SECCIÓN 4.02 Implementación de bases de datos..... 95

Sub-sección 4.02.1 Diseño de bases de datos..... 95

Sub-sección 4.02.2 Tipos de bases de datos..... 97

Apartado 4.02.2.1 Bases de datos relacionales..... 97

Apartado 4.02.2.2 Implementación de servidor de base de datos relacionales..... 101

Apartado 4.02.2.3 Bases de datos no relacionales..... 112

SECCIÓN 4.03 Archivos para almacenamiento de datos..... 115

Sub-sección 4.03.1 Tipos de archivos de almacenamiento.....116

Apartado 4.03.1.1 Formato de notación de objetos de JavaScript116

Apartado 4.03.1.2 Lenguaje de marcas extensible 118

Apartado 4.03.1.3 Formato de valores separados por coma 122

Apartado 4.03.1.4 Ficheros de valores separados tabulaciones . 123

SECCIÓN 4.04 Documentación del sistema..... 124

Sub-sección 4.04.1 Esquema documental..... 124

SECCIÓN 4.05 Manejo de código fuente con Git..... 130

Sub-sección 4.05.1 Instalación de Git..... 130

Apartado 4.05.1.1 Instalación desde código fuente..... 131

Apartado 4.05.1.2 Instalación en Linux..... 132

Apartado 4.05.1.3 Instalación en Mac..... 132

Apartado 4.05.1.4 Instalación en Windows..... 133

Sub-sección 4.05.2 Configurando Git por primera vez..... 133

Apartado 4.05.2.1 Personalizando la identidad..... 134

Apartado 4.05.2.2 Personalizando el editor..... 135

Apartado 4.05.2.3 Herramienta de diferencias..... 135

Apartado 4.05.2.4 Comprobando la configuración..... 135

Sub-sección 4.05.3 Obtener Ayuda..... 136

Apartado 4.05.3.1 Obteniendo un repositorio Git..... 136

Apartado 4.05.3.2 Clonando un repositorio existente..... 137



Apartado 4.05.3.3 Guardando cambios en el repositorio.....	138
Apartado 4.05.3.4 Comprobando el estado de los archivos.....	139
Apartado 4.05.3.5 Seguimiento de nuevos archivos.....	141
Apartado 4.05.3.6 Preparando archivos modificados.....	142
Apartado 4.05.3.7 Ignorando archivos.....	144
Apartado 4.05.3.8 Viendo los cambios preparados y no preparados.....	146
Apartado 4.05.3.9 Confirmando los cambios.....	149
Apartado 4.05.3.10 Saltar el área de preparación.....	151
Apartado 4.05.3.11 Eliminando archivos.....	152
Apartado 4.05.3.12 Moviendo archivos.....	154
Sub-sección 4.05.4 Metodología avanzada de manejo de calidad de código con Git.....	155
Apartado 4.05.4.1 Inicio de git-flow.....	158
Apartado 4.05.4.2 Creación de una nueva funcionalidad o corrección de un error.....	159
Apartado 4.05.4.3 Nombres de las ramas.....	159
Apartado 4.05.4.4 Publicando la nueva rama	160
Apartado 4.05.4.5 Otros colaboradores.....	161
Apartado 4.05.4.6 Finalización de la funcionalidad.....	162
Apartado 4.05.4.7 Limpieza.....	162
Apartado 4.05.4.8 Distribuciones.....	163
Apartado 4.05.4.9 Nombres de las distribuciones.....	165
Apartado 4.05.4.10 Corrección de errores.....	165
SECCIÓN 4.06 Interfaz de usuario web.....	167
Sub-sección 4.06.1 Características de una buena interfaz de usuario.....	167
Sub-sección 4.06.2 Marcos de trabajo para interfaces de usuario.....	169
Apartado 4.06.2.1 Descargando Bootstrap.....	169
Apartado 4.06.2.2 Requerimientos para trabajar con Bootstrap.....	171
Apartado 4.06.2.3 Instalación de Bootstrap.....	172



Apartado 4.06.2.4 Sistema de cuadrícula para la web.....	173
Apartado 4.06.2.5 Componentes básicos.....	175
Apartado 4.06.2.6 Componentes avanzados.....	179
GLOSARIO DE TÉRMINOS.....	183
ABREVIATURAS Y ACRÓNIMOS.....	196
BIBLIOGRAFÍA.....	200
ANEXOS.....	203
EQUIPO DE TRABAJO.....	210

PRÓLOGO

La Oficina Presidencial de Tecnologías de la Información y Comunicación (OPTIC), es el organismo del Estado Dominicano responsable de fomentar el uso de las Tecnologías de la Información y Comunicación (TIC), creado mediante el decreto No. 1090-04, en fecha 3 de septiembre de 2004, como dependencia directa del Poder Ejecutivo, con autonomía financiera, estructural y funcional, a fin de garantizar eficiencia, transparencia, servicios en línea y mecanismos para rendición de cuentas disponibles a favor de la ciudadanía.

El plan estratégico vigente de la OPTIC consta de 5 ejes sobre los cuales se ejecuta la estrategia de gobierno electrónico del estado dominicano. El eje normativo es el primero y el que tiene la mayor incidencia de transversalidad ya que tiene por objetivo impulsar la implementación del gobierno electrónico en el país.

La OPTIC ha creado el departamento de Estandarización, Normativas y Auditoría Técnica, para que desarrolle las normas y estándares tecnológicos que velen por del correcto uso e implementación de las TIC en el Estado.

Las normas sobre TIC, denominadas NORTIC, creadas desde el año 2013, fueron concebidas para normalizar, estandarizar y auditar el efectivo uso e implementación de las TIC, para ser difundidas e implementadas en toda la administración pública, con el fin de lograr homogeneidad y mejora de los procesos entre los organismos gubernamentales.





En este contexto, se han definido 5 categorías o tipos de NORTIC, según el alcance, estas son:

Categoría A (normas universales), para los aspectos normativos que aplican a todos los organismos gubernamentales.

Categoría B (normas para los departamentos de TIC), para aquellas normas necesarias y exclusivas a la efectiva gestión de los departamentos o áreas de TIC dentro de los distintos organismos del Estado Dominicano.

Categoría C (normas municipales), para las normas que aplican a las iniciativas de TIC en los ayuntamientos o municipios.

Categoría D (normas para embajadas), para las normas que aplican únicamente a las iniciativas de TIC de las embajadas, consulados o misiones en el extranjero.

Categoría E (normas especiales), para las normas que aplican a organismos gubernamentales con características específicas dependiendo de sus funciones y estructura orgánica, así como para iniciativas, proyectos o programas de Gobierno, en el cual se haga uso de las TIC.

Esta Guía de implementación de la Norma para la Interoperabilidad entre los Organismos del Gobierno Dominicano, por tener un alcance universal, pertenece a la categoría A; mientras que por ser la cuarta NORTIC en elaborarse, su denominación sería NORTIC A4:2015, siendo los últimos 4 dígitos los referidos al año de lanzamiento de esta norma.



En algunos casos, esta guía de implementación puede presentarse de la forma siguiente NORTIC A4-2:2015, seguida de trece caracteres (#####-##-#####), donde el número “2” que aparece después del guion (-) especifica la serie del documento (1 para directrices, 2 para guías de implementación, 3 para código de buenas prácticas, entre otros) y los demás caracteres, el Número de Identificación Único (NIU) para cada organismo del Estado.

La evaluación de cada NORTIC y Guías de implementación son realizadas por dos comités, la primera evaluación es ejecutada por el Comité Interno para Evaluación de las Normas (CIEN), el cual está conformado por expertos en TIC dentro de la OPTIC, mientras que la segunda evaluación es realizada por el Comité de Estándares de Tecnologías de la Información y Comunicación (COETIC), el cual está conformado por los responsables de TIC de cada organismo gubernamental, o a quienes la máxima autoridad de cada organismo designe.

En vista de la responsabilidad de la OPTIC en la elaboración de políticas, estrategias y controles de TIC y de los avances en el uso de las tecnologías, de los cuales los organismos gubernamentales no quedan al margen, surge esta normativa para garantizar el uso efectivo de las plataformas y los procesos tecnológicos que son implementados por cada uno de dichos organismos.

MARCO LEGAL

La OPTIC, en su rol de entidad normalizadora sobre el uso e implementación de TIC en la administración pública, ha establecido las directrices por las cuales debe regirse todo organismo gubernamental del Estado Dominicano, tanto para aquellos que están físicamente dentro del país, como para los organismos que se encuentran fuera, como son las embajadas, consulados y misiones en el extranjero.

El marco legal que soporta esta guía está compuesto por las leyes y decretos presidenciales presentados a continuación:

1. El **Decreto 1090-04**, a través del cual se constituye la OPTIC como dependencia directa del poder ejecutivo, donde se establece lo siguiente:
 - Artículo 3.- Serán funciones de la Oficina Presidencial de Tecnologías de la Información y Comunicación, diseñar, proponer y definir las políticas, establecer los lineamientos y elaborar las estrategias, planes de acción y programas para promover, desarrollar y garantizar mayor acceso, conectividad e implantación de nuevas tecnologías de la información y comunicación, con el fin de fomentar su uso, como soporte del crecimiento económico y competitividad hacia la sociedad de la información, gestionando los proyectos conforme a tales directrices; y que garanticen el acceso equitativo a los mercados y al sector productivo como esfuerzo para la política de generación de empleo, para mejorar la calidad de vida, crear oportunidades de educación, trabajo, justicia, cultura y recreación, y otros.



- Artículo 5.- La Oficina Presidencial de Tecnologías de la Información y Comunicación será responsable de la formulación de políticas y la implementación del proceso de desarrollo e innovación tecnológica para la transformación y modernización del Estado hacia la sociedad de la información, promoviendo la integración de nuevas tecnologías, su compatibilidad, interoperabilidad y estandarización en materia de TIC.
 - Artículo 7.- La Oficina Presidencial de Tecnologías de la Información y Comunicación podrá proponer políticas para difundir y promover la generación de una cultura de TIC en el país.
 - Artículo 9.- La Oficina Presidencial de Tecnologías de la Información y Comunicación deberá velar, asistir y supervisar en los aspectos y políticas relativas a la seguridad y privacidad de la información digitalizada y electrónica en el ámbito del sector público.
2. Para el tratamiento de los derechos sobre la protección de datos personales, esta guía se ampara en la propia Constitución de la República Dominicana del 26 de enero de 2010.
- Artículo 44.- Derecho a la intimidad y el honor personal. Toda persona tiene derecho a la intimidad. Se garantiza el respeto y la no injerencia en la vida privada, familiar, el domicilio y la correspondencia del individuo. Se reconoce el derecho al honor, al buen nombre y a la propia imagen. Toda autoridad o particular que los viole está obligado a resarcirlos o repararlos conforme a la ley. Por tanto:
 - o Toda persona tiene el derecho a acceder a la información y a los datos que sobre ella o sus bienes reposen en los registros oficiales o privados, así como conocer el destino y el uso que se haga de los mismos, con las limitaciones fijadas



por la ley. El tratamiento de los datos e informaciones personales o sus bienes deberá hacerse respetando los principios de calidad, licitud, lealtad, seguridad y finalidad. Podrá solicitar ante la autoridad judicial competente la actualización, oposición al tratamiento, rectificación o destrucción de aquellas informaciones que afecten ilegítimamente sus derechos.

- o Se reconoce la inviolabilidad de la correspondencia, documentos o mensajes privados en formatos físico, digital, electrónico o de todo otro tipo. Sólo podrán ser ocupados, interceptados o registrados, por orden de una autoridad judicial competente, mediante procedimientos legales en la sustanciación de asuntos que se ventilen en la justicia y preservando el secreto de lo privado, que no guarde relación con el correspondiente proceso. Es inviolable el secreto de la comunicación telegráfica, telefónica, cablegráfica, electrónica, telemática o la establecida en otro medio, salvo las autorizaciones otorgadas por juez o autoridad competente, de conformidad con la ley.
- o El manejo, uso o tratamiento de datos e informaciones de carácter oficial que recaben las autoridades encargadas de la prevención, persecución y castigo del crimen, sólo podrán ser tratados o comunicados a los registros públicos, a partir de que haya intervenido una apertura a juicio, de conformidad con la ley.

3. La **Ley 107-13**, sobre los derechos de las personas en sus relaciones con la administración pública y de procedimiento administrativo, en donde se regulan los derechos y deberes de las personas y sus relaciones con la administración pública y se establecen los principios que sirven de sustento a esa relación, indicando los procedimientos administrativos.



- Artículo 4. Derecho a la buena administración y derechos de las personas en sus relaciones con la administración pública. Se reconoce el derecho de las personas a una buena administración pública, que se concreta, entre otros, en los siguientes derechos subjetivos de orden administrativo:
 - Derecho a no presentar documentos que ya obren en poder de la administración pública o que versen sobre hechos no controvertidos o no relevantes.

- Artículo 27. Actos de instrucción o investigación. Los actos de instrucción o investigación podrán consistir, entre otros, en los siguientes medios:
 - Párrafo I. Las actuaciones para la obtención y tratamiento de la información necesaria para adoptar una decisión bien informada podrán consistir en cualquier medio, como la cooperación, asistencia e intercambio de información con otras administraciones competentes, o las consultas a los expertos. En los términos establecidos en la legislación o en convenios internacionales, podrá recabarse la colaboración informativa de otras agencias y administraciones especializadas de otros Estados, o de organismos internacionales, al objeto de adoptar la decisión mejor informada, al servicio de los intereses generales.

4. La Ley 53-07 contra Crímenes y Delitos de Alta Tecnología.

- Artículo I.- Objeto de la Ley. La presente ley tiene por objeto la protección integral de los sistemas que utilicen tecnologías de la información y comunicación, y su contenido, así como la prevención y sanción de los delitos cometidos contra estos o cualquiera de sus componentes o los cometidos mediante



el uso de dichas tecnologías en perjuicio de personas física o morales, en los términos previstos en esta ley. La integridad de los sistemas de información y sus componentes, la información o los datos, que se almacenan o transmiten a través de estos, las transacciones y acuerdos comerciales o de cualquier otra índole que se llevan a cabo por su medio y la confidencialidad de estos, son todos bienes jurídicos protegidos.

5. La **Ley 200-04**, sobre el Libre Acceso a la Información Pública, que establece la implementación de la sección “Transparencia” en los portales del Gobierno Dominicano.
 - Artículo 5.- Se dispone la informatización y la incorporación al sistema de comunicación por Internet⁽¹⁾ o a cualquier otro sistema similar que en el futuro se establezca, de todos los organismos públicos centralizados y descentralizados del Estado, incluyendo el Distrito Nacional y los municipios, con la finalidad de garantizar a través de este, un acceso directo del público a la información del Estado. Todos los poderes y organismos del Estado deberán instrumentar la publicación de sus respectivas “páginas web⁽²⁾” a los siguientes fines:
 - o Difusión de información: Estructura, integrantes, normativas de funcionamiento, proyectos, informes de gestión, base de datos.
 - o centro de intercambio y atención al cliente o usuario: Consultas, quejas y sugerencias.
 - o trámites o transacciones bilaterales.

(1) Es una red mundial que conecta computadoras de todos los lugares del mundo para que los usuarios puedan interactuar entre sí sin importar el lugar.

(2) También llamada página electrónica; no es más que un documento electrónico o una página que presenta cierta información de interés.



- o la información a que hace referencia el párrafo anterior, será de libre acceso al público sin necesidad de petición previa.
- Artículo 6.- La administración pública, tanto centralizada como descentralizada, como cualquier otro órgano o entidad que ejerza funciones públicas o ejecute presupuesto público, y los demás entes y órganos mencionados en el Artículo 1 de esta ley, tienen obligación de proveer la información contenida en documentos escritos, fotografías, grabaciones, soportes magnéticos o digitales, o en cualquier otro formato, y que haya sido creada u obtenida por ella o que se encuentre en su posesión y bajo su control.
- Artículo 11.- La información solicitada podrá ser entregada en forma personal, por medio de teléfono, facsímile, correo ordinario, certificado o también correo electrónico⁽³⁾, o por medio de formatos disponibles en la página de Internet que al efecto haya preparado la administración a la que hace referencia el Artículo 1 de esta ley.
- Artículo 24.- Las entidades o personas que cumplen funciones públicas o que administren recursos del Estado deberán prever en sus presupuestos las sumas necesarias para hacer publicaciones en los medios de comunicación colectiva, con amplia difusión nacional, de los proyectos de reglamentos y actos de carácter general, a los que se ha hecho referencia en el artículo anterior.
- o Párrafo.- En los casos en que la entidad o persona correspondiente cuente con un portal de Internet o con una página en dicho medio de comunicación, deberá prever la existencia de un lugar específico en ese medio

(3) Es un servicio de mensajería en red que permite el intercambio de mensajes, a través de sistemas de comunicación electrónicos.



para que los ciudadanos puedan obtener información sobre los proyectos de reglamentación, de regulación de servicios, de actos y comunicaciones de valor general, que determinen de alguna manera la forma de protección de los servicios y el acceso de las personas de la mencionada entidad. Dicha información deberá ser actual y explicativa de su contenido, con un lenguaje entendible al ciudadano común.

o Debe publicarse el contenido utilizando medios tecnológicos que garanticen la autenticidad de la información, tales como certificados digitales.

6. La **Ley No. 126-02** sobre Comercio Electrónico, Documentos y Firma Digital.
7. La **Ley 1-12**, sobre estrategia nacional de desarrollo.
 - Artículo 16. En el diseño y ejecución de los programas, proyectos y actividades en que se concretan las políticas públicas, deberá promoverse el uso de las tecnologías de la información y comunicación como instrumento para mejorar la gestión pública y fomentar una cultura de transparencia y acceso a la información, mediante la eficientización de los procesos de provisión de servicios públicos y la facilitación del acceso a los mismos.
8. El **Decreto No. 229-07**, el cual es el instructivo de aplicación de Gobierno Electrónico, contentivo de las pautas generales para el desarrollo de la Estrategia de Gobierno Electrónico en la República Dominicana.
9. El **Decreto No. 709-07** sobre las normas y estándares elaboradas por la OPTIC.



- Artículo 1.- Se instruye a toda administración pública del Estado Dominicano a cumplir con las normas y los estándares tecnológicos para: (i) el desarrollo de portales gubernamentales, (ii) conectividad interinstitucional, (iii) interoperabilidad tecnológica, (iv) de seguridad, auditoría e integridad electrónica, (v) digitalización de documentos; así como cualquier otra normativa que sea redactada, aprobada y coordinada por la Oficina Presidencial de Tecnologías de la Información y Comunicación (OPTIC), en materia de tecnología de la información y la comunicación (TIC) y Gobierno Electrónico.
10. El **Decreto No. 130-05**, que aprueba el reglamento de la Ley General de Libre Acceso a la Información Pública.
 11. El **Decreto 335-03**, que aprueba el Reglamento de Aplicación de la Ley No. 126-02, sobre Comercio Electrónico, Documentos y Firmas Digitales.

INTRODUCCIÓN

La Guía de Implementación de la Norma para la Interoperabilidad entre los Organismos del Gobierno Dominicano establece las pautas que pueden seguir los organismos gubernamentales para lograr interoperar entre sí, permitiéndoles intercambiar información de una manera más efectiva y fluida para el beneficio de las instituciones del Estado Dominicano y del ciudadano.

En esta guía se presentan las 3 dimensiones de interoperabilidad presentadas en la NORTIC A4:2014. Cada una de estas dimensiones se encuentra en un capítulo diferente, en el cual se detallan las pautas que se utilizan para llevar a cabo estas dimensiones.

El primer capítulo presenta el alcance de la misma, la cual comprende todos los organismos del Estado Dominicano de manera mandataria, tanto para aquellos que están físicamente dentro del territorio dominicano, como para aquellos organismos que se encuentran fuera del país, como las embajadas, consulados y misiones en el extranjero y, de manera referencial, para los demás Poderes del Estado. En dicho capítulo se habla sobre el uso adecuado del catálogo de estándares. Además se mencionan los criterios que se deben tomar en cuenta para la selección de estándares del catálogo así como también el uso de los estándares no presentes en la NORTIC A4:2014.

El capítulo II trata sobre la implementación de la interoperabilidad organizacional, en donde se indica cómo se pueden definir los diferentes tipos de acuerdos establecidos en la NORTIC A4:2014. Además se indican las pautas para llevar a cabo una buena gestión de procesos organizacionales y una gestión de proyectos de interoperabilidad.



En el capítulo siguiente sobre la implementación de la interoperabilidad semántica, se indica la documentación que se debe tener en cuenta para visualizar la información, así como también las clases y propiedades que utilizan los metadatos⁽⁴⁾ para que puedan intercambiar información entre los diferentes sistemas informáticos, garantizando el entendimiento y aplicación de dicha información.

Para el capítulo final, sobre la implementación de la interoperabilidad técnica, se presentan los protocolos de intercambios de datos que permiten transferir archivos a través de internet. En el mismo se especifica tanto la implementación de un servidor del Protocolo de Transferencia de Archivos Seguro (SFTP⁽⁵⁾, por sus siglas en inglés), como la de una base de datos. También se indica la documentación requerida para los sistemas que se desarrollen dentro del organismo.

- (4) Son un conjunto de datos de información que describe las características de otra información. Es “datos sobre datos”.
- (5) Es un protocolo de red de utilizado para acceder y manejar archivos de manera remota utilizando métodos de encriptación.

CAPÍTULO I

GUÍA DE IMPLEMENTACIÓN DE LA NORMA PARA LA INTEROPERABILIDAD ENTRE LOS ORGANISMOS DEL ESTADO DOMINICANO

Esta guía establece los lineamientos y recomendaciones que puede seguir cada organismo del Gobierno Dominicano para garantizar que en las informaciones intercambiadas no exista ningún tipo ambigüedad por parte las aplicaciones que intervengan en una determinada transacción y al mismo tiempo habilitar a los sistemas para combinar información recibida con otros recursos de información. Todos estos lineamientos y recomendaciones tienen como objetivo permitir que esta guía sea una herramienta de significativa utilidad para el desarrollo e implantación de la interoperabilidad en el Estado Dominicano. Por lo que, en este capítulo se exponen los puntos necesario para poder trabajar con la guía de la NORTIC A4:2014, como es el alcance de esta, los términos utilizados y las pautas para utilizar el catálogo de estándares interoperables.

SECCIÓN 1.01

ALCANCE

Las pautas de esta guía pueden ser aplicadas por todos los organismos pertenecientes al Poder Ejecutivo, ya sean centralizados, descentralizados, o embajadas, consulados y misiones en el extranjero para la implementación de la NORTIC A4:2014.

Entre los organismos centralizados se encuentran los ministerios y sus dependencias, así como los organismos con nivel de ministerios, viceministerios, organismos adscritos a la Presidencia de la República,



consejos y organismos asesores, direcciones generales, oficinas nacionales, procuradurías fiscales, escuelas públicas, hospitales públicos, bibliotecas y museos.

Entre los organismos descentralizados se encuentran los organismos financieros y no financieros, reguladores, seguridad social y empresas públicas.

Los organismos pertenecientes a los demás Poderes del Estado, así como aquellos que entran dentro de la clasificación de “Organismos Especiales”, según el Ministerio de Administración Pública (MAP), también pueden implementar las pautas indicadas en esta guía como un modelo de buenas prácticas, en apoyo a la estandarización del Estado Dominicano.

SECCIÓN 1.02 ¿CÓMO UTILIZAR ESTA GUÍA?

Cada uno de los capítulos de esta guía están didáctica y secuencialmente redactados para un eficiente seguimiento de su contenido. Básicamente la guía consta de las siguientes partes en su estructura:

- Los capítulos:
 - o Los capítulos de esta guía están estructurados de la siguiente forma:

Figura No. 1 Estructura de capítulos.



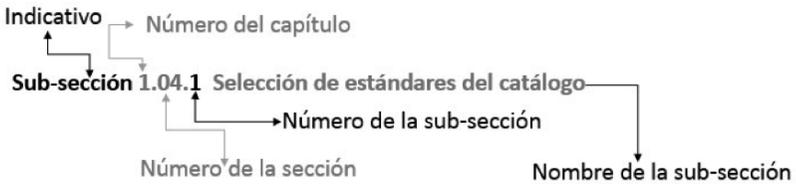
- Las secciones:
 - o Las secciones de esta guía están estructuradas de la siguiente forma:

Figura No. 2 Estructura de Sección.



- Las sub-secciones:
 - Las sub-secciones de esta guía están estructuradas de la siguiente forma:

Figura No. 3 Estructura de Sub-sección.



- Los apartados:
 - Los apartados de esta guía están estructurados de la siguiente forma:

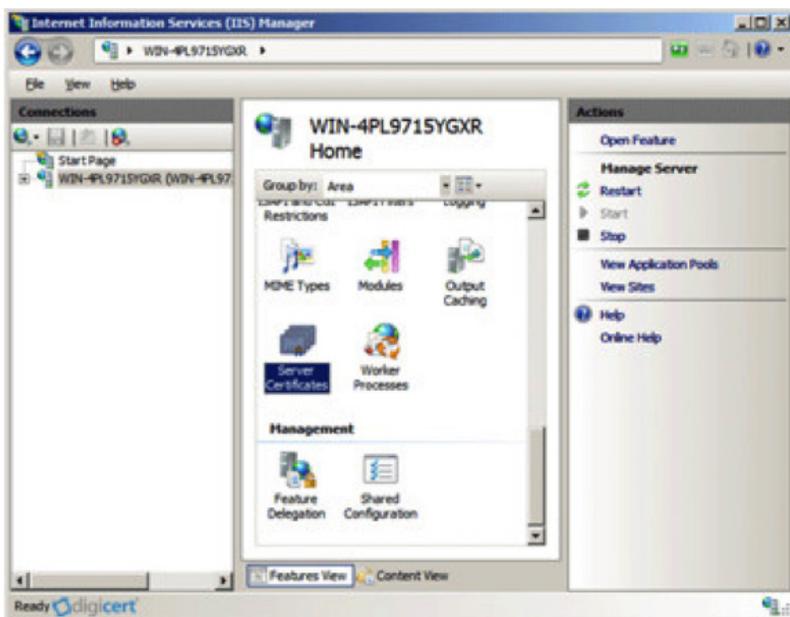
Figura No. 4 Estructura de Apartado.



Dentro de cada **capítulo**, **sección**, **sub-sección** o **apartado**, se puede encontrar los siguientes elementos:

- Las figuras:
 - Estas están presentadas de la siguiente forma:

Figura No. 5 Estructura de Figuras.



“En la parte “superior” se indica el número de la figura correspondiente, en conjunto con el desglose del nombre, y luego se presenta la figura como tal.”

- Las tablas:
 - Estas están presentadas de la siguiente forma:

Figura No. 6 Estructura de Tablas.

	SFTP	FTPS
Encriptación	SSL/TLS	SSH
Aplicación	Implícita/Explícita	Por Defecto

- Los procesos:
 - Cuando se describen o se realizan procesos, estos están numerados en orden descendente y en casos especiales con letras en orden alfabético, como se muestra a continuación:



3. crear el grupo (sftp) y usuario (opticusr)

Figura No. 7 Estructura de Procesos.

```
root@optic:~# groupadd sftp
root@optic:~# useradd -g sftp -s /bin/false -d /home/sftp/optic opticusr
root@optic:~# passwd redorbita
```

- Introduzca la nueva contraseña de UNIX.
- Vuelva a escribir la nueva contraseña de UNIX.
- passwd: contraseña actualizada correctamente.

“El tres “3” indica el número del paso correspondiente en el proceso, en conjunto con el desglose del paso a realizar, y luego se presenta la ejecución correspondiente, acompañada en casos particulares, con comandos⁽⁶⁾ o instrucciones como se muestra en el ejemplo más arriba.”

- Palabras claves o destacadas:
 - o Estas se utilizan para enfatizar y destacar parte del contenido.

Figura No. 8 Estructura de Palabras clave o destacadas.

Ejecutar el programa *FreeSSHd* con permisos de **Administrador** (presionar clic con el botón derecho del ratón en el icono de la aplicación y luego presionar clic en **Ejecutar como Administrado** para configurarlo adecuadamente.

Las mismas se presentan en “negrita” y en “cursiva”, en algunos caso se hace uso de comillas dobles (“ ”).

SECCIÓN 1.03

TÉRMINOS Y DEFINICIONES

Para fines de esta guía, el término “Organismo gubernamental” será utilizado en ciertos casos como “Organismo”.

(6) Son cada una de las sentencias u órdenes que a través de una interfaz el usuario le pasa al sistema con el fin de realizar una tarea o función específica.



Cuando aparezca el término “Catálogo” este hará referencia al “Catálogo de estándares” descrito en el capítulo II.

En el caso del término “Interorganizacional”, este hace referencia a cualquier actividad que se realiza de manera interna dentro del organismo. Por otro lado, el término “Intraorganizacional” hace referencia a cualquier actividad que se realiza en conjunto con otros organismos.

Además, los términos “Software⁽⁷⁾”, “Aplicaciones” y “Programas”, para fines de esta guía, se utilizarán indistintamente. Mientras, que “aplicación web” se refiere a toda aplicación creada con lenguajes y tecnologías estándares de Internet aprobados por la W3C⁽⁸⁾, entre ellos HTML⁽⁹⁾, Javascript y CSS⁽¹⁰⁾.

El término “Elemento de datos⁽¹¹⁾” hace referencia a una entidad de información relacionada dentro de un proceso de interoperabilidad.

SECCIÓN 1.04 USO DEL CATÁLOGO DE ESTÁNDARES

El uso adecuado del catálogo de estándares es fundamental para la creación de soluciones realmente interoperables, por ello en esta sección se proporcionan los pasos a seguir para obtener un estándar adecuado para su solución, así como también la petición de evaluación de un estándar que no se encuentre en el catálogo.

- (7) *Se refiere a todos los componentes lógicos o intangibles de un sistema de información, tales como programas, aplicaciones, sistemas operativos, entre otros.*
- (8) *Es una entidad internacional que produce pautas y recomendaciones para la usabilidad y accesibilidad de la web.*
- (9) *Es el lenguaje de programación utilizado para la creación de páginas web.*
- (10) *Es un lenguaje de programación para la web destinado a dar estilo visual.*
- (11) *Hace referencia a un valor íntegro sobre un elemento determinado, el cual por si solo carece de importancia y a través del procesamiento adecuado logra convertirse en información útil.*



Sub-sección 1.04.1

Selección de estándares del catálogo

Para la correcta selección de estándares del catálogo es importante que se tomen en consideración los siguientes criterios:

- Utilizar la última versión existente.
- Investigar casos de uso para conocer el estándar especializado y más adecuado para su solución.
- Tomar en cuenta si el estándar es de uso privado o abierto, ya que esto puede derivar en costos adicionales por licencia o simplemente no poder acceder a las funcionalidades que se necesita.
- A integrar un estándar con una solución existente o empaquetada, debe investigarse la documentación necesaria para obtener la certeza de que dicho estándar esta optimizado para trabajar con la solución.
- Si el estándar seleccionado se utiliza para interacciones con organismos internacionales, debe verificarse que el uso del mismo este permitido por las políticas de uso de dicho país.

Sub-sección 1.04.2

Uso de estándares no presentes en la normativa

En caso de utilizar un estándar que no se encuentre especificado en el catálogo presentado en la NORTIC A4:2014, debe enviarse la justificación de uso al correo electrónico enat@optic.gob.do para fines de evaluación del requerimiento. El formato a utilizar es el siguiente:

[Saludos Sres. de OPTIC]

Mediante el presente les hacemos la solicitud de evaluación del estándar con las siguientes características:



Tabla No. 1 Levantamiento de información

METADATO	VALOR
Nombre	
Fecha	
Descripción	
Organismo	
Formato	
Valores permitidos	
Identificador	
Estado	
Tipo	
Versión	
Alias	
Validación	

(Para completar esta tabla y sus valores se puede utilizar el modelo especificado en la **Sección 3.02 sobre Metadatos**).

La justificación del uso de este estándar está dada por las siguientes razones:

[Especificar listado de las razones por las cuales se seleccionó el estándar].

[Gracias.]

CAPÍTULO II

IMPLEMENTACIÓN DE LA INTEROPERABILIDAD ORGANIZACIONAL

En este capítulo se presentan los acuerdos organizacionales, los cuales están compuestos por los siguientes tipos: acuerdos técnicos, semánticos, organizacionales, legales, económicos y culturales. Los puntos citados en el mismo se enfocan en temas para lograr una buena gestión de procesos organizacionales así como también una buena gestión para los proyectos de interoperabilidad.

SECCIÓN 2.01 ACUERDOS ORGANIZACIONALES

Los acuerdos organizacionales forman parte del pilar central de la implementación de la interoperabilidad, tanto dentro de un mismo organismo como fuera ya sea en el mismo país u otros. Es sencillamente la base que deja por escrito todas las reglas del juego en los distintos aspectos descritos a continuación, los cuales se han agrupado en 6 tipos de acuerdos.

Sub-sección 2.01.1

Tipos de acuerdos

Acuerdos técnicos: En estos acuerdos se incluyen protocolos de comunicación, sistema de seguridad, así como la descripción y definición de los estándares seleccionados. Estos deben definirse tomando en cuenta las siguientes indicaciones:

- Se recomienda definir los protocolos describiendo los



siguientes campos: Nombre del protocolo, Alias o Nombre Abreviado, versión (en caso de que esté disponible).

- Se recomienda seleccionar qué campos incluir en las transacciones de los sistemas, que garanticen la correcta y posterior auditoría. Se recomienda que el formato de referencia sea: Nombre del campo, Tipo de dato, Entidades del sistema donde estará presente.
- Hacer una selección de estándares adecuadamente, tal como lo expresa la **Sección 1.04 Uso del Catálogo de Estándares**.

Acuerdos semánticos: En estos acuerdos se incluyen los datos, metadatos e informaciones para una interpretación correcta de lo que se está intercambiando. Se recomienda definirlos de la siguiente forma:

- Los estándares seleccionados deben ser seleccionados bajo el formato y criterios descritos en la **Sección 3.02 Documentación de Metadatos**.

Acuerdos organizacionales: En estos acuerdos se incluyen las bases y las reglas de acceso y caducidad de la información, servicios sobre las conexiones, autenticación, privacidad, seguridad de la información, entre otros.

- Para la definición de los sistemas de seguridad se recomienda separarlos en 2 sub-tipos:
 - **Autenticación:** debe especificarse si este estará “integrado”, lo cual es muy común en sistemas operativos windows (Windows Autenticación) o si es autenticación propia de la solución tecnológica utilizada (Form Authentication).



- o **Comunicación:** definir protocolos de comunicación y transferencia de datos como los expuestos en la **Sección 4.02 Protocolos de intercambio de datos.**

Acuerdos legales: Para estos acuerdos se requiere el uso de una política claramente definida, basada y sustentada en la creación, determinación de especificaciones y reglas dentro de un marco legalmente viable y flexible que permita agilizar la implementación fluida de la interoperabilidad gubernamental. A continuación se presentan algunos puntos a tomar en cuenta como guía para lograr estos acuerdos:

- Seleccionar de forma eficiente, evaluando perfiles y posiciones actuales, las personas con influencia política en la institución o instituciones afectadas en el proceso.
- Evaluar las leyes existentes o normativas que puedan estar relacionadas o puedan afectar la gestión de la interoperabilidad entre los sistemas.
- Si se trata de un organismo internacional tiene que evaluarse cualquier tratado o convenio internacional entre los países sobre el marco de interoperabilidad y confidencialidad de la información.
- Por último, debe continuar con la gestión organizacional como lo describe la **Sección 2.02 Gestión de procesos organizacionales.**

Acuerdos económicos: Estos acuerdos consisten en la forma de hacer sustentable la implementación de un sistema interoperable, en el cual se incluyen los acuerdos de cooperación, formas de pago, servicios con costos, entre otros. Se recomienda definirlos tomando en cuenta lo siguientes aspectos:



- Determinar si la implementación llevará algún costo, y de ser así, crear un presupuesto para tales fines.
- Si se determina la inclusión de organizaciones con o sin fines de lucro para trabajar en el proceso, estas deben ser listadas en una tabla, con campos presentes como: Nombre de la organización, con o sin fines de lucro, actividades a realizar y presupuesto asignado.
- Se recomienda trabajar el proceso de implementación de interoperabilidad como un proyecto, donde se contemplen tareas, responsables, tiempos y metas, para de esta forma mantener un costo lo menos variante posible y mantenerse apegado al presupuesto inicial.

Acuerdos culturales: Estos son acuerdos intraorganizacionales, los cuales refieren a los mecanismos que se utilizarán para generar una cultura organizacional basada en el modelo de sistema de información interoperable. Se recomienda que estos se definan utilizando las siguientes recomendaciones:

- Gestionar de forma constante la difusión de la NORTIC A4:2014 de interoperabilidad y de la presente guía de implementación.
- Crear un comité encargado de la supervisión del cumplimiento de las normas de interoperabilidad, tanto a nivel de los proyectos como en la parte técnica, para monitorear y mantener una evaluación constante del proceso de desarrollo de los proyectos.
- Colaborar con otros organismos que estén implantando interoperabilidad.



SECCIÓN 2.02

GESTIÓN DE PROCESOS ORGANIZACIONALES

En todo organismo debe existir una cultura y estructura organizativa que garantice la eficiencia en el desarrollo de las actividades del organismo, esto es fundamental y sirve como base sólida para crear una correcta implementación de la interoperabilidad de forma intraorganizacional como también la interorganizacional.

Sub-sección 2.02.1 Objetivos para la gestión de procesos

Para una gestión de procesos organizacionales eficiente, es necesario enfocarse en los siguientes temas y su aplicación.

- Determinación del uso de las TIC en el organismo definidos por 3 simples criterios:
 - **Frecuencia:** uso de los sistemas informáticos por parte de los usuarios de forma recurrente.
 - **Habilidad:** basándose en la capacidad y habilidad de interacción de los usuarios con las tecnologías presentes en el organismo.
 - **Equipos tecnológicos:** total de equipos tecnológicos por cantidad de usuarios.

Para una correcta medición de estos aspectos, puede establecerse niveles de puntuación (según el criterio del organismo), los cuales deben dar como resultado un valor superior al 60% aproximadamente para tomarse como nivel aceptable, (un ejemplo de esto sería, que si el rango de puntuación es del 1 al 5 el valor mínimo aceptable sería 3, en el caso de ser sobre 100%). Todos los criterios deben tener una relevancia equitativa.



- Manejo de las iniciativas basados en los criterios de la NORTIC A4:2014:

- **Iniciativa individual:** Es aquella que solo se genera en un solo organismo de forma única.
- **Iniciativa común:** Es aquella que se genera entre varios organismos basadas en necesidades similares.

El formato recomendado para manejar estas iniciativas debe contener las siguientes informaciones: Título de la Iniciativa, Fecha, Organismos y Descripción.

- Mecanismo de difusión y propagación de la cultura de interoperabilidad:

- El departamento encargado para el manejo de interoperabilidad en el organismo puede realizar seminarios internos dentro de la institución que propicien la importancia de la misma.
- Gestionar entrenamientos para el personal técnico y de proyecto con relación a la implementación y conocimiento de la NORTIC A4:2014.
- Presentar los avances de los proyectos donde exista un impacto significativo en las ventajas que refiere a la interoperabilidad con otros organismos tanto nacionales como internacionales.
- Enviar la NORTIC A4:2014 y la presente guía de implementación a todo el personal interesado dentro del organismo.

La forma más rápida de implementar la cultura de interoperabilidad en un organismo es utilizando lo que llamamos el “apalancamiento”, el cual consiste en buscar el apoyo e influencia de los altos mandos o personas con poder de decisión. Pero para lograr dicho apoyo debemos mantenerlos informados de forma constante de la importancia que tienen la implementación de las normativas y los proyecto que se están llevando a cabo para estos fines.



- Cómo alcanzar mayores niveles de madurez:
 - Se recomienda realizar la implementación de un controlador de dominio, un ejemplo de esto es el “Servicio de directorio”, para centralizar la seguridad de los usuarios, de lo contrario el organismo puede tener implementadas herramientas que propicien la colaboración de información, tales como servidores de correo electrónico, portales de intranet con funcionalidades de colaboración de documentos, formularios digitales, flujos de trabajo y controles de acceso granulares.
 - Implementar tecnologías de intercambio de datos de forma remota.
 - Implementar servidores de bases de datos donde se encuentren organizadas las bases de datos y todas las aplicaciones del organismo.
 - Deben implementar un plan de desastres que garantice la protección de los datos en caso de pérdida o catástrofe.

El nivel mínimo que se espera alcanzar para comenzar un estrategia de interoperabilidad aceptable es el 2, para mayor conocimiento sobre niveles de madures véase en la normativa A4-2014, Anexo C. Niveles de madurez de la interoperabilidad.

SECCIÓN 2.03.

GESTIÓN DE PROYECTOS DE INTEROPERABILIDAD

Para lograr el éxito de todo proyecto este debe ir acompañado de una buena gestión. En la NORTIC A4:2014 sobre interoperabilidad, se hace referencia a la NORTICA1:2014, sección 3.01 sobre planificación de proyectos de TIC en donde se plantea el proceso a seguir para la



planificación de estos tipos de proyectos y los documentos requeridos para esto, por tal razón en esta sección se explica de manera detallada la forma de llevar a cabo estos procesos.

Sub-sección 2.03.1

Definición de roles

De contar con un departamento de proyectos, los roles de este pueden ser:

- Planificar los proyectos de interoperabilidad.
- Coordinar los proyectos de interoperabilidad.
- Administrar y dar el seguimiento correspondiente a los proyectos de interoperabilidad.
- Identificar y mitigar posibles riesgos que puedan afectar el cumplimiento de los proyectos de interoperabilidad.

Los roles correspondientes a cada persona por separado pueden definirse de la forma siguiente:

- **El líder del proyecto de interoperabilidad:** este tomará las decisiones pertinentes, tomando en cuenta el presupuesto asignado al proyecto, el tiempo de realización, y cómo el proyecto impactará, ya sea fuera o dentro del organismo.
- **El manejador del proyecto de interoperabilidad:** este velará por la calidad de los entregables, el cumplimiento del tiempo de los entregables, así como también sobre cómo se asignarán los recursos del proyecto.



- **El líder técnico del proyecto de interoperabilidad:** este coordinará el trabajo a realizar por parte del equipo técnico y mantendrá la supervisión del mismo.

Dependiendo de la naturaleza del proyecto y la cantidad de recursos humanos asignados, los roles pueden ser asumidos indistintamente por el personal, siempre y cuando estos sean cumplidos.

Sub-sección 2.03.2

Creación del acta de constitución del proyecto

El acta de constitución del proyecto es un documento en el cual se define el alcance, los objetivos y los participantes del proyecto. Presenta una visión preliminar de los roles y responsabilidades, de los objetivos, de los principales interesados y del presupuesto.

A continuación se presenta la información que debe contener el documento del acta de constitución del proyecto:

- **Nombre del proyecto:** Nombre del Proyecto.
- **Código del proyecto:** Código único del proyecto. Se recomienda establecer una nomenclatura de codificación. Ejemplo: OPT-001.
- **Partes involucradas en el proyecto:** integrantes o actores involucrados directamente con el proyecto.
- **Director del proyecto y su nivel de autoridad dentro del proyecto:** Persona que tiene la autoridad de firmar el cierre del proyecto y la aceptación final del mismo.



- **Descripción del proyecto:** información general del proyecto donde se contemple los objetivos dentro del contexto.
- **Requisitos del proyecto:** Listado de requerimientos necesarios para que el proyecto se lleve a cabo.
- **Criterios de aceptación del proyecto:** listado de aseveraciones que deben cumplirse para dar el proyecto por terminado de forma satisfactoria.
- **Riesgos del proyecto:** detalles de cada una de las posibles situaciones, las cuales puedan colocar al proyecto en una situación de riesgo o de incumplimiento de sus actividades y/o tiempos establecidos.
- **Objetivos generales y específicos del proyecto:** El objetivo general debe expresar la esencia y visión real del proyecto, el cual deriva en su finalidad. Los objetivos específicos, se derivan del objetivo general y lo detallan. Indican los efectos específicos que se quieren conseguir aunque en estos no se explican las acciones a tomar.
- Método de escalamiento de comunicación, el cual se define por los siguientes niveles:
 - **Nivel 1:** Persona designada por el director del proyecto.
 - **Nivel 2:** Persona encargada de la dirección del proyecto.
 - **Nivel 3:** Director general del organismo.

Para mayor información de como laborar el acta de constitución , ver el Anexo A. Acta de constitución.



Sub-sección 2.03.3

Elaboración del diagrama de planificación del proyecto

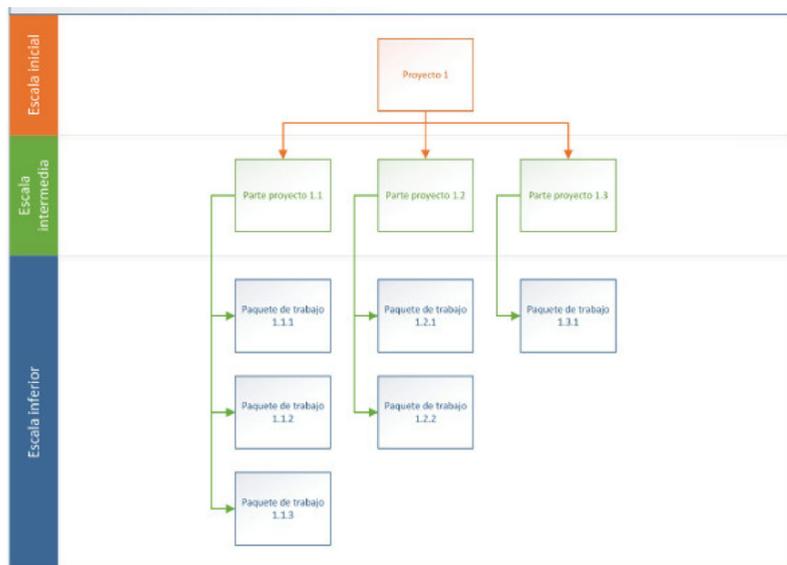
El diagrama de planificación de proyecto contiene 3 partes, las cuales se presentan de la manera siguiente:

- Una **escala inicial**, en donde se especifique el nombre del proyecto y el código.
- Una **escala intermedia**, en la cual se muestren las principales partes o componentes del proyecto.
- Una **escala inferior**, en donde se establecen los conjuntos de trabajo, en los cuales se especifiquen los recursos, el tiempo y las estimaciones de los costos, de cada uno de los componentes o partes del proyecto.

Estos conjuntos de trabajo deben ser asignados a los diferentes miembros del equipo o unidades organizativas para la realización de las actividades.

En la **Figura No. 9 Escalas de proyectos** Se muestra detalladamente cómo sería la forma ejemplar de un Diagrama de Planificación del proyecto:

Figura No. 9 Escalas de proyectos.



Sub-sección 2.03.4

Elaboración del documento de cierre del proyecto

El documento de cierre del proyecto es un escrito en el cual se destacan los aspectos necesarios para la aceptación, tanto del manejador del proyecto como de la persona responsable o encargada del proyecto. Este documento llevará la siguiente información:

- **Nombre del proyecto:** nombre del Proyecto.
- **Código del proyecto:** código único del proyecto. Se recomienda establecer una nomenclatura de codificación. Ejemplo: OPT-001.



- **Partes involucradas en el proyecto:** integrantes o actores involucrados directamente con el proyecto.
- **Director del proyecto y su nivel de autoridad dentro del proyecto:** persona que tiene la autoridad de firmar el cierre del proyecto y la aceptación final del mismo.
- **Descripción del proyecto:** información general del proyecto donde se contemplen los objetivos dentro del contexto.
- **Información Objetivos Logrados/ Acuerdo de Cierre (Criterios de aceptación):** listado de aserciones que deben cumplirse para dar el proyecto por terminado de forma satisfactoria.

Para mayor información de cómo constituir el acta de cierre, ver el Anexo B. Acta de cierre.

CAPÍTULO III

IMPLEMENTACIÓN DE LA INTEROPERABILIDAD SEMÁNTICA

Para que la interoperabilidad cumpla su función de manera correcta, debe abarcarse cada aspecto que esta envuelve. Si no hay una correcta visualización y entendimiento de la información que se comparte, puede asegurarse que no tendrá éxito dicha interoperabilidad. Estos aspectos se abarcan en el capítulo IV de la NORTIC A4:2014 sobre interoperabilidad semántica y dada la importancia que estos presentan, se detallan en este capítulo de la guía de implementación.

SECCIÓN 3.01 DOCUMENTACIÓN DE VISUALIZACIÓN DE LA INFORMACIÓN

La interoperabilidad semántica para la visualización expuesta en el Capítulo IV de la NORTIC A4:2014 expresa que debe tenerse una forma de documentar el objetivo visual de un campo o atributo, así como esquematizar el contenido y comportamiento de la página o aplicación, de esta forma se concreta el objetivo de este tipo de interoperabilidad, que nos es más que la estandarización del producto visual de la solución para su posterior uso en el desarrollo de la misma.

El idioma utilizado en las imágenes que aparecen en las explicaciones de los procesos en esta guía ha sido el inglés, esto debido a que se seleccionó el idioma estándar utilizado a nivel internacional en cuanto a herramientas y procesos tecnológicos se refiere para mayor entendimiento del lector.



Sub-Sección 3.01.1

Especificación de formato de visualización de la información

La percepción es la realidad de todos y en tecnología no es una excepción, por ello los factores visuales y de representación de la información son uno de los factores principales para implementar interoperabilidad para la visualización entre sistemas es simplemente tomar en cuenta la especificación de cómo el usuario final observará el atributo o campo en la aplicación web, para ello este debe aparecer en el metadato “Formato” como una parte del “Valor”, ejemplo:

Tabla No. 2 Referencia de metadatos.

NOMBRE:	FORMATO:	VISUALIZACIÓN:
Dirección	Tipo de dato: Caracteres variantes Longitud mínima: 0 Longitud máxima: 200	Caja de texto

Sub-Sección 3.01.2

Diseño de presentación de datos para dispositivos

Es importante tomar en cuenta para toda aplicación web realizar la documentación necesaria de diseño, en donde se pueda visualizar el comportamiento de tipo “responsivo”, con la finalidad de facilitar la adaptación a cualquier dispositivo que la consume, para ello se debe utilizar la técnica de documentación más profesional y eficiente para cumplir con el objetivo, la cual se encuentra basada en los llamados “Modelos Gráficos”.



Sub-Sección 3.01.3

Documentación con esquemas de páginas

Un esquema de página, es un modelo visual donde se representa la estructura de un portal o aplicación web, el mismo esquematiza un diseño u ordenamiento contemplando el contenido y el comportamiento visual, donde se incluye los elementos y sistemas de navegación. Estos esquemas carecen de estilo tipográfico, color o aplicaciones gráficas, el objetivo es mostrar visualmente las funcionalidades, el comportamiento y la jerarquía de los contenidos.

Dichos esquemas están enfocados en:

- Tipo de contenido que será mostrado.
- Funcionalidades disponibles.
- Prioridades relativas de la información y las funcionalidades.
- Las reglas para mostrar ciertos tipos de información.
- Visualización de los distintos modelos o escenarios en la pantalla.

El proceso iterativo de creación de esquemas de páginas, es una forma efectiva de hacer prototipos en el menor tiempo posible, midiendo la viabilidad y eficiencia de un concepto de diseño.

Ejemplos de esquemas de página son los siguientes:

Figura No. 10 Esquema de página 1.

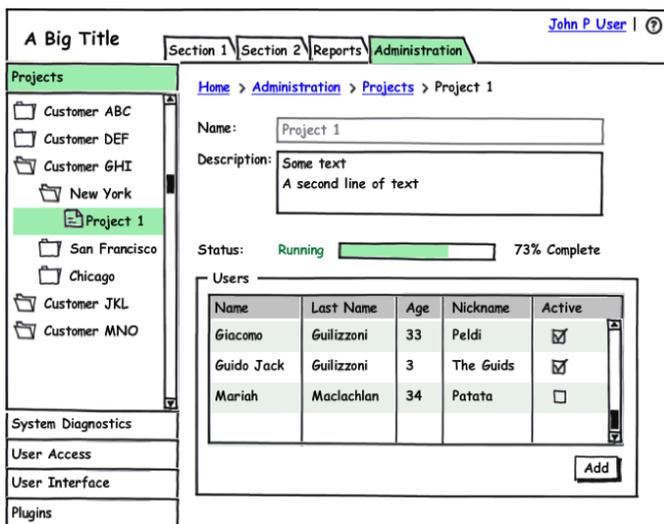


Figura No. 11 Esquema de página 2.

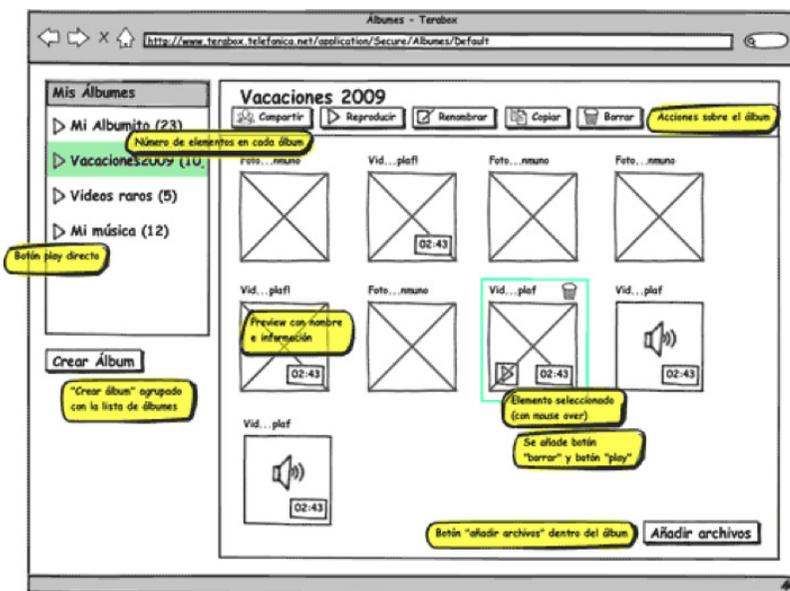
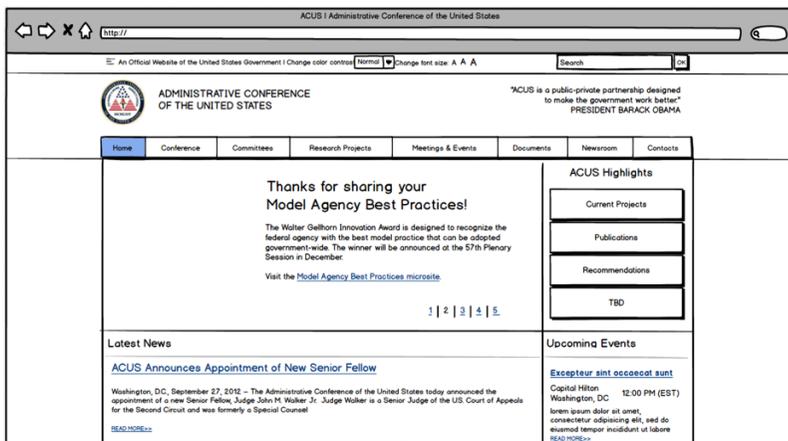




Figura No. 12 Esquema de página 3.



Los esquemas son la forma recomendada para la generación de la documentación necesaria para la correcta implementación de la interoperabilidad semántica para la visualización, así como la expresa en sección 4.01 sobre Interoperabilidad semántica para la visualización la NORTIC A4:2014.

SECCIÓN 3.02

METADATOS

En esta sección, se describe la práctica de los tipos de documentación según su alcance en referencia a la exposición que tendrán los componentes e interfaces que se desarrollen. Los tipos de documentación se dividen de la siguiente manera: documentación interna de metadatos y documentación externa de metadatos.

Sub-sección 3.02.1 Documentación interna de metadatos

La documentación interna de metadatos es la que se utiliza para describir sistemas o aplicaciones de uso interno en el organismo, como lo explica la NORTIC A4:2014, puede tomarse en cuenta el siguiente formato:





Tabla No. 3 Gráfica de documentación interna de metadatos.

METADATO	VALOR
NOMBRE	Dirección.
FECHA	3/3/2015
DESCRIPCIÓN	Atributo que refiere a la dirección física de una persona o negocio en el formulario de envío de correspondencia.
ORGANISMO	Oficina Presidencial de Tecnologías de la Información y Comunicación (OPTIC).
FORMATO	Tipo de dato: Carácter variante. Longitud mínima: 0. Longitud máxima: 200. Visualización: Caja de texto.
VALORES PERMITIDOS	Solo caracteres.
IDENTIFICADOR	chrDir.
ESTADO	Disponible.
TIPO	Campo de información.
VERSIÓN	1.0
ALIAS	chrDireccion.
VALIDACIÓN	No disponible.

Sub-sección 3.02.2 Documentación externa de metadatos

La documentación externa o pública es aquella que se realiza para crear una fuente de datos que será consumida por un gran número de aplicaciones de diferentes organizaciones, empresas o público a nivel general, por ello se utiliza el marco de descripción de recursos, conocido como Vocabulario para Catálogo de Datos (DCAT⁽¹²⁾, por sus siglas en inglés), el cual es un esquema internacionalmente reconocido por la W3C especialmente para gobierno electrónico. Este tipo de documentación de metadatos es fundamental para la creación

(12) Es un estándar definido por el Consorcio World Wide Web (W3C) y diseñado para facilitar la interoperabilidad entre catálogos de datos publicados en la web.



y exposición de plataformas de datos abiertos⁽¹³⁾, las cuales son el presente y futuro de la implementación del gobierno electrónico.

No se indica ningún método de implantación, ni de utilización de los datos expresados en DCAT, ya que podrían ser utilizados por diversos medios como SPARQL⁽¹⁴⁾ endpoints, RDFa⁽¹⁵⁾, RDF⁽¹⁶⁾/XML⁽¹⁷⁾ o Turtle.

Los ejemplos que se incluyen utilizan la notación Turtle⁽¹⁸⁾, simplemente por su facilidad en la interpretación.

Sub-sección 3.02.3

Vocabularios externos a utilizar

En esta sub-sección se detalla una serie de clases y propiedades, que componen el vocabulario externo. Se ha intentado componer este utilizando otros esquemas y estándares ya existentes.

No se debería utilizar valores como “desconocido” o “no aplica” para las propiedades, incluso si en la fuente original apareciera especificado. En su lugar la propiedad simplemente sería omitida.

A continuación se muestra un listado con los vocabularios externos utilizados para componer DCAT, con los prefijos que se utilizan en el documento para abreviar los espacios de nombres correspondientes.

(13) Son datos que están disponibles para cualquier persona sin restricciones, los cuales pueden ser utilizados, reutilizados y redistribuidos libremente.

(14) Es un lenguaje de consulta para grafos estandarizados en RDF.

(15) Es una técnica que permite proporcionar un conjunto de atributos de marcas para aumentar la información visual en la web.

(16) Es un modelo estándar del Consorcio World Wide Web (W3C), diseñado para almacenar datos en forma legible e intercambio de datos en la web.

(17) Es un lenguaje desarrollado por el Consorcio World Wide Web (W3C) para almacenar datos en forma legible y utilizada para el intercambio de información entre diferentes plataformas.

(18) Conocido también como Lenguaje de Notación 3 o “N3”, es un lenguaje utilizado para sintaxis XML de RDF.

Tabla No. 4 Vocabularios externos

VOCABULARIO	PREFIJO	ESPACIO DE NOMBRE
Sintaxis RDF	rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
Esquema RDF	rdfs	http://www.w3.org/2000/01/rdf-schema#
Esquema XML	xsd	http://www.w3.org/2001/XMLSchema#
SKOS	skos	http://www.w3.org/2004/02/skos/core#
Términos de metadatos DCMI	dct	http://purl.org/dc/terms/
Tipos de DCMI	dctype	http://purl.org/dc/dcmitype/
Amigo de un amigo (FOAF)	foaf	http://xmlns.com/foaf/0.1/
Vocabulario Anular (void)	void	http://rdfs.org/ns/void#
Vocabulario para Catálogo de Datos	dcat	http://vocab.deri.ie/dcat#

A continuación se muestra un listado con las clases y sus propiedades:

1. **Clase catálogo:** Un catálogo de datos es una colección de metadatos sobre un conjunto de datos. El mismo está compuesto de la siguiente manera:
 - Clase RDF: dcat:Catalog.
 - Subclase RDF: dcat:Collection.
 - Nota: Típicamente, un catálogo de datos basado en web se presenta como una instancia de la clase.
 - También puede verse: Registro de Catálogo, Clase: Dataset.

Dentro de esta clase se encuentran las siguientes propiedades:

- a) **Página web principal:** esta es la página del catálogo y se encuentra compuesta por los siguientes metadatos:
 - Propiedad RDF: foaf:homepage.



- Rango: foaf:document.
 - Nota: foaf:homepage, esta es una propiedad funcional inversa, lo que significa que debería ser única e identificar de forma precisa el catálogo. Esto permite especificar varias descripciones del catálogo cuando se usan diferentes URI⁽¹⁹⁾.
- b) **Agente editor:** este es el responsable de publicar el catálogo en la web y se encuentra compuesto por los siguientes metadatos:
- Propiedad RDF: dct:publisher.
 - Rango: foaf:agent.
- c) **Cobertura espacial:** esta es el área geográfica que se encuentra cubierta por el catálogo y se encuentra compuesta por los siguientes metadatos:
- Propiedad RDF: dct:spatial.
 - Rango: dct:location (Región espacial o lugar determinado).
- d) **Tema o categoría:** esta propiedad de Valores del Sistema de Organización (KOS, por sus siglas en inglés), se utiliza para clasificar los conjuntos de datos del catálogo. Se encuentra compuesta por los siguientes metadatos:
- Propiedad RDF: dcat:themeTaxonomy.
 - Rango: skos:ConceptScheme.

(19) Es una dirección exacta y precisa que permite ubicar un recurso en el internet o en una red de cómputos.



- e) **Título:** esta propiedad es la que se utiliza para darle los diferentes nombres al catálogo y está compuesta por los siguientes metadatos:
- Propiedad RDF: dct:title.
 - Rango: rdfs:Literal.
- f) **Descripción:** esta propiedad se utiliza para describir en texto libre el catálogo, la misma se encuentra compuesta por los siguientes metadatos:
- Propiedad RDF: dct:description.
 - Rango: rdfs:Literal.
- g) **Licencia:** esta es la propiedad bajo la cual, el catálogo puede ser usado y reutilizado, no los conjuntos de datos que la componen. Se encuentra compuesta por los siguientes metadatos:
- Propiedad RDF: dct:license
 - Rango: dctype:LicenseDocument
 - Nota de uso: para permitir el análisis automático de los conjuntos de datos es importante usar identificadores canónicos para licencias conocidas. Ver también la Propiedad: licencia de la clase Dataset.
- h) **Dataset:** esta propiedad es un conjunto de datos que componen el catálogo. Se encuentra compuesta por los siguientes metadatos:
- Propiedad RDF: dcat:dataset.



- Rango: dcat:Dataset.
- i) **Registro de catálogo:** esta propiedad es un registro de datos que componen el catálogo. Se encuentra compuesta por los siguientes metadatos:
- Propiedad RDF: dcat:record.
 - Rango: dcat:CatalogRecord.
2. **Clase de registro del catálogo:** esta clase es un registro en el catálogo de datos, que describe un único conjunto de datos. Se encuentra compuesta de la siguiente manera:
- Clase RDF: dcat:CatalogRecord.
 - Nota de uso: esta clase es opcional y no todos los catálogos la utilizan.
 - Ver también: Clase: Dataset.

Si se utilizan grafos⁽²⁰⁾ con un nombre, las tripletas RDF que describen la entrada del catálogo, el dataset y sus distribuciones deberían estar incluidas en grafos cuyo identificador sea el URI del registro del catálogo.

En los catálogos basados en web, la URI de la página web del catálogo se utiliza como una URI para el registro del catálogo si este es un enlace permanente.

Dentro de esta clase se encuentran las siguientes propiedades:

(20) Son un conjunto de elementos enlazados entre sí mediante arcos.



- a) **Fecha de publicación en el catálogo:** esta propiedad se utiliza para colocar la fecha en la que se añade el conjunto de datos al catálogo. Se encuentra compuesta por los siguientes metadatos:
- Propiedad RDF: dct:issued.
 - Rango: rdfs:Literal con el tipo xsd:date, usando el formato de fechas y horas de la ISO 86013. Básicamente, la fecha se codifica como un literal según el formato “YYYY-MM-DD”. Si no se conoce un día o mes específico, se indicará 01.
 - Nota de uso: indica la fecha de listado del conjunto de datos en el catálogo, no la fecha de publicación en sí.
 - Ver también propiedad: fecha de lanzamiento del conjunto de datos.
- b) **Fecha de actualización o modificación:** esta propiedad es conocida por la fecha en que se actualizó o modificó la entrada del catálogo. Se encuentra compuesta por los siguientes metadatos:
- Propiedad RDF: dct:modified.
 - Rango: rdfs:Literal con el tipo xsd:date, usando el formato de fechas y horas de la ISO 8601. Básicamente, la fecha se codifica como literal según el formato “YYYY-MM-DD”. Si no se conoce el día o mes específico, se indicará 01.
 - Nota de uso: esta propiedad indica la fecha de último cambio de una entrada del catálogo, por ejemplo, los metadatos de la descripción del conjunto de datos, no la fecha de cambio del mismo.



c) **Conjunto de datos (dataset):** esta propiedad enlaza el registro del catálogo con el recurso del conjunto de datos (dcat:Dataset) descrito en el registro. La misma se encuentra compuesta por los siguientes metadatos:

- Propiedad RDF: foaf:primaryTopic.
- Rango: dcat:Dataset.

3. **Clase dataset:** esta clase es una colección de datos publicados o almacenados en una única fuente y disponible para el acceso o descarga en uno o más formatos. La misma se encuentra compuesta de la siguiente manera:

- Clase RDF: dcat:Dataset.
- Nota de uso: esta clase representa al actual conjunto de datos publicado por el agente publicador. En casos donde se establece una distinción entre el actual conjunto de datos y su entrada en el catálogo, es necesario utilizar la clase registro del catálogo (dcat:CatalogRecord) para el segundo caso.
- Ver también Clase: Registro del catálogo.

Dentro de esta clase se encuentran las siguientes propiedades:

- a) **Fecha de actualización o modificación:** Como se menciona en la clase de registro del catálogo, punto b. fecha de actualización o modificación.
- b) **Título:** Como se menciona en la clase catálogo, punto e. Título.
- c) **Descripción:** Como se menciona en la clase catálogo, punto f. Descripción.



- d) **Agente editor:** esta propiedad se hace responsable de realizar el conjunto de datos disponibles, este está compuesto por los siguientes metadatos:
- Propiedad RDF: dct:publisher.
 - Rango: foaf:Organization.
 - Ver también: Clase: organización/persona.
- e) **Fecha de Lanzamiento:** Como se menciona en la clase de registro del catálogo, punto a. Fecha de publicación en el catálogo.
- f) **Frecuencia:** esta se utiliza para ver la frecuencia con la que se publica el conjunto de datos. Se encuentra compuesta por los siguientes metadatos:
- Propiedad RDF: dct:accrualPeriodicity.
 - Rango: dct:Frequency (frecuencia con lo que algo ocurre).
 - Dominio: dct:Collection.
- g) **Identificador:** esta propiedad es un identificador único del conjunto de datos, la cual está compuesta por los siguientes metadatos:
- Propiedad RDF: dct:identifier.
 - Rango: rdfs:Literal.
 - Nota de uso: el identificador debe ser utilizado para crear la URI permanente y única para el conjunto de datos, aunque puede ser útil representarlo en esta propiedad.



- h) **Cobertura espacial:** esta propiedad es la ubicación geográfica del conjunto de datos. Se encuentra compuesta por los siguientes metadatos:
- Propiedad RDF: dct:spatial.
 - Rango: dct:Location (Región espacial o lugar).
 - Nota de uso: debería utilizarse un vocabulario controlado.
- i) **Cobertura temporal:** este es el periodo temporal que cubre el conjunto de datos. Se encuentra compuesta por los siguientes metadatos:
- Propiedad RDF: dct:temporal.
 - Rango: dct:PeriodOfTime (Intervalo de tiempo definido por su inicio y final).
- j) **Licencia:** Como se menciona en la Clase Catálogo, punto g. Licencia.
- Nota de uso: usando identificadores canónicos de licencias conocidas se permitirá el análisis automático de los conjunto de datos.
- k) **Nivel de detalle:** esta propiedad describe el detalle de los conjunto de datos y se encuentra compuesta por los siguientes metadatos:
- Propiedad RDF: dcat:granularity.
 - Rango: rdfs:Resource.



- Nota de uso: el detalle es generalmente geográfico o temporal, pero también puede tener otra dimensión.
- l) **Diccionario de datos:** esta propiedad ofrece información descriptiva que ayuda a la comprensión de los datos. Suele consistir en una tabla con explicación sobre significados, interpretación de valores, y abreviaturas o códigos usados en los datos. Se encuentra compuesto por los siguientes metadatos:
- Propiedad RDF: `dc:terms`.
 - Rango: `foaf:Document`.
 - Nota de uso: no tiene un uso consistente, sin embargo se puede enlazar a un documento relacionado con el conjunto de datos. Se recomienda representarlo mediante una URI de un documento web.
- m) **Calidad de los datos:** esta propiedad describe la calidad en que se encuentran los datos. Se encuentra compuesto por los siguientes metadatos:
- Propiedad RDF: `dc:terms`.
 - Nota de uso: se espera que los catálogos definan sub-propiedades más específicas para definir las características.
- n) **Tema:** esta propiedad es la categoría del principal del conjunto de datos y este puede tener múltiples temas. Se encuentra compuesto por los siguientes metadatos:
- Propiedad RDF: `dc:terms`.
 - Rango: `skos:Concept`.



- Nota de uso: se utilizan conjuntos de conceptos skos:Concept organizados en esquemas de conceptos (skos:ConceptScheme) para categorizar los conjunto de datos.
- o) **Palabra clave:** esta propiedad utiliza una etiqueta o palabra clave para describir el conjunto de datos. Se encuentra compuesto por los siguientes metadatos:
- Propiedad RDF: dcat:keyword.
 - Rango: rdfs:Literal.
- p) **Documentos relacionados:** esta propiedad relaciona un documento con el conjunto de datos. Se encuentra compuesto por los siguientes metadatos:
- Propiedad RDF: dct:references
 - Nota de uso: se debería incluir la URI del documento relacionado.
- q) **Distribución de conjunto de datos:** esta propiedad conecta un conjunto de datos con sus distribuciones disponibles. Se encuentra compuesto por los siguientes metadatos:
- Propiedad RDF: dcat:distribution.
 - Rango: dcat:Distribution.
4. **Clase distribución:** esta representa una forma específica de un conjunto de datos. Cada conjunto de datos puede tener distintas formas, que pueden representar diferentes formatos del mismo y diferentes puntos de consulta. Se encuentra compuesta de la siguiente manera:



- Clase RDF: `dc:Distribution`.
- Nota de uso: mediante esta clase se representa la disponibilidad general de un dataset, pero no indica información sobre el método de acceso al dato en concreto (descarga directa, API⁽²¹⁾ o página web).

Dentro de esta clase se encuentran las siguientes propiedades:

- a) **URL de Acceso:** Esta propiedad apunta a la localización de una distribución⁽²²⁾. Esta puede ser un enlace de descarga directo, un enlace a una página HTML que incluye el enlace al dato actual, un canal RSS, un servicio web, etc. La semántica del tipo de acceso se determina por el dominio de la URL⁽²³⁾. Se encuentra compuesto por los siguientes metadatos:

- Propiedad RDF: `dc:accessURL`.
- Rango: Literal.
- Nota de uso: el valor es una URL.

- b) **Tamaño:** Esta propiedad indica el tamaño de una distribución de conjunto de datos. Se encuentra compuesto por los siguientes metadatos:

- Propiedad RDF: `dc:size`.

(21) Son los procedimientos, funciones o servicios que se encuentran disponibles en determinadas librerías y/o bibliotecas para ser utilizados por otro software.

(22) Son cada una de las diferentes versiones funcionales que se desarrollan de un sistema.

(23) Se utiliza para especificar la dirección exacta de un recurso dentro del portal web.



- Rango: `rdfs:Resource`.
 - Nota de uso: se usa generalmente con un nodo en blanco descrito usando las propiedades `rdfs:label` y `dcat:bytes`, que indica el tamaño.
- c) **Formato:** Esta propiedad indica el formato de la distribución del conjunto de datos. Se encuentra compuesto por los siguientes metadatos:
- Propiedad RDF: `dct:format`
 - Rango: `dct:MediaTypeOrExtent`
 - Nota de uso: se utiliza valores de tipos MIME8.
5. **Clase descarga:** esta clase representa una distribución de un conjunto de datos que se puede descargar. Se encuentra compuesta de la siguiente manera:
- Clase RDF: `dcat:Download`.
 - Nota de uso: la propiedad `accessUrl` de una distribución `Download` debería ser un enlace directo al fichero de descarga.
6. **Clase servicio web:** esta clase representa un servicio web que permite el acceso a los datos de un conjunto de datos. Se encuentra compuesta de la siguiente manera:
- Clase RDF: `dcat:WebService`
 - Nota de uso: describe el servicio web usando `accessUrl`, `format` y `size`.



7. **Clase canal de sindicación:** esta clase representa la disposición de un conjunto de datos como un canal de sindicación RSS, Atom o similar. Se encuentra compuesta de la siguiente manera:
 - Clase RDF: dcat:Feed.
 - Nota de uso: describe el canal de sindicación usando accessUrl, format y size.

8. **Clase categoría y esquema de categorías:** en esta clase se utiliza el Sistema de Organización del Conocimiento⁽²⁴⁾ (KOS, por sus siglas en inglés), el cual representa los temas y categorías de los conjunto de datos del catálogo. Se encuentra compuesta de la siguiente manera:
 - Clase RDF: skos:ConceptScheme, skos:Concept
 - Nota de uso: es necesario usar o skos:inScheme o skos:topConceptOf en cada skos:Concept, de otra forma no sería posible determinar a qué esquema pertenece cada concepto.

9. **Clase organización y persona:** esta clase se utiliza para la organización del conjunto de datos. Se encuentra compuesta de la siguiente manera:
 - Clase RDF: foaf:Person para personas y foaf:Organization para agencias públicas u otras organizaciones.

(24) Este término engloba diferentes esquemas que permiten crear y organizar la información, y promover la administración del conocimiento.

CAPÍTULO IV

IMPLEMENTACIÓN DE LA INTEROPERABILIDAD TÉCNICA

En este capítulo se plantean las bases para lograr la interoperabilidad persé, donde los sistemas convergen en un solo idioma para entenderse uno al otro y comunicarse debidamente por medio de las tecnologías implementadas. Para esto se explican los métodos para las implementaciones de protocolos de comunicación y transferencia de datos entre sistemas, implementación de ambientes en diferentes sistemas operativos y demás datos técnicos imprescindibles para los fines propuestos en este capítulo.

SECCIÓN 4.01 PROTOCOLOS PARA EL INTERCAMBIO DE INFORMACIÓN E HIPERTEXTOS

Los protocolos de transferencia de datos, así como los de transferencia de hipertexto⁽²⁵⁾ son de suma importancia, ya que por medio a estos las aplicaciones pueden intercambiar datos mediante redes privadas⁽²⁶⁾ e Internet.

En primera instancia el Protocolo para Transferencia de Datos e Hipertexto

El idioma utilizado en las imágenes que aparecen en las explicaciones de los procesos en esta guía ha sido el inglés, esto debido a que se seleccionó el idioma estándar utilizado a nivel internacional en cuanto a herramientas y procesos tecnológicos se refiere para mayor entendimiento del lector.

(25) También conocido como hipervínculos. Es una estructura de enlace asociativo en Internet, que permite compartir, agregar y enlazar recursos de diferentes fuentes.

(26) Es una red de datos con un alcance geográficamente limitado.



(FTP⁽²⁷⁾, HTTP⁽²⁸⁾, por sus siglas en inglés), se diseñó exclusivamente para proporcionar la transferencia de datos e hipertexto. Por la carencia de seguridad de estos, se desarrolló el FTPS⁽²⁹⁾, SFTP y el HTTPS⁽³⁰⁾ en sus diferentes versiones.

Sub-sección 4.01.1 Clases de Protocolos Seguros de tipo FTP

Para una transferencia de información segura a través de Internet, se utilizan protocolos que encriptados aseguren la confiabilidad e integridad de los datos transmitidos, como son los diferentes tipos de protocolos FTP.

Apartado 4.01.1.1 Protocolo de transferencia de archivos mediante una capa de conexión segura

Esta versión del protocolo FTP, permite transferir los datos de una manera segura, a través de la aplicación de la tecnología de Capa de Conexión Segura⁽³¹⁾ (SSL, por sus siglas en inglés), y los Protocolos de Seguridad de la Capa de Transporte (TLS⁽³²⁾, por sus siglas en inglés). También existen dos variantes de este protocolo que son:

- **FTP de manera implícita:** Para la utilización de esta versión, tanto el servidor como el cliente deben cumplir con una conexión cifrada. En caso de que las mismas no sean seguras o estén cifradas la conexión no sería posible.

(27) Es un protocolo utilizado para la transferencia de archivos.

(28) Es un protocolo utilizado para realizar la transferencia de texto o hipervínculos, a través de la web.

(29) Es un protocolo utilizado para permitir la negociación de mecanismos de seguridad por medio de la autenticación, confidencialidad e integridad de los mensajes.

(30) Es un protocolo utilizado para realizar la transferencia de texto o hipervínculos de manera segura mediante el uso de cifrado, a través de la web.

(31) Es un protocolo de seguridad para conexiones de transmisión de información, el cual emplea autenticación y cifrado de datos.

(32) Se encarga de proporcionar privacidad e integridad de datos entre dos aplicaciones que se comunican a través del internet.



- **FTP de manera explícita:** Para la utilización de esta versión, tanto el servidor como el cliente pueden tener la opción de elegir una conexión cifrada o no.

Existen diferentes tipos de clientes FTP bajo licencia GNU/GPL, de los cuales podemos mencionar los siguientes: FileZilla, FireFTP y gFTP.

Con los mandos AUTH SSL o TLS AUTH, se establece que la encriptación para la conexión es necesaria, y con el comando PROT se puede abrir la encriptación.

Para evitar los fallos de conexión cifrada por la utilización de SSL o TLS y la incompatibilidad con el servidor de seguridad, se pueden utilizar los siguientes puertos:

- **Puerto 989:** para el control de datos⁽³³⁾.
- **Puerto 990:** para el canal de control.

Apartado 4.01.1.2

Protocolo de transferencia de archivos seguro

Este protocolo utilizado con el Intérprete de Órdenes Seguras (SSH⁽³⁴⁾, por sus siglas en inglés), proporciona una forma para acceder y manejar archivos de manera remota utilizando métodos de encriptación. Para poder utilizar el protocolo SFTP, se necesita de clientes especiales.

Los puertos para configurar una conexión SFTP son los siguientes:

(33) Hace referencia a un valor íntegro sobre un elemento determinado, el cual por sí solo carece de importancia y a través del procesamiento adecuado logra convertirse en información útil.

(34) Es un protocolo y aplicación por el cual se accede remotamente a una computadora a través de una red de comunicación.



- Puerto TCP⁽³⁵⁾ 22.
- Puerto UDP⁽³⁶⁾ 22.

Existen clientes SFTP bajo licencia GNU/ GPL, de los cuales se puede mencionar el siguiente: WinSCP

Apartado 4.01.1.3

Comparación entre el SFTP y el protocolo FTPS

Tabla No. 5 Tabla de comparación de protocolos.

	SFTP	FTPS
Encriptación	SSL / TLS	SSH
Aplicación	Implícita / Explícita	Por defecto

Por consiguiente, se puede decir que el protocolo SFTP es mucho más seguro en el momento de la transferencia de archivos, frente a las características que ofrece el FTPS. La recomendación sería utilizar SFTP siempre que sea posible, aunque el uso de FTPS es válido, ya que es un protocolo que cuenta con alto nivel de seguridad.

Sub-sección 4.01.2

Implementación de un servidor SFTP en Linux

La instalación del servidor SFTP en el sistema operativo⁽³⁷⁾ Linux es bastante simple, ya que este lo soporta de forma nativa. Los pasos a seguir son los siguientes:

1. Se necesita instalar el `openssh`⁽³⁸⁾:

(35) Es un protocolo orientado a la conexión de extremo a extremo diseñado para encajar en una jerarquía de capas de protocolos que soportan aplicaciones multi-red.

(36) Es un protocolo de transporte de datos, específicamente datagramas, el cual opera en la capa de transporte del modelo OSI.

(37) Es un software utilizado en los sistemas de información para gestionar y administrar los recursos de los dispositivos o computadores.

(38) Es una herramienta utilizada para conectividad remota a través de Internet.





```
root@optic: ~ # apt-get install openssh-s
```

2. Se deben crear los directorios y asignar los permisos donde van a acceder los usuarios, los cuales deben tener los permisos de lugar:

```
root@optic: ~ # mkdir /home/sftp
root@optic: ~ # mkdir /home/sftp/optic
root@optic: ~ # chmod 755 /home/sftp
root@optic: ~ # chmod 755 /home/sftp/optic
```

3. Crear el grupo (sftp) y usuario (opticusr):

```
root@optic: ~ # groupadd sftp
root@optic: ~ # useradd -g sftp -s /bin/false -d /home/sftp/optic
opticusr
root@optic: ~ # passwd redorbita
```

- a. Introduzca la nueva contraseña de UNIX⁽³⁹⁾.
 - b. Vuelva a escribir la nueva contraseña de UNIX.
 - c. passwd: contraseña actualizada correctamente.
4. Configurar Enjaulado SFTP:

- a. Acceder al archivo de configuración:

```
root@optic: ~ # vi /etc/ssh/sshd_config
```

- b. Buscar la siguiente línea y comentarla (Se pone delante #), de la siguiente forma:

(39) Es un sistema operativo multitarea desarrollado por Laboratorios Bell.



```
#Subsystem sftp /usr/lib/openssh/sftp-server
```

- c. Ir al final del documento y añadir lo siguiente:

```
Subsystem sftp internal-sftp
```

Si se desea proporcionar permisos al grupo en el cual pertenecen los usuarios, se asigna el directorio raíz que se creó anteriormente.

De esta forma se obtendrá lo siguiente:

```
Match group sftp
ChrootDirectory /home/sftp
ForceCommand internal-sftp
```

En cambio, si se desea proporcionar permisos a un usuario en concreto, se añade lo siguiente:

```
UsePAM yes
Match user opticusr
ChrootDirectory /home/sftp/optic
ForceCommand internal-sftp
```

5. Reiniciar ssh:

```
root@optic: ~ # /etc/init.d/ssh start
Starting OpenBSD Secure Shell server: sshd.
```

6. Finalmente, se accede a los archivos del servidor desde cualquier sistema operativo con un cliente que soporte SFTP, tal como FileZilla, WinSCP, entre otros, los cuales son gratuitos; o utilizar cualquier otro de su preferencia.

Sub-sección 4.01.3

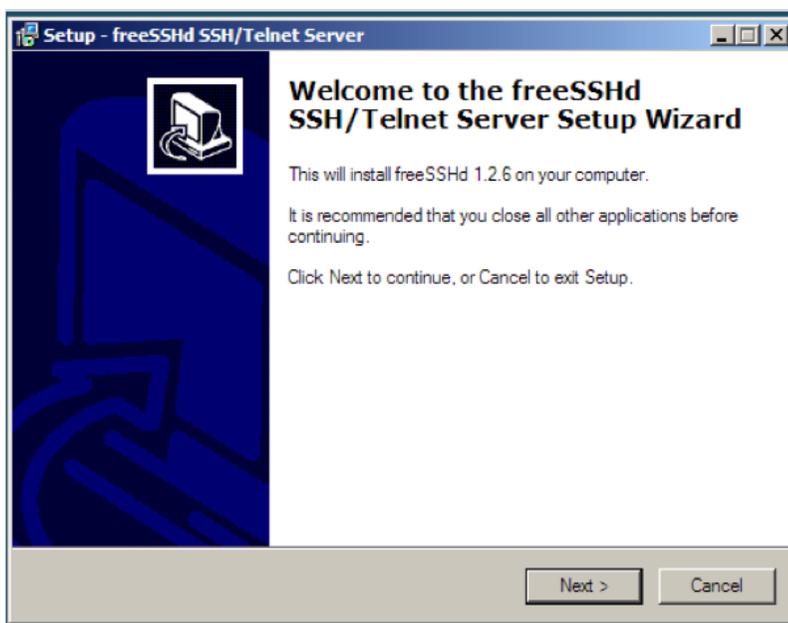
Implementación de
un servidor SFTP en Windows

Para la instalación de un servidor de SSH en el sistema operativo Windows, se necesitará una aplicación adicional para configurar SSH, una de las más utilizadas es la llamada freesshd.

A continuación se presentan los pasos para la instalación de una de las herramientas gratuitas y de código⁽⁴⁰⁾ abierto más utilizados:

1. Una vez descargado freesshd se debe ejecutar el instalador, el cual abrirá un asistente de instalación. Luego presionar clic en “Next”, como se indica en la **Figura No. 13 Asistente de Instalación**.

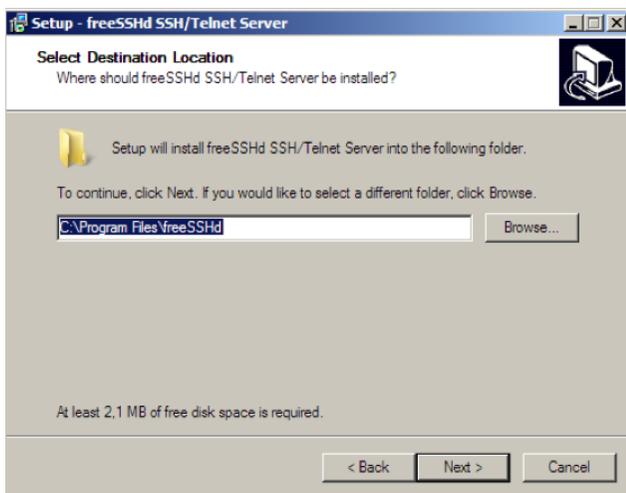
Figura No. 13 Asistente de Instalación.



(40) Es un conjunto de instrucciones redactas en base a las reglas sintácticas de un lenguaje de programación para desarrollar un software determinado.

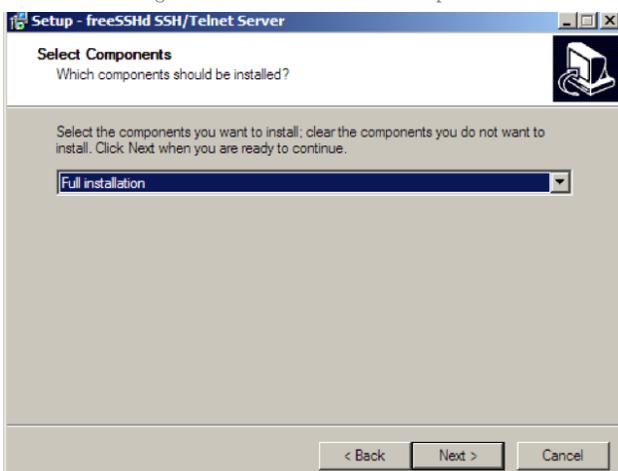
2. Indicar la ruta donde desea instalarlo. Luego presionar clic en “Next”, como se indica en la **Figura No. 14 Selección de la ruta de instalación.**

Figura No. 14 Selección de la ruta de instalación.



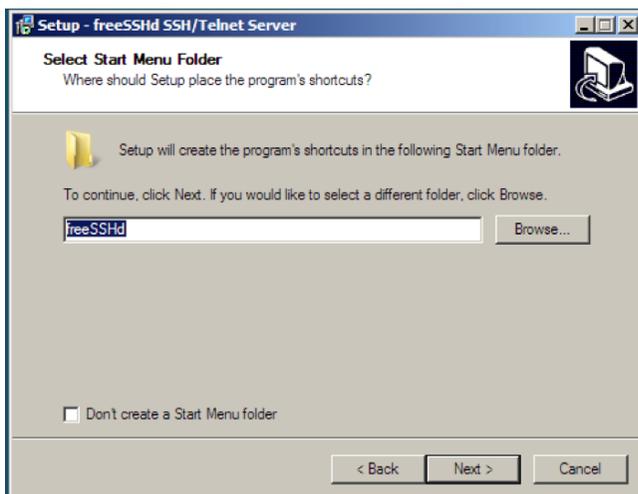
3. Seleccione “Full installation”. Luego presione clic en “Next”, como se indica en la **Figura No. 15 Selección de componentes.**

Figura No. 15 Selección de componentes.



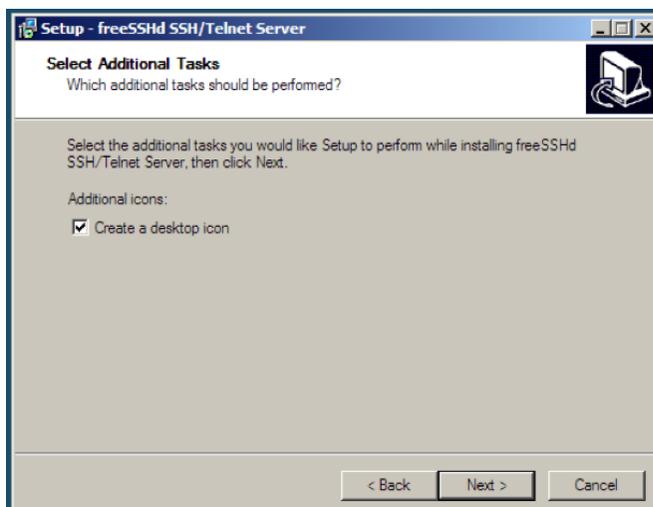
4. Asignar un nombre a la carpeta. Luego presione clic en “Next”, como se indica en la **Figura No. 16 Selección de la carpeta.**

Figura No. 16 Selección de la carpeta.



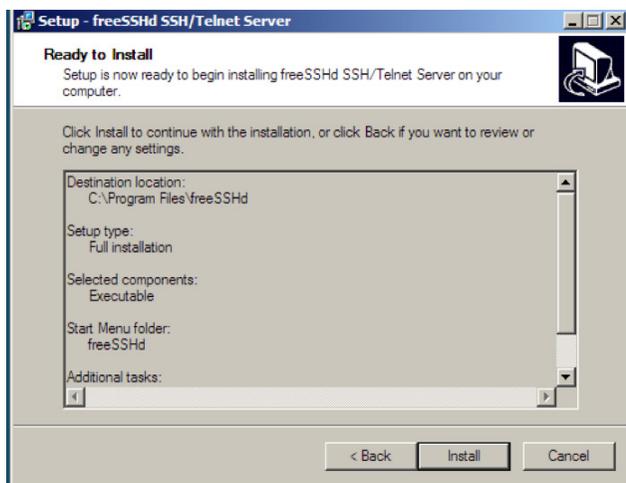
5. Puede seleccionar la opción para crear un icono en el escritorio. Luego presione clic en “Next”, como se indica en la **Figura No. 17 Creación de ícono.**

Figura No. 17 Creación de ícono.



6. Se muestra un resumen antes de la instalación. Luego presionar “Install”, como se indica en la **Figura No. 18 Resumen de la instalación.**

Figura No. 18 Resumen de la instalación.



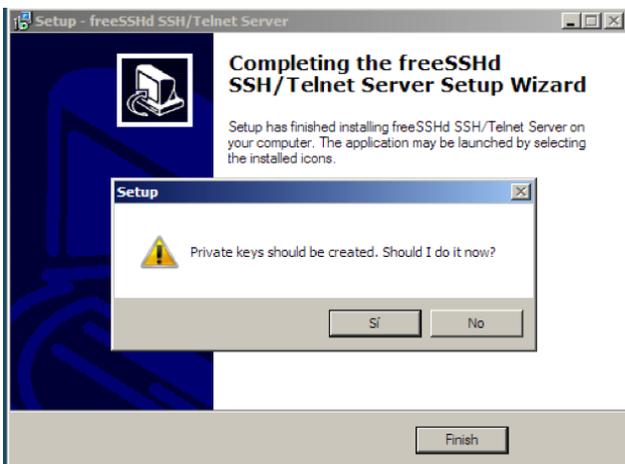
7. Presionar clic en “Close”, como se indica en la **Figura No. 19 Productos adicionales.**

Figura No. 19 Productos adicionales.



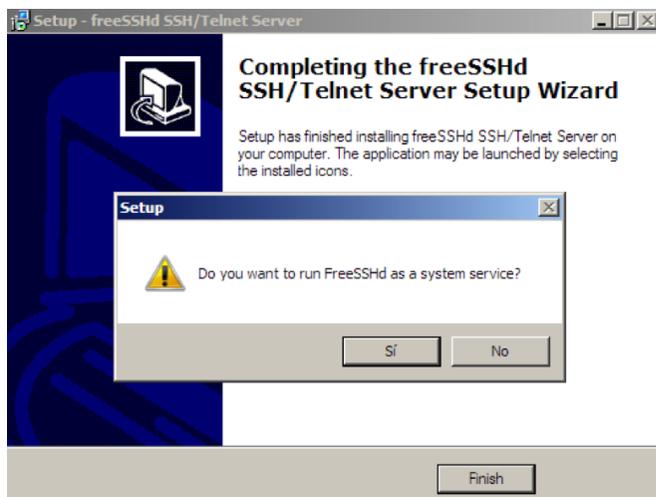
- Se presentará un mensaje el cual indicará, si se desea crear la clave privada. Presione clic en **Sí**, como se indica en la **Figura No. 20 Creación de clave privada.**

Figura No. 20 Creación de clave privada.



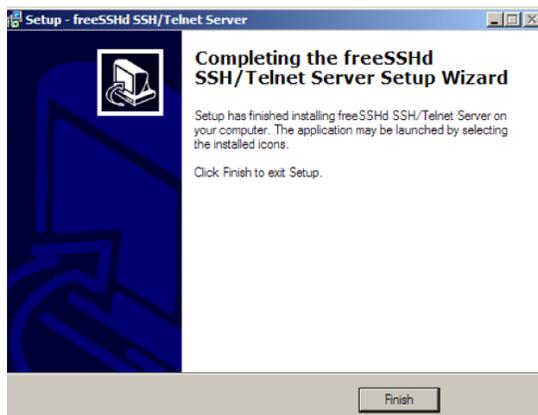
- Se mostrará un mensaje indicando si desea ejecutar FreeSSHd como un servicio del sistema. Luego presionar clic en **Sí**, como se indica en la **Figura No. 21 Servicio del sistema.**

Figura No. 21 Servicio del sistema.



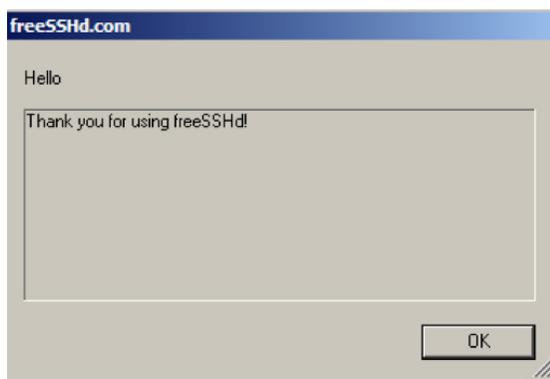
10. Para finalizar la instalación, presionar clic en **“Finish”**, como se indica en la **Figura No. 22 Finalizando la instalación.**

Figura No. 22 Finalizando la instalación.



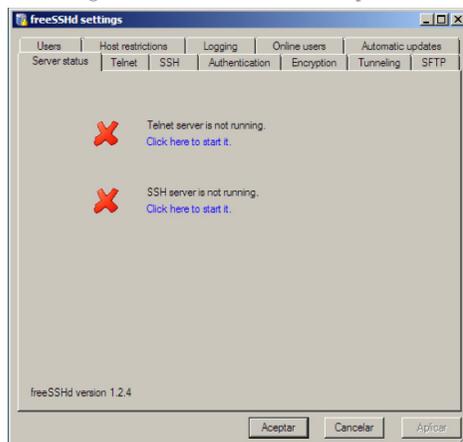
11. Ejecutar el programa FreeSSHd con permisos de **Administrador** (presionar clic con el botón derecho del ratón en el icono de la aplicación y luego presionar clic en **Ejecutar como Administrador** para configurarlo adecuadamente).
12. Luego aparecerá esta pantalla, la cual debe **Aceptar**, como se indica en la **Figura No. 23 Pantalla de uso.**

Figura No. 23 Pantalla de uso.



13. Se mostrará la pantalla principal. En la cual iniciará el servicio una vez configurado, como se indica en la **Figura No. 24 Pantalla Principal.**

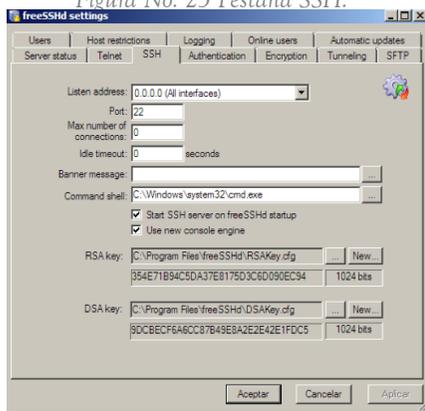
Figura No. 24 Pantalla Principal.



14. Ir a la pestaña SSH. En ella se podrá configurar la dirección IP por la cual escuchará las llamadas del servicio, el puerto, máximo de usuarios conectados y el shell que se utilizará.

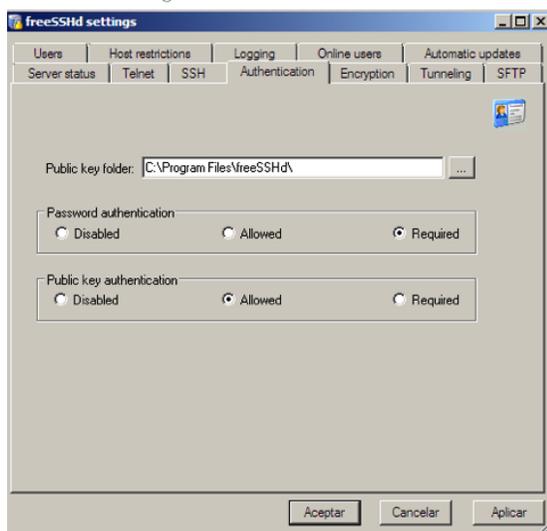
Tanto para una instalación simple como para un ambiente de desarrollo o prueba, se puede dejar todo por defecto, aunque para un ambiente de producción se debe especificar direcciones IP y puertos reservados.

Figura No. 25 Pestaña SSH.



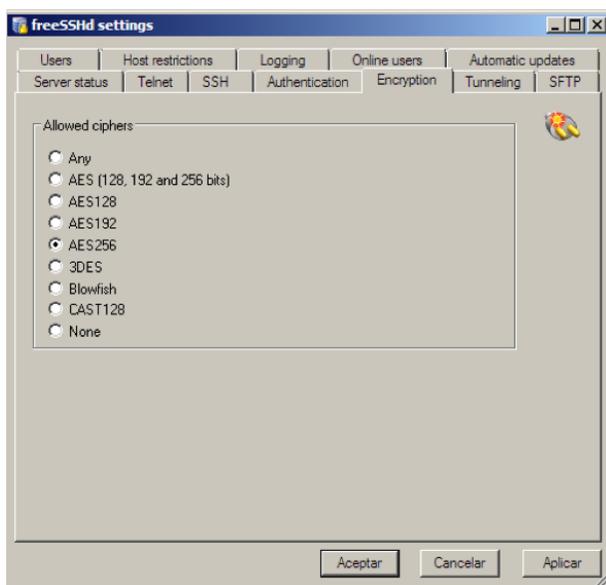
15. Presionar clic en la pestaña “**Authentication**”, para configurarlo que pida contraseña.

Figura No. 26 Pantalla de uso.



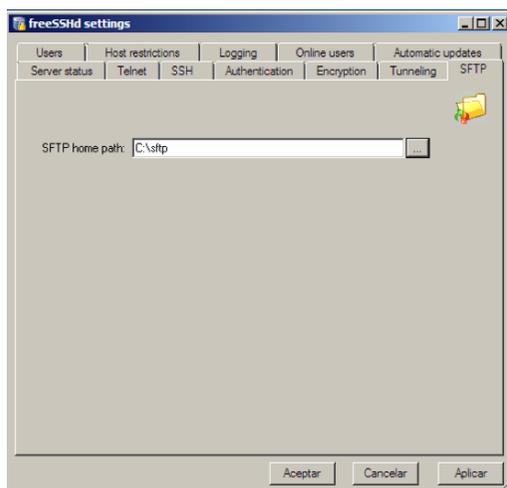
16. Configurar el tipo de encriptación.

Figura No. 27 Pestaña de encriptación.



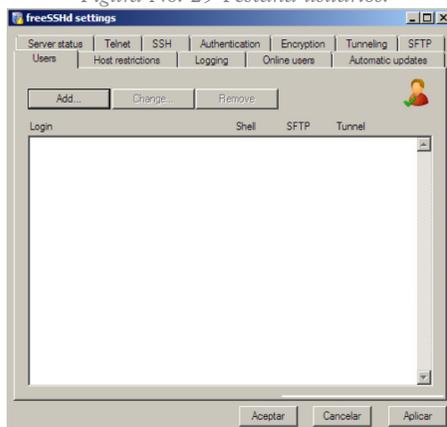
17. Una vez configurado el servidor SSH, se configurará el servidor SFTP. Para ello debe dirigirse a la pestaña SFTP, como se indica en la **Figura No. 28 Selección de carpeta destino.**
18. Indicar la carpeta donde se conectarán los usuarios que tengan acceso al servidor SFTP, como se indica en la **Figura No. 28 Selección de carpeta destino.**

Figura No. 28 Selección de carpeta destino.



19. Dirigirse a la pestaña de “Users”, para configurar el usuario de acceso, como se indica en la **Figura No. 29 Pestaña usuarios.**

Figura No. 29 Pestaña usuarios.



20. Se abrirá un asistente donde deberá añadir el nombre de usuario, la forma de autenticación y los accesos a dicho usuario, en este caso se le han asignado todos los permisos, así que de desearse que solo sea para el uso vía SFTP solo se debe marcar SFTP, como se indica en la **Figura No. 30 Propiedades de usuarios.**

Figura No. 30 Propiedades de usuarios.

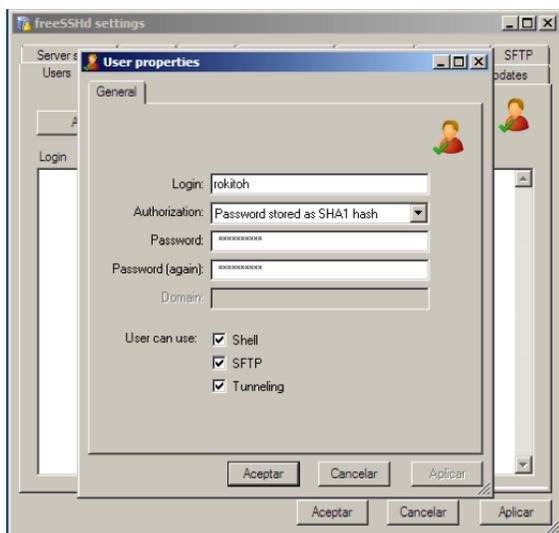
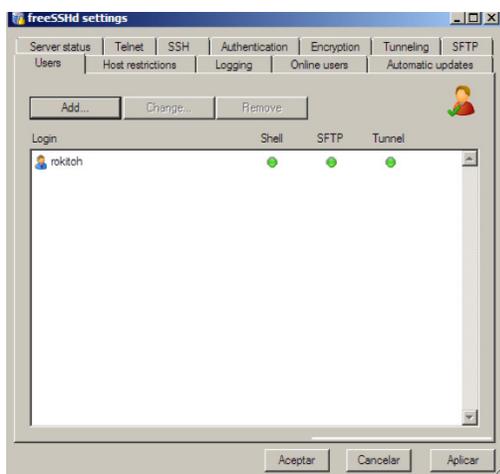
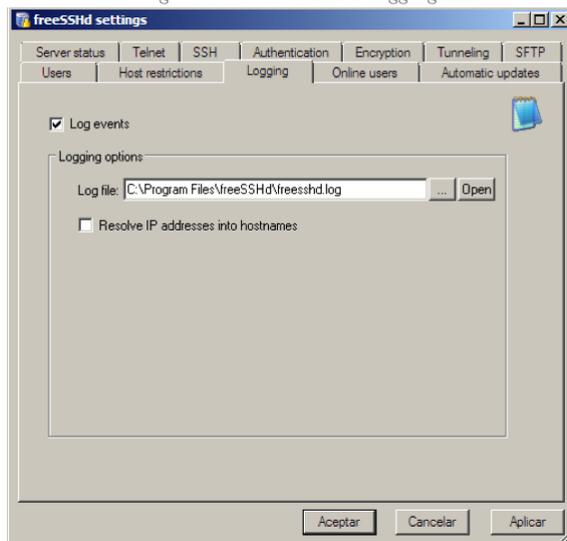


Figura No. 31 Propiedades de usuarios (2).



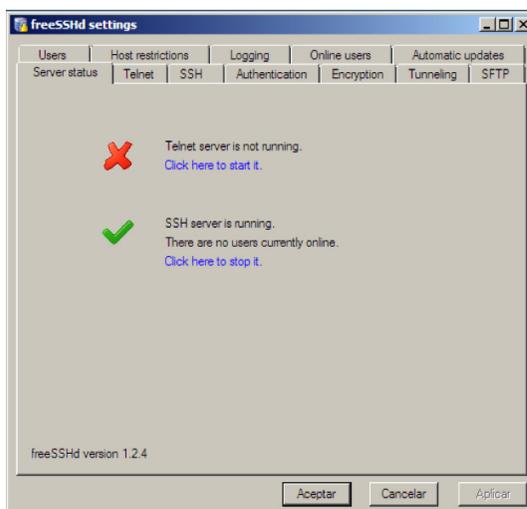
21. Seleccionar la pestaña “**Logging**” y presione clic en “**Log events**”, como se indica en la **Figura No. 32 Pestaña Logging**.

Figura No. 32 Pestaña Logging.



22. Nuevamente dirigirse a la pestaña principal e inicie el servidor “**Server status**”, como se indica en la **Figura No. 33 Configuraciones de freeSSFd**.

Figura No. 33 Configuraciones de freeSSFd.



23. Acceder mediante SSH para comprobar que la configuración es funcional, como se indica en la **Figura No. 34 Comprobando la configuración** y **Figura No. 35 Comprobando la configuración (2)**.

Figura No. 34 Comprobando la configuración.

```
rokito@lusu:~$ ssh rokito@192.168.1.188
rokito@192.168.1.188's password: █
```

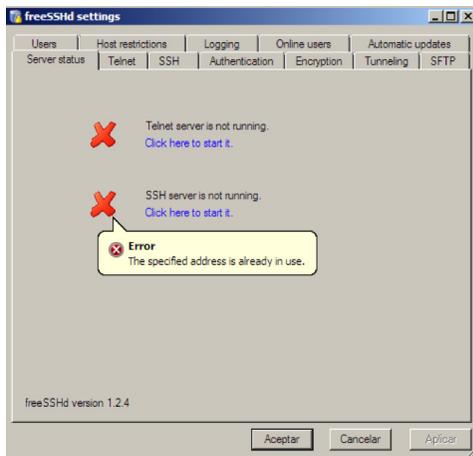
Figura No. 35 Comprobando la configuración (2).

```
Microsoft Windows [Version 6.0.6001] 1.188
Copyright (c) 2006 Microsoft Corporation. Reservados todos los derechos.
C:\Users\Administrador\Desktop> █
```

Apartado 4.01.3.1 Posibles problemas durante la instalación

1. Si al intentar ejecutarlo nos indica que el puerto ya se está usando, probablemente se tienen abiertas 2 instancia del servicio de freesshd o está corriendo otro proceso por el mismo puerto, como se indica en la **Figura No. 36 Error de puerto**.

Figura No. 36 Error de puerto.

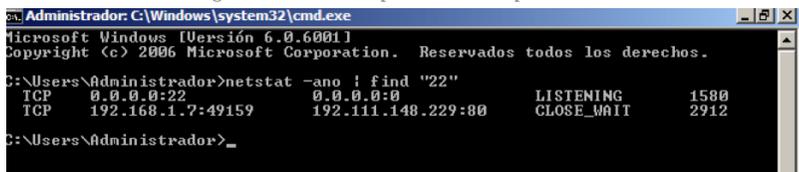




2. Debe acceder a la consola⁽⁴¹⁾ de MS-DOS y compruebe que está corriendo por el puerto 22, como se indica en la **Figura No. 37 Comprobación del puerto 22.**

```
>/> netstat -ano | FIND "22"
```

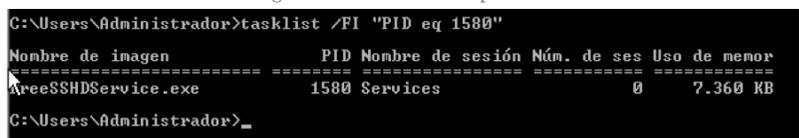
Figura No. 37 Comprobación del puerto 22.



3. Listar el proceso.

```
>/> tasklist /FI "PID eq 1580"
```

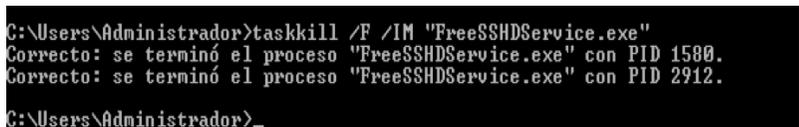
Figura No. 38 Listar el proceso.



4. Terminar el proceso.

```
taskkill /F /IM "FreeSSHDSvc.exe"
```

Figura No. 39 Terminar el proceso.



(41) Es el medio por el cual el usuario interactúa con el sistema mediante la ejecución de comandos.

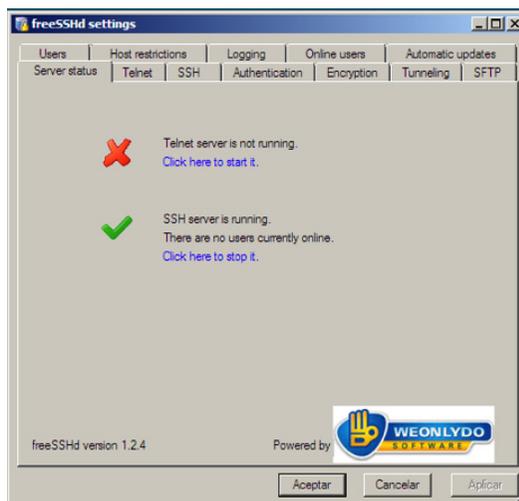
5. Comprobar nuevamente.

Figura No. 40 Comprobar el proceso.

```
C:\Users\Administrador>netstat -ano | find "22"  
C:\Users\Administrador>
```

6. Inicie nuevamente el programa y tendría que aparecerle en verde.

Figura No. 41 Revisión del servicio.



Sub-sección 4.01.4

Clases de protocolos de transferencia de hipertexto

Para una transferencia de información segura a través del Internet utilizando hipertexto, se implementan protocolos encriptados que aseguren la confiabilidad e integridad de los hiperenlaces, como son los diferentes tipos de protocolos HTTP.

Apartado 4.01.4.1

Protocolo de transferencia de hipertexto

Es un protocolo utilizado para realizar la transferencia de texto o hipervínculos, a través de la web y pertenece al nivel de aplicación.



La primera versión de HTTP implementada fue la HTTP / 0.9, era un protocolo sencillo para la transferencia de datos en bruto a través de Internet. Luego se estableció la versión 1.0 y la 1.1, la cual permite las jerarquía de proxies⁽⁴²⁾, almacenamiento en caché⁽⁴³⁾ y otras particularidades.

No se requiere de ningún tipo de implementación especial para utilizar el protocolo, debido a que HTTP es un protocolo estándar para la transferencia de hipertexto y todos los servidores web deben ser compatible con el estándar.

Apartado 4.01.4.2

Protocolo de transferencia de hipertexto seguro

Es un protocolo utilizado para realizar la transferencia de texto o hipervínculos de manera segura mediante el uso de cifrado, a través de la web.

Este protocolo es el más utilizado y seguro para acceder a los contenidos que nos ofrece la web, debido a que toda la información que se introduzca será cifrada, permitiendo que el dato o la información no podrá ser vista por nadie más que el cliente y el servidor, invalidando de esta forma la posibilidad de que la información pueda ser utilizada.

Apartado 4.01.4.3

Recomendación de uso del protocolo HTTP

Cuando se utiliza HTTP/SSL se tiene más privacidad y seguridad que una conexión web que no contenga encriptación como lo es HTTP. Usando HTTPS se disminuye el riesgo de que terceras personas intercepten y usen indebidamente la información que se está

(42) Es un servidor o aplicación utilizado como intermediario para el envío de información en una red de datos.

(43) Es un tipo de memoria de acceso rápido, la cual almacena la última información de manera temporal.



utilizando en la web. Este protocolo es más utilizado cuando se realizan pagos y compartir información personal.

Para verificar el HTTPS en páginas web, la mayoría de los navegadores web exponen un ícono en forma de candado cuando se ha establecido una conexión con SSL.

La forma más rápida de verificar la tecnología HTTPS es introducir la dirección web en el navegador con <https://> al comienzo, por ejemplo, <https://URL-de-la-página.gob.do>. Si se muestra un ícono en forma de candado en el navegador, se puede hacer clic en él para obtener más información que confirme la seguridad de la conexión. Si no se muestra el ícono de candado, significa que la página no está protegida con SSL.

Apartado 4.01.4.4 Implementación de protocolo HTTPS

Para que este protocolo pueda funcionar, se implementará un certificado digital⁽⁴⁴⁾ en el servidor web donde este vaya a funcionar:

1. Generar primero una Solicitud de Certificación (CSR⁽⁴⁵⁾, por sus siglas en inglés) para el servidor:
 - La CSR es un cuerpo de texto cifrado. Su CSR contendrá información codificada y específica para su compañía y el nombre de dominio; información que se conoce como un nombre completo o DN.
 - El DN, para la mayoría de servidores, contiene los siguientes campos: País, Estado (o provincia), Localidad (o ciudad),

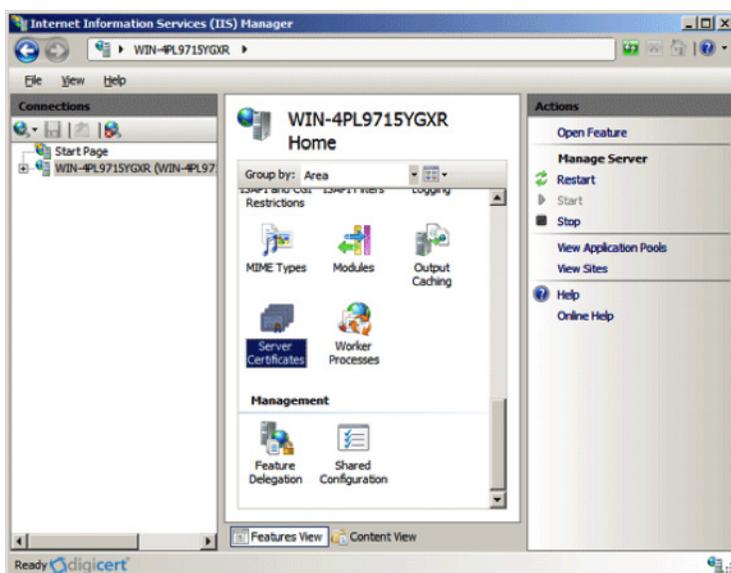
(44) Es un documento digital que permite garantizar la identidad de una persona en la red, a través de una firma electrónica. Se utiliza como una forma segura de garantizar la autenticación, integridad y confidencialidad de la información.

(45) Es la forma los servicios y sitios web en el Internet obtienen una validación para el cifrado de los datos que se envían por la red.

la Organización de la Unidad Organizacional y nombre común.

2. Para generar un Código CSR en los Servicio de Información de Internet⁽⁴⁶⁾ de Microsoft (IIS, por sus siglas en inglés) 7 o superior:
 - a) Ir a Inicio, hacer clic en **Herramientas Administrativas** y seleccionar “**Internet Information Services (IIS)**”.
 - b) Hacer clic en el **Nombre de servidor**.
 - c) En el menú central, hacer doble clic en el botón **Certificados del Servidor** en la sección Seguridad, que está cerca del botón del menú, como se muestra en la gráfica siguiente:

Figura No. 42 Servidor de certificados.



(46) Es un servidor web que permite compartir información con usuarios en Internet, en una intranet o en una extranet.

- d) Después, ir al menú de “**Actions**” que se encuentra a la derecha, hacer clic en “**Create Certificate Request**”. Esta acción abrirá el Asistente de Solicitud de Certificado.

Figura No. 43 Solicitud de certificado.



- e) En la ventana “**Distinguished Name Properties**”, introducir la información a continuación, como se muestra en la **Figura No. 37 Propiedades de Nombre Distintivo**:

- **Nombre común:** El nombre mediante el cual se accederá al certificado (usualmente es el nombre de dominio completo, p. ej. www.dominio.com o mail.dominio.com).
- **Organización:** La razón social registrada de su organización o empresa.
- **Unidad organizativa:** El nombre de su departamento dentro de la organización (frecuentemente esta entrada se muestra como “Informática”, “Seguridad web” o simplemente se deja en blanco).
- **Ciudad/localidad:** la ciudad en donde está ubicada su organización.

- **Estado/provincia:** el estado en donde está ubicada su organización.
- **País/región:** en caso de ser necesario.

Figura No. 44 Propiedades de Nombre Distintivo.

Request Certificate

Distinguished Name Properties

Specify the required information for the certificate. State/province and City/locality must be specified as official names and they cannot contain abbreviations.

Common name:

Organization:

Organizational unit:

City/locality:

State/province:

Country/region:

Previous Next Finish Cancel

- f) Hacer clic en “Next”.
- g) En la ventana “Cryptographic Service Provider Properties”, dejar ambos parámetros con sus valores predeterminados y, después, hacer clic en “Next”.

Figura No. 45 Propiedades del proveedor de servicios criptográficos.

Request Certificate

Cryptographic Service Provider Properties

Select a cryptographic service provider and a bit length. The bit length of the encryption key determines the certificate's encryption strength. The greater the bit length, the stronger the security. However, a greater bit length may decrease performance.

Cryptographic service provider:

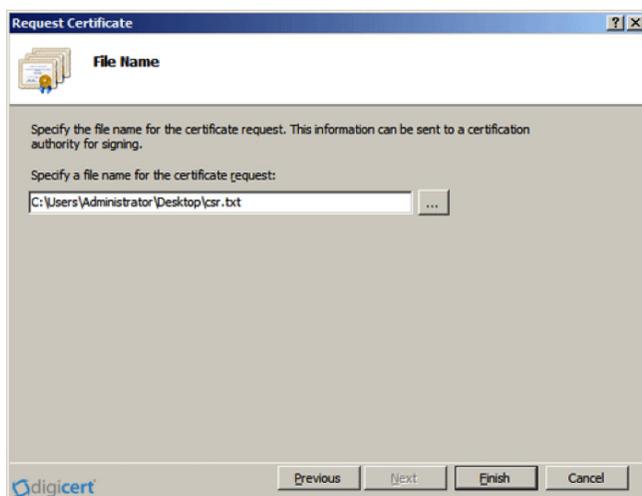
Bit length:

Previous Next Finish Cancel

- h) Se debe introducir un nombre al archivo de **Solicitud de Firma de Certificado**.

Especificar el nombre de archivo y la ubicación en donde guardará. Será necesario abrir el archivo como un documento de texto y copiar todo el contenido (incluidas las etiquetas *BEGIN CERTIFICATE REQUEST* y *END CERTIFICATE REQUEST*) en el proceso del pedido en línea cuando el sistema se lo solicite.

Figura No. 46 Solicitud de Firma de Certificado.



3. Instalar el certificado digital en el IIS7 o superior:
- a) Abrir el archivo ZIP⁽⁴⁷⁾ que contiene el certificado. Guardar el archivo con el nombre `su_dominio_com.cer` en el escritorio del servidor web que está protegiendo.
 - b) Hacer clic en Inicio, y después, en Herramientas administrativas e Administrador de servicios de información de Internet (IISM⁽⁴⁸⁾, por sus siglas en inglés).

(47) Es un formato de compresión de archivos sin pérdida que comprime cada uno de los archivos de forma separada

(48) Es una interfaz gráfica para configurar los grupos de aplicaciones o los sitios Web, FTP, SMTP o NNTP.

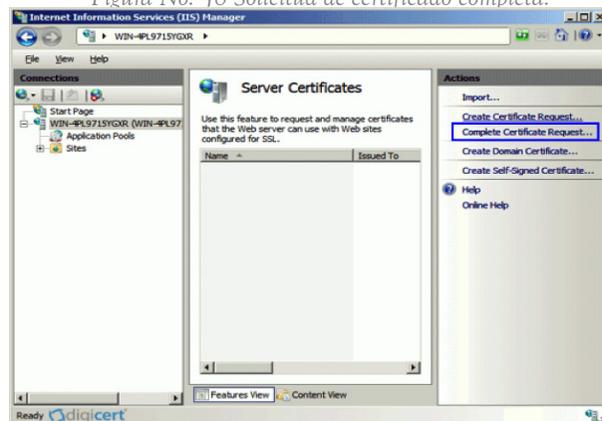
- c) Hacer clic en el Nombre del servidor.
- d) En el menú central, hacer doble clic en el botón “**Server Certificates**” en la sección Seguridad (cerca de la parte inferior del menú), como se muestra en la **Figura No. 47** **Certificados del servidor**.

Figura No. 47 *Certificados del servidor.*



- e) En el menú “**Actions**”, ubicado a la derecha, hacer clic en “**Complete Certificate Request**”. Esta etapa abrirá el asistente de **Complete la Solicitud de Certificado**, como se muestra en la **Figura No. 48** **Solicitud de certificado completa**.

Figura No. 48 *Solicitud de certificado completa.*



- f) Ir hasta el archivo `su_dominio_com.cer` que le proporcionó DigiCert. Después, se requerirá introducir un nombre simple. El nombre simple no forma parte del certificado propiamente dicho, pero el administrador del servidor lo usará para distinguir fácilmente el certificado.

Figura No. 49 Especificación del certificado.

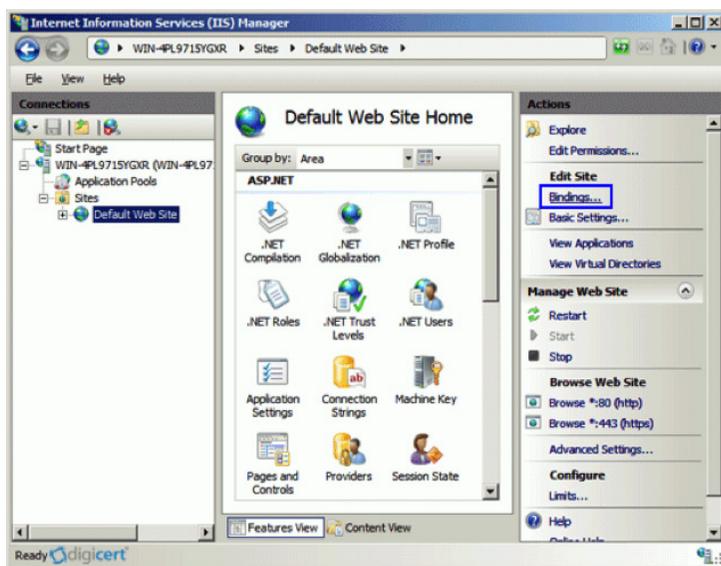


- g) Hacer clic en **Aceptar**, y se instalará el certificado en el servidor.
- h) Una vez que el certificado SSL se haya instalado correctamente en el servidor, se asignará ese certificado al sitio web correspondiente mediante IIS.

Existe un problema conocido en IIS7 que causa el siguiente error: "No se puede encontrar la solicitud de certificado asociada a este archivo del certificado. Debe completarse una solicitud de certificado en el equipo donde se creó". Además, es posible que reciba un mensaje que indica "Se encontró un valor de etiqueta ASN1 incorrecto". Si este es el mismo servidor en donde generó la CSR, en la mayoría de los casos el certificado está realmente instalado. Simplemente cancele el cuadro de diálogo y presione "F5" para actualizar la lista de certificados del servidor. Si el nuevo certificado está ahora en la lista, puede continuar con la etapa siguiente. Si no está en la lista, deberá volver a emitir su certificado mediante una nueva CSR. Después de crear una nueva CSR, inicie sesión en su cuenta de DigiCert y hacer clic en el botón para volver a emitir su certificado.

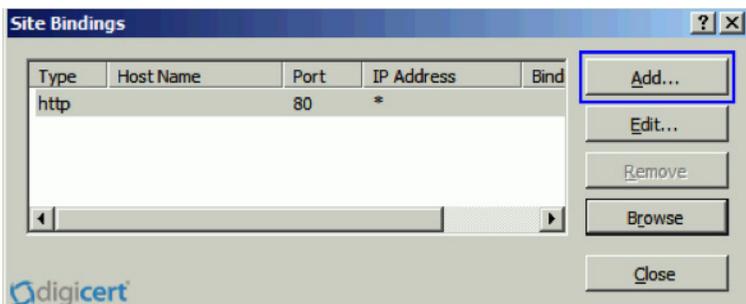
- i) En el menú “**Connections**” en la ventana principal del **Administrador de servicios de información de Internet (IIS)**, seleccionará el nombre del servidor en donde se instaló el certificado.
- j) En “**Sites**”, seleccionará el sitio que se protegerá con SSL.
- k) En el menú “**Actions**” ubicado a la derecha, hacer clic en **Enlaces**. Esta acción abrirá los **Enlaces del sitio**, como se muestra en la **Figura No. 50 Administrador de servicios de información de Internet**.

Figura No. 50 Administrador de servicios de información de Internet.



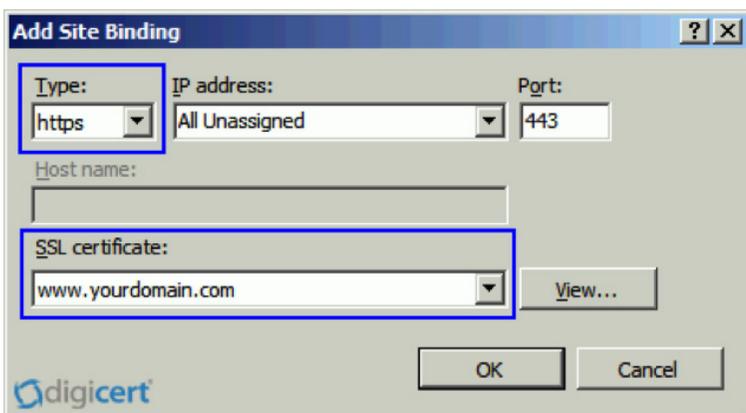
- l) En la ventana Enlaces del sitio, hacer clic en “**Add**”. Esto abrirá la ventana **Agregar enlace del sitio**, como se muestra en la **Figura No. 51 Enlace del sitio**.

Figura No. 51 Enlace del sitio.



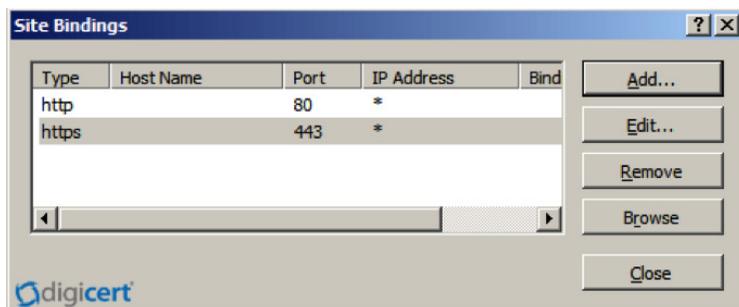
- m) En “Type” elegir **https**. La dirección de IP debe ser la dirección de IP del sitio o **enlace del sitio**, y el puerto en donde el tráfico estará protegido por SSL es generalmente **443**. El campo Certificado SSL debe especificar que el certificado se instaló en la etapa 7. Como se muestra en la **Figura No. 52 Agregar enlace del sitio**.

Figura No. 52 Agregar enlace del sitio.



- n) Hacer clic en **Aceptar**, como se muestra en la **Figura No. 53 Enlaces del sitio (2)**.

Figura No. 53 Enlaces del sitio (2).



- o) El certificado SSL está ahora instalado y el sitio web configurado para aceptar conexiones seguras.

Sección 4.02

Implementación de bases de datos

Esta sección, se enfoca en todo lo referente al uso de las bases de datos, estará explicando los conceptos básicos de la implementación, diseño y tipos de bases de datos para lograr una correcta implementación de la misma en los organismos gubernamentales.

Una base de datos es un conjunto de datos almacenados de una forma lógica y generalmente basados en el mismo contexto de la información. Al día de hoy, las bases de datos han evolucionado de forma significativa año tras año, debido, principalmente, a las necesidades de empresas e instituciones que son las que adicionan nuevos retos a la tecnología.

Sub-sección 4.02.1

Diseño de bases de datos

Para la implementación de una base de datos, es necesario definir correctamente la estructura de datos que esta debe tener, lo cual se logra por medio del diseño de la misma basándose en tres (3) diferentes tipos, estos son:



- **Diseño conceptual:** Para este diseño se procede a hacer una descripción detallada y de alto nivel de la estructura de la base de datos, independientemente del **Sistema Gestor de Bases de Datos (SGBD)**, por sus siglas en inglés) que se utilizará para gestionarla. El objetivo de este diseño es describir el contenido de información de la base de datos y no las estructuras de almacenamiento que se necesitarán para manejar dicha información.
- **Diseño lógico:** Este diseño parte del resultado del diseño conceptual antes mencionado, que resulta en una descripción de la estructura de la base de datos para determinar las estructuras que puede procesar un tipo de SGBD. El diseño lógico depende del tipo de SGBD que se vaya a utilizar; este se puede adaptar a la tecnología que se debe implementar, pero no depende del producto final. En el caso de bases de datos convencionales o relacionales, basadas en **Lenguaje Estructurado de Consultas (SQL⁽⁴⁹⁾)**, por sus siglas en inglés) por ejemplo, el diseño lógico consiste en definir las tablas que existirán, las relaciones entre ellas y normalizarlas.
- **Diseño físico:** Este diseño tiene como base el diseño lógico antes mencionado y da como resultado una descripción de la implementación de una base de datos en memoria secundaria, de igual manera las estructuras adecuadas de almacenamiento y los métodos necesarios para tener un acceso eficiente a los datos. El diseño físico tiene como objetivo conseguir una mayor eficiencia en la gestión de la base de datos.

(49) También conocido como lenguaje de consulta estructurado, es un lenguaje para acceder a bases de datos relacionales que permite la especificación de acciones y operaciones.



Sub-sección 4.02.2

Tipos de bases de datos

A continuación se desglosan los tipos de bases de datos, dando aspectos generales de cada uno y explicando brevemente cómo lograr exitosamente la implementación de cada uno de los tipos de bases de datos mencionadas.

Apartado 4.02.2.1

Bases de datos relacionales

El modelo de base de datos relacionales utiliza las dos capas de diseño conceptual y lógico. Por lo general se implementan mediante diagramas de Entidad/Relación, lo que muestra un esquema conceptual de su estructura de tablas y relaciones entre sí, siendo esta la parte lógica. Este es el modelo utilizado por los sistemas gestores de datos más habituales como SQL Server⁽⁵⁰⁾, Oracle⁽⁵¹⁾, MySQL⁽⁵²⁾, entre otros.

El lenguaje ideal para la elaboración de consultas a bases de datos es SQL, este es un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales. En el modelado de bases de datos relacionales se deben tomar en consideración las conocidas “formas normales⁽⁵³⁾”, las cuales se utilizan para la nacionalización de bases de datos y se detallan a continuación:

(50) SQL Server es un sistema creado por Microsoft creado para la administración y gestión de bases de datos relacionales.

(51) Es una base de datos para la gestión de base de datos objeto-relacional. Es multiplataforma y es conocida por su gran estabilidad y desempeño.

(52) Es una base de datos para la gestión de base de datos relacional, multihilo y multiusuario.

(53) Las formas normales son teorías que proporcionan criterios para determinar qué tan vulnerable puede ser una tabla de una base de datos a anomalías e inconsistencias.



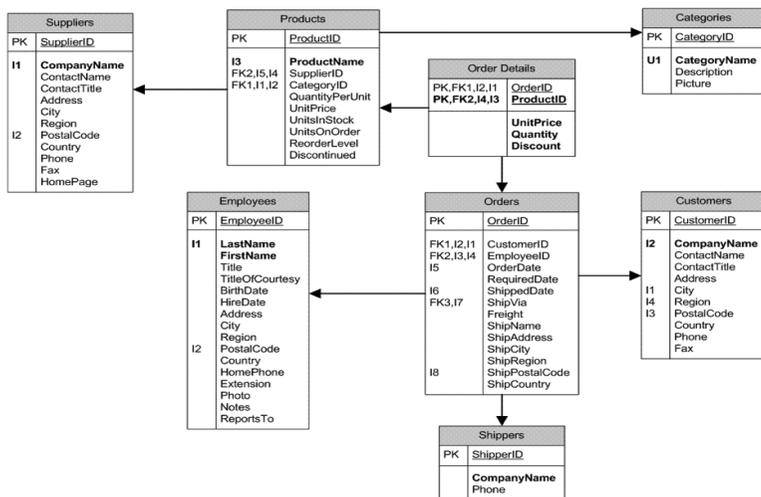
- **Primera forma normal (1FN)** tiene las siguientes características:
 - Contiene atributos indivisibles, es decir atómicos.
 - Tabla con contenido primario único.
 - Claves primarias sin atributos nulos.
 - Tabla sin múltiples valores por columna.
- **Segunda forma normal (2FN)** tiene las siguientes características:
 - Permite crear tablas separadas de datos.
 - Relación de tablas por claves externas.
 - Dependencia funcional.
- **Tercera forma normal (3FN)** tiene las siguientes características:
 - Elimina aquellos campos que no dependen de la clave.
 - La tabla está en la segunda forma normal (2FN).
 - Contiene atributos no primarios que dependen de claves primarias.
- **Cuarta forma normal (4FN)** tiene las siguientes características:



- o Dependencias multivaluadas eficientes en la base de datos.
- o No posee dependencias multivaluadas no triviales.
- o Dependencia de dos o más relaciones independientes.
- **Quinta forma normal (5FN)** tiene las siguientes características:
 - o Busca reducir redundancia en bases de datos.
 - o Contiene dependencias no triviales que no siguen los criterios de las claves.
 - o La tabla en la cuarta forma normal (4FN) está en la quinta forma normal (5FN) si existe relación de dependencia por claves.

Un diseño conceptual para gestionar múltiples informaciones puede estructurarse como se muestra la **Figura No. 54 Estructura de base de datos relacionales**, esto se denomina diagrama Entidad/Relación o diagrama E-R.

Figura No. 54 Estructura de base de datos relacionales.



Como se puede ver en la figura anterior, esta base de datos contiene tablas para representar cada entidad como son proveedores, productos, categorías, entre otros; estas a su vez están relacionadas unas con otras donde un producto, por ejemplo, pertenece a una categoría, a la vez que el producto tiene un **Identificador Único (ID)**, por sus siglas en inglés); lo mismo sucede con las demás tablas de la base de datos.

Los campos identificados como **Claves Primarias (PK)**, por sus siglas en inglés) son aquellos marcados como únicos para cada entidad, por ejemplo, el **ID de producto**, que está marcada como una clave primaria, será única lo que incrementa un dígito entero por cada producto nuevo que se inserte en la base de datos. En cambio los campos marcados como **Claves Foráneas (FK)**, por sus siglas en inglés) son aquellos que almacenan las claves primarias de otras tablas de una forma relacional. Los campos marcados como **Índices (I)**, por su sigla en inglés), son campos que contienen información adicional para una rápida localización de registros en la base de datos. Por último, los campos marcados como **Únicos (U)**, por su sigla en inglés) hacen referencia a campos que deben ser únicos y no deben repetirse.



¿Qué es SQL Server?

SQL Server es un gestor de base de datos relacional, este utiliza los lenguajes de consulta T-SQL y ANSI SQL; a parte de esta solución de base de datos relacional, también existen otras opciones muy potentes como Oracle, PostgreSQL o MySQL.

Las características de SQL Server son:

- Soporte de transacciones.
- Soporta procedimientos almacenados.
- Incluye también un entorno gráfico para la administración de las bases de datos.
- Permite el uso de comandos de **Lenguaje de Definición de Datos (DDL**, por sus siglas en inglés) y **Lenguaje de Manipulación de Datos (DML**, por sus siglas en inglés) gráficamente.
- Permite trabajar en modo cliente-servidor.
- Permite administrar información de otros servidores de datos.

Apartado 4.02.2.2

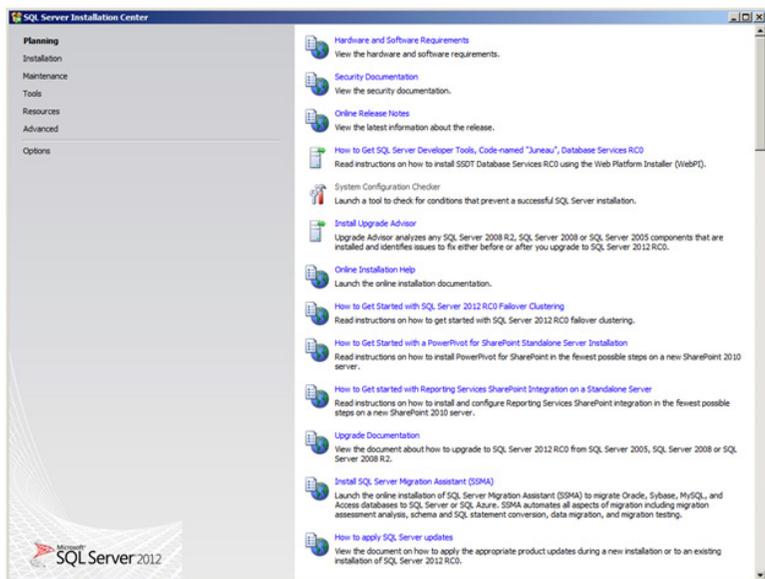
Implementación de servidor de base de datos relacionales

Para la implementación de una base de datos relacional, se estará utilizando de ejemplo SQL Server, corriendo por todo el proceso de instalación y configuración que este requiere para lo cual se procede con los siguientes pasos:



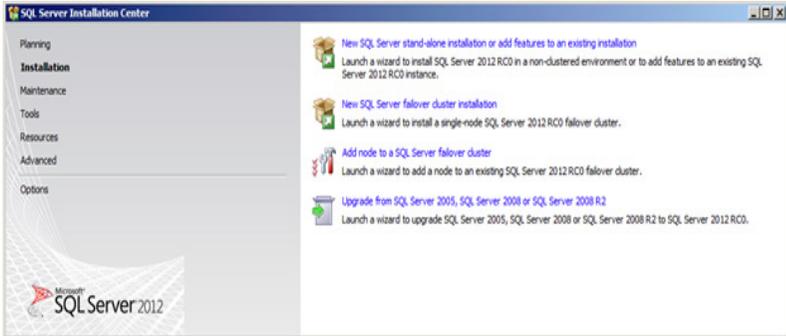
1. Correr el instalador con el nombre “**Setup.exe**”, este instalador debe de ejecutarse con permisos de administrador, para lo cual se da clic derecho sobre el archivo “**setup.exe**” y en el menú contextual se selecciona “**Correr como administrador**”, luego de esto se abrirá la ventana de instalación como se muestra en la **Figura No. 55 Pantalla de planificación de la instalación.**

Figura No. 55 Pantalla de planificación de la instalación.



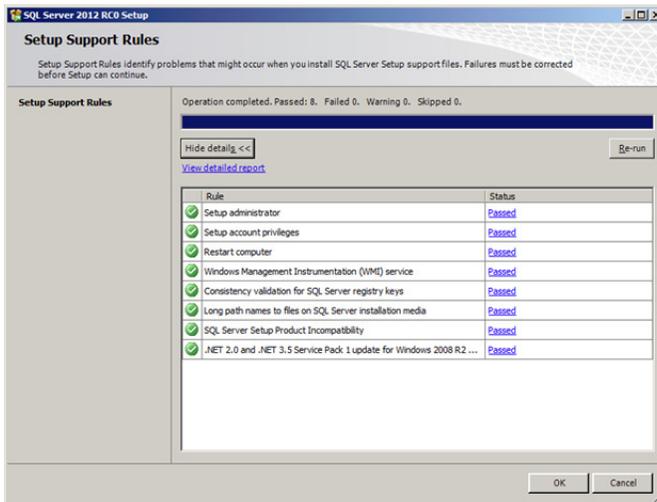
- En el panel izquierdo, se da clic en “Installation”, y luego en el panel derecho, clic en “New SQL Server stand-alone installation or add features to an existing installation” como se muestra en la **Figura No. 56 Pantalla de instalación.**

Figura No. 56 Pantalla de instalación.



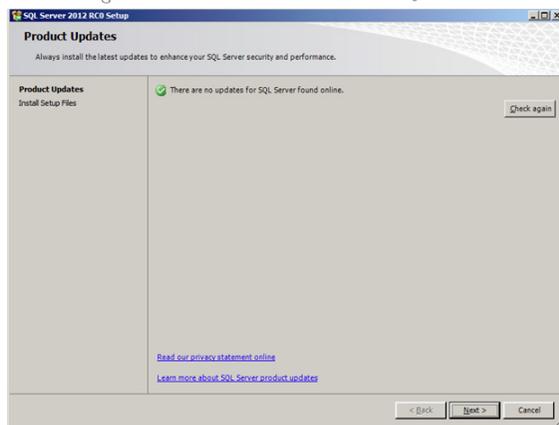
- Luego de esperar una revisión del instalador, se da clic en “OK” para comenzar el proceso de instalación como se muestra en la **Figura No. 57 Pantalla de revisión.**

Figura No. 57 Pantalla de revisión.



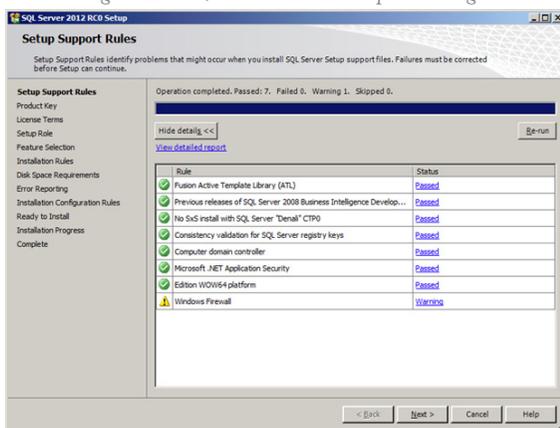
4. Luego de esto, el instalador intentará buscar actualizaciones, si las encuentra las instalará, de lo contrario se da clic en “Next” para proceder con el próximo paso como se muestra en la **Figura No. 58 Pantalla de actualización.**

Figura No. 58 Pantalla de actualización.



5. Si el cortafuegos⁽⁵⁴⁾ está habilitado en el servidor, saldrá el aviso de precaución acompañado de acciones sugeridas para la correcta configuración de su cortafuegos. Luego de esto se puede proceder con la instalación para lo cual se da clic en el botón “Next” como se muestra en la **Figura No. 59 Pantalla de chequeo de reglas.**

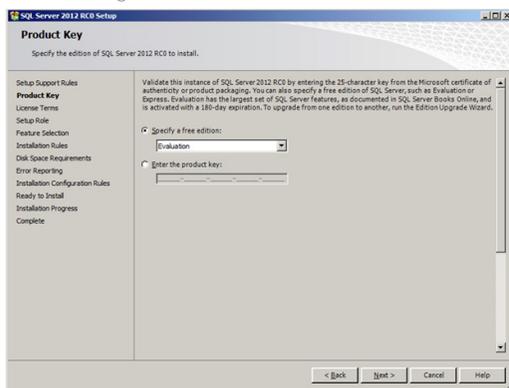
Figura No. 59 Pantalla de chequeo de reglas.



(54) Es un equipo diseñado para bloquear los accesos no permitidos a una red.

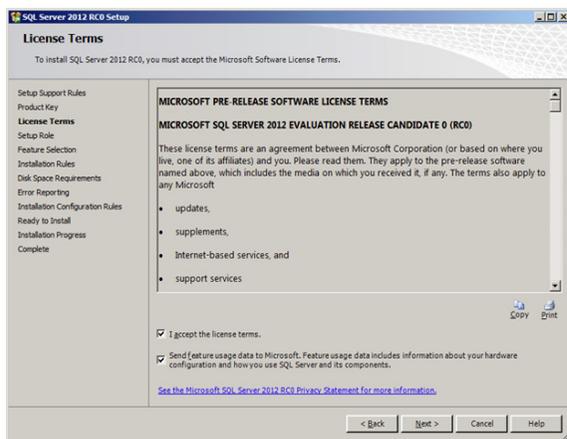
6. En esta pantalla se debe seleccionar el tipo de licenciamiento que se tiene, para este ejemplo, se selecciona un licenciamiento de evaluación y luego se da clic en “Next” como se muestra en la **Figura No. 60 Selección de licencia.**

Figura No. 60 Selección de licencia.



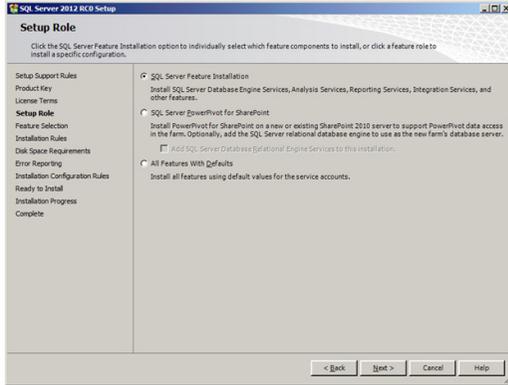
7. Luego se aceptan los términos de licenciamiento y se indica si se quiere enviar información sobre el desempeño del programa a su proveedor para mejorar la experiencia en próximas actualizaciones; luego se da clic en “Next” como se muestra en la **Figura No. 61 Términos de licencia.**

Figura No. 61 Términos de licencia.



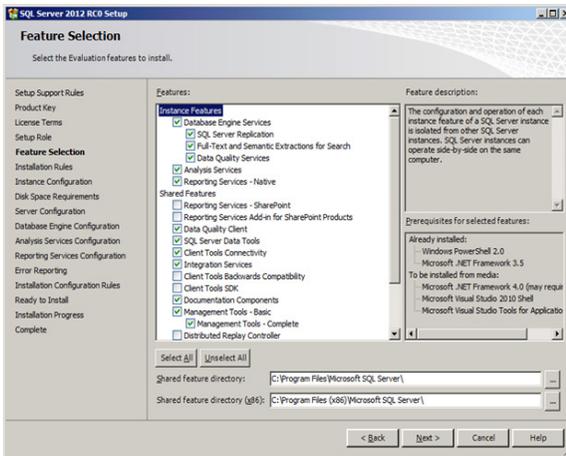
8. Se selecciona la opción “SQL Server Feature Installation” y se da clic en “Next” como se muestra en la **Figura No. 62 Selección de características de la instalación.**

Figura No. 62 Selección de características de la instalación.



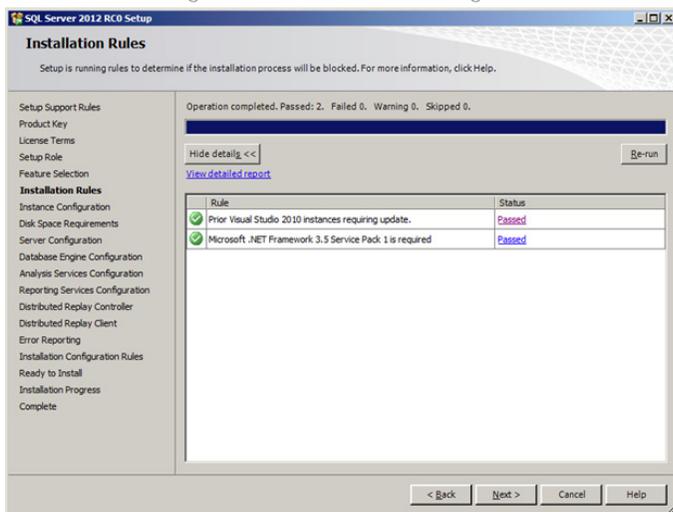
9. Luego se procede a seleccionar en esta pantalla las características que tendrá el servidor de base de datos. Entonces se selecciona del listado “Analisis Services”, componente que administra bases de datos cúbicas, “Reporting Services”, componente para generar los reportes web y por último “Integration Services” componente para crear paquetes de integración de datos. Luego de esto se da clic en “Next” como se muestra en la **Figura No. 63 Selección de características.**

Figura No. 63 Selección de características.



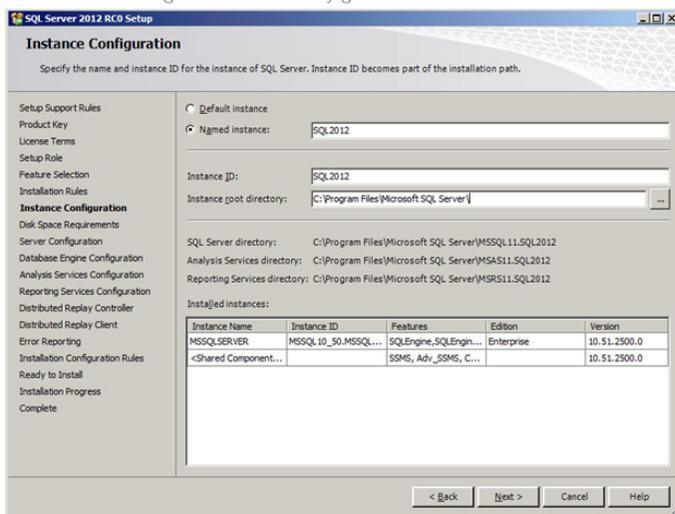
10. Se presiona “Next” como se muestra en la **Figura No. 64** Instalación de reglas.

Figura No. 64 Instalación de reglas.



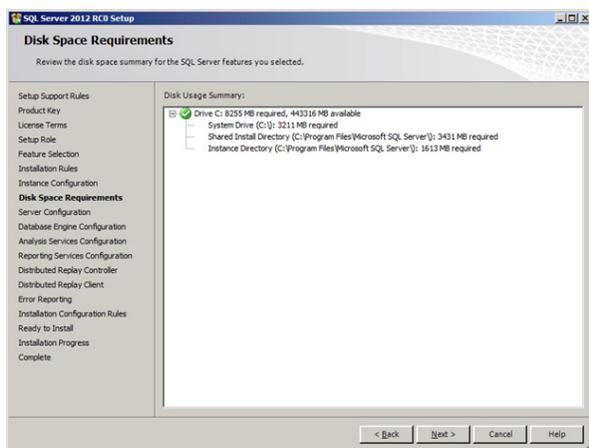
11. Se procede a especificar un nombre de la instancia de nuestro gestor de base de datos y luego se da clic en el botón “Next” como se muestra en la **Figura No. 65 Configuración de instancia.**

Figura No. 65 Configuración de instancia.



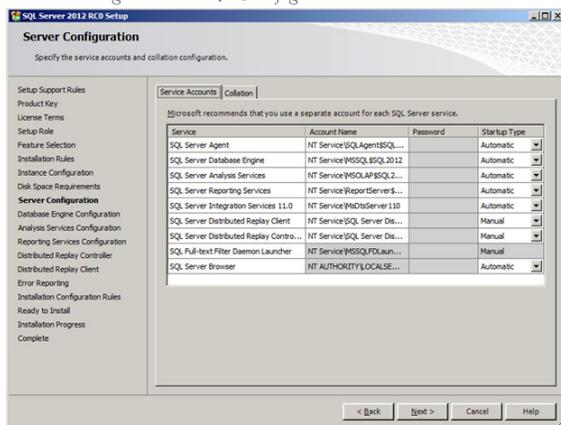
- En la próxima pantalla se tendrá un resumen de los requerimientos de espacio en el disco duro del servidor, el instalador verifica que el servidor cumpla con los requerimientos y luego se da clic en “Next” como se muestra en la **Figura No. 66** Requerimientos de espacio.

Figura No. 66 Requerimientos de espacio.



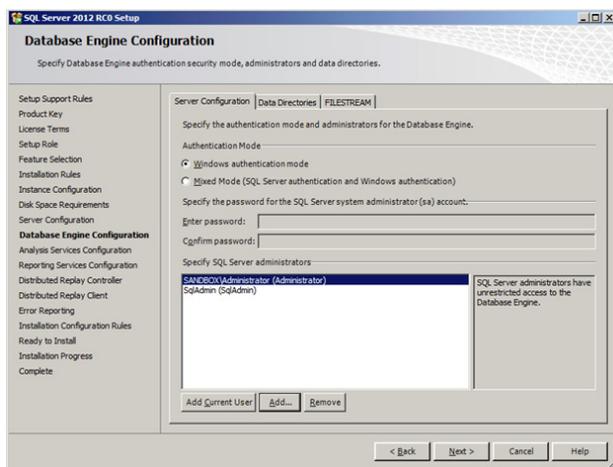
- En la siguiente pantalla se procede a configurar los usuarios por cada servicio de este gestor de base de datos y asignarle contraseñas a cada servicio, luego terminar con la configuración, se da clic en “Next” como se muestra en la **Figura No. 67** Configuración de servidor.

Figura No. 67 Configuración de servidor.



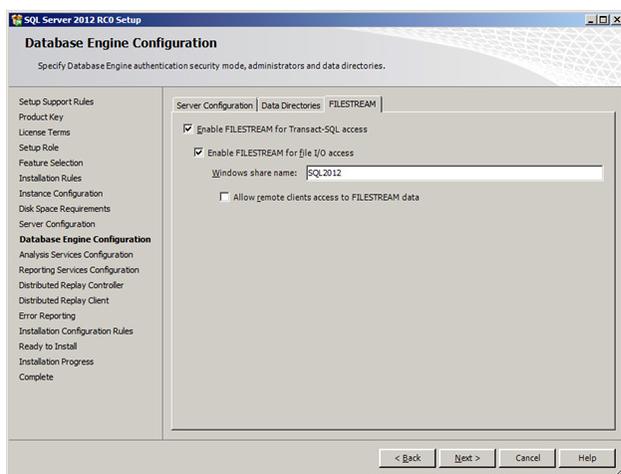
14. Luego de haber configurado las credenciales para los servicios, se procede a configurar en esta pantalla una cuenta de administración como mejor práctica y se da clic en “Next” como se muestra en la **Figura No. 68 Configuración de base de datos y administración.**

Figura No. 68 Configuración de base de datos y administración.



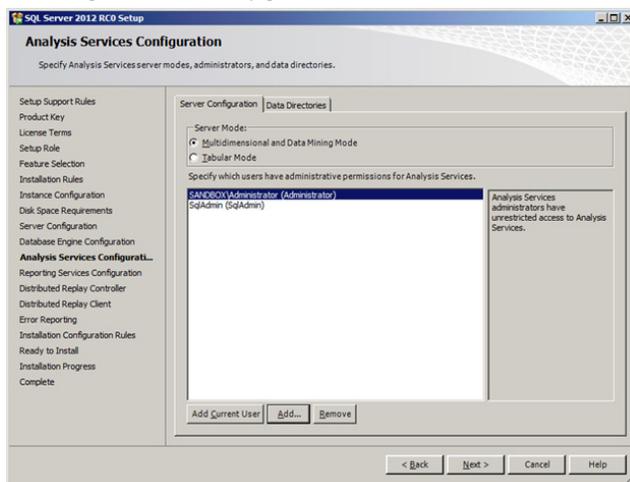
15. Luego ir a la pestaña “FILESTREAM” dando clic sobre ella, donde habilita esta función y luego se da clic en “Next” como se muestra en la **Figura No. 69 Configuración de accesos a archivos.**

Figura No. 69 Configuración de accesos a archivos.



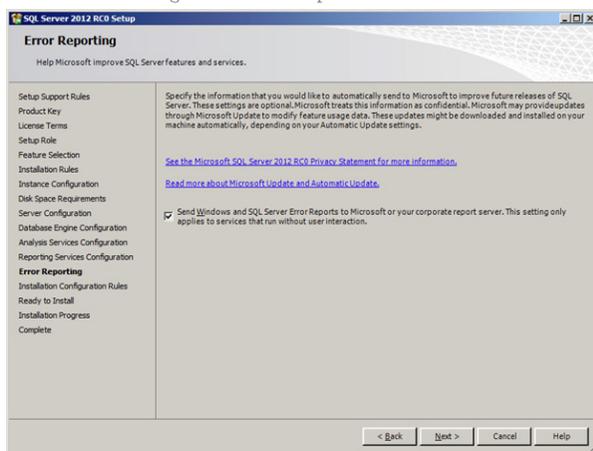
16. Luego en la pestaña “**Server configuration**” se selecciona “**Seleccionar Multidimensional and Data Mining Mode**” y se agregan las mismas cuentas de administración igual que en pantallas anteriores, luego se da clic en “**Next**” como se muestra en la **Figura No. 70 Configuración de servicios de análisis**.

Figura No. 70 Configuración de servicios de análisis.



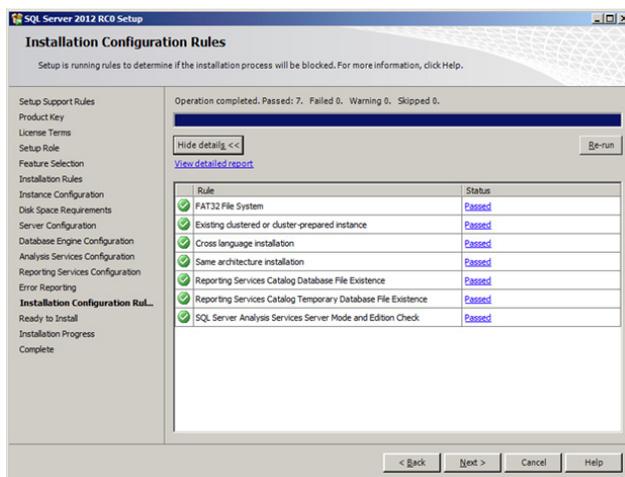
17. En esta pantalla se selecciona la casilla disponible para dar permiso al programa de enviar reportes de errores al proveedor, luego de la selección u omisión, se da clic en “**Next**” como se muestra en la **Figura No. 71 Reporte de errores**.

Figura No. 71 Reporte de errores.



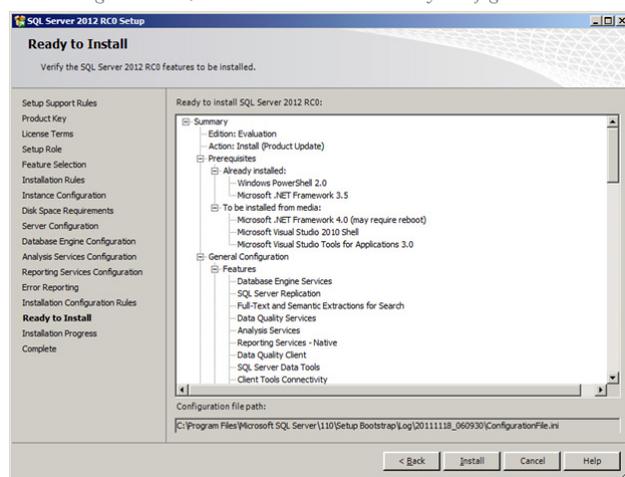
18. Luego el instalador corre validaciones adicionales para comenzar con la instalación, se procede a dar clic en “Next” como se muestra en la **Figura No.72 Verificación pre-instalación.**

Figura No. 72 Verificación pre-instalación.



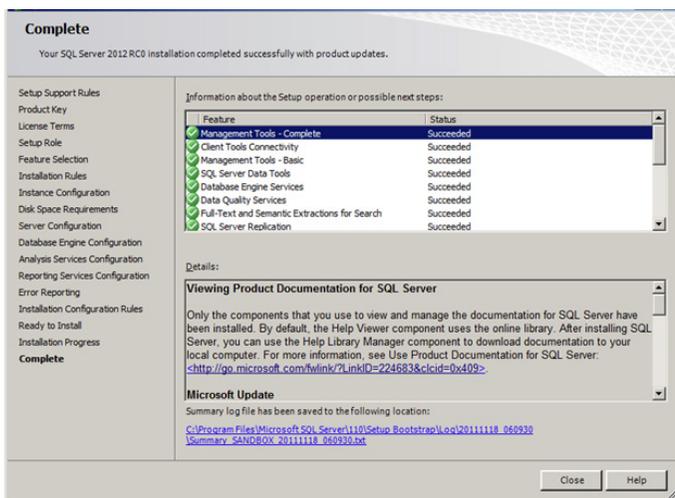
19. Finalmente se tiene un resumen de todo el proceso con todas las selecciones y configuraciones realizadas, luego de revisarlas, se procede a dar clic en “Install” como se muestra en la **Figura No. 73 Resumen de selección y configuración.**

Figura No. 73 Resumen de selección y configuración.



20. Una vez terminado el proceso de instalación, se nos presentará una última pantalla con un reporte de los componentes instalados, ya sean correctamente instalados, no instalados o con advertencias, luego se procede a dar clic en “Close” y para terminar el proceso de instalación como se muestra en la **Figura No.74 Reporte de instalación de componentes.**

Figura No. 74 Reporte de instalación de componentes.



Apartado 4.02.2.3

Bases de datos no relacionales

Dentro de las bases de datos no relacionales, existen varias categorías a las cuales se puede referir. Para fines de esta guía se hablara solamente de las bases de datos orientadas a objetos⁽⁵⁵⁾, las cuales son aquellas que incorporan todos los conceptos del paradigma de objetos como son:

- **Encapsulación:** Es la propiedad que permite ocultar informaciones al resto de los objetos, para evitar accesos incorrectos o conflictos.

(55) Es un elemento de datos que consta de un estado y comportamiento el cual se le asigna funcionalidades o tareas realizables



- **Herencia:** Es la propiedad que permite a los objetos heredar los atributos de los cuales dependen en su jerarquía de estructura.
- **Polimorfismo:** Es la propiedad que permite la aplicación de una operación a más de un objeto.

Las bases de datos orientadas a objetos, permiten definir operaciones sobre los datos como parte de la definición de la base de datos. Dichas operaciones, llamadas también funciones se especifican en dos partes:

- **La interfaz:** La visual de interacción de una operación que incluye el nombre de la operación y los tipos de datos de sus argumentos o parámetros.
- **La implementación o método:** Esta da especificaciones separadamente y puede modificarse sin afectar la interfaz. Los programas de aplicación pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sin importar la manera de su implementación lo que se denomina como independencia entre programas y operaciones; este es el modelo utilizado por los sistemas gestores de datos tales como MongoDB, CouchDB, Redis, entre otros.

La estructura de una base de datos orientada a objetos se basa prácticamente en el encapsulamiento de datos y del código que está directamente relacionado con cada objeto en una sola unidad. Conceptualmente, todas las interacciones entre cada objeto y el resto del sistema se realizan mediante mensajes, debido a esto la interfaz entre cada objeto y el resto del sistema se define mediante un conjunto de mensajes permitidos, donde los objetos están asociados con los siguientes elementos:



- Un conjunto de variables que contiene los datos del objeto, donde las variables corresponden con los atributos del modelo “**Entidad-relación (E-R)**, por siglas en inglés)”.
- Un conjunto de mensajes a los que responde donde cada mensaje puede o no tener parámetros.
- Un conjunto de métodos donde cada uno es el código que implementa un mensaje. Cada método devuelve un valor como respuesta al mensaje.

Las características básicas que no pueden faltar en una base de datos no relacional son las siguientes:

- **Predominancia combinada con enlace retardado:** Se puede definir que sea Excel, Autocad, entre otros. Desde la programación.
- **Extensibilidad:** Proporciona los tipos de datos como: Carácter, booleano, string, entre otros.
- **Concurrencia:** Permite que varios usuarios tengan acceso a una base de datos al mismo tiempo.
- **Recuperación:** Cuando se hace una transacción, pero esta no se puede realizar, se permite regresar al estado anterior.
- **Facilidad de “Consultas a Modo”:** Esto es muy útil cuando se tienen implementaciones en diferentes estándares.

Ventajas de una base de datos no relacional:

- Proporciona herramientas para asegurar la no duplicidad de registros.



- Proporciona garantía de la integridad referencial, de modo que si se elimina un elemento, también se eliminarán los elementos que tengan una relación dependiente con este.
- Proporciona mejor normalización por ser más comprensible y aplicable.

Desventajas de una base de datos no relacional:

- Es deficiente al momento de trabajar con datos gráficos, multimedia y sistemas de información geográfica.
- No se manipulan de forma manejable los bloques de texto como tipo de dato.

SECCIÓN 4.03

ARCHIVOS PARA ALMACENAMIENTO DE DATOS

Los archivos planos son una colección de información que está almacenada y es accedida de forma organizada en una base de datos. Se suele usar para almacenar información en una forma no estructurada. Estos archivos generalmente están sustentados por algún estándar de codificación, tales como: el Código Estándar Estadounidense para el Intercambio de Información (ASCII, por sus siglas en inglés), la norma ISO 8859-1, de la Organización Internacional de Normalización (ISO, por sus siglas en inglés), que define la codificación del alfabeto latino y el Formato de Transformación Unicode de 8-bit (UTF-8⁽⁵⁶⁾, por sus siglas en inglés).

Los archivos de almacenamiento ayudan a su vez a establecer de forma implícita la interoperabilidad entre sistemas, por ello es indispensable que cada aplicación o solución tecnológica que se desarrolle tenga facultades, tanto de importar como de exportar a uno o varios de los diferentes tipos de archivos de almacenamiento que se nombran a continuación, los cuales son los más utilizados en la actualidad.

(56) Es un formato estándar de codificación de caracteres capaz de representar cualquier carácter. Por sus características es recomendado para ser utilizado en la codificación de correos electrónicos y páginas web.



Sub-sección 4.03.1 Tipos de archivos de almacenamiento

En esta sub-sección se muestran los diferentes tipos de formatos recomendados para el almacenamiento de datos, así como también ejemplos de los mismos.

Apartado 4.03.1.1 Formato de notación de objetos de JavaScript

El formato de Notación de Objetos de JavaScript (JSON⁽⁵⁷⁾, por sus siglas en inglés), es un formato ligero de intercambio de datos, permitiendo de manera fácil leerlo y escribirlo para las personas, mientras que para las máquinas interpretarlo y generarlo es simple, se basa en sub-conjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edición.

JSON está compuesto por dos estructuras:

- Una colección de pares de nombre/valor, que en varios lenguajes se puede conocer de la siguiente manera: objeto, registro, estructura, diccionario, tabla hash⁽⁵⁸⁾, lista de claves o un arreglo⁽⁵⁹⁾ asociativo.
- Una lista ordenada de valores, esto se implementa como arreglos, vectores⁽⁶⁰⁾, listas o secuencias en la mayoría de los lenguajes.

Estas estructuras universales; virtualmente todos los lenguajes de programación las soportan, es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras.

(57) Es un formato ligero usado como alternativa al XML para intercambio de datos.

(58) Es una estructura de datos donde se asocia llaves o claves con valores.

(59) Es una estructura de datos, el cual sirve para almacenar y manejar un número "n" de elementos, todos del mismo tipo.

(60) También conocido como matriz, es una zona de almacenamiento continuo, donde se presentan una serie de elementos de la misma clase.



Sintaxis de JSON:

- JSON, se presenta de las siguientes maneras:
 - Objeto, siendo este un conjunto desordenado de pares nombre/valor. Ver **Anexo C. Diagrama de la sintaxis⁽⁶¹⁾ de JSON representada en “Objeto”**.
 - Arreglo, siendo este una colección de valores. Ver **Anexo D. Diagrama de la sintaxis de JSON representada en “Arreglo”**.
 - Valor, siendo este una cadena⁽⁶²⁾ de caracteres pudiendo el mismo anidarse. Ver **Anexo E. Diagrama de la sintaxis de JSON representada en “Valor”**.
 - Cadena, siendo este una colección de cero o más caracteres Unicode, encerrados entre comillas dobles. Ver **Anexo F. Diagrama de la sintaxis de JSON representada en “Cadena”**.
 - Número, siendo el mismo, similar a un número “C” o “java”, excepto que no utilizan formatos octales y hexadecimales, los espacios en blanco pueden insertarse entre cualquier par de símbolos. Ver **Anexo G. Diagrama de la sintaxis de JSON representada en “Numero”**.
 - Notación, existe una variación del tipo de datos JSON, el cual se llama Notación de Objetos de JavaScript (JSONP, por sus siglas en inglés), es el acrónimo de JavaScript, es decir, una forma de extensión del mismo para soportar

(61) Es la instrucción gramática que determina la regla para la implementación de un código.

(62) Es una secuencia ordenada, compuesta de caracteres ligados entre sí, las cuales pueden estar asignadas a una variable.



llamadas entre dominios. Es mandatorio, utilizar este estándar cuando se realicen llamadas a La interfaz de programación de aplicaciones (API, por sus siglas en inglés), externas fuera del dominio en donde se ejecuta la aplicación que consume los datos transferidos en formato JSON. Ver **Anexo H. Diagrama de la sintaxis de JSON representada en “Notación”**.

Apartado 4.03.1.2

Lenguaje de marcas extensible

El Lenguaje de Marcas Extensible (XML, por sus siglas en inglés), se trata de un metalenguaje que se utiliza para definir algo acerca de otro. Es importante entender que XML fue diseñado para almacenar, transportar e intercambiar datos, así como también es significativo señalar que el lenguaje XML no fue diseñado para mostrar los datos, sino para establecer un formato estándar de transporte y almacenamiento de los mismos.

La utilidad del lenguaje XML en la industria de la tecnología es muy diversa, las utilidades más comunes de archivos en este formato son las siguientes:

- **Intercambio de datos:**
Este formato permite el intercambio de datos entre sistemas incompatibles. La conversión de los datos a XML permite reducir la complejidad y crear datos que pueden ser verificados por diferentes tipos de aplicaciones.
- **Habilitación de transacciones B2B:**
Este formato podría convertirse en el lenguaje principal para el intercambio de información financiera, a través de Internet. Actualmente están en desarrollo varias aplicaciones de Negocios a Negocios (B2B, por sus siglas en inglés).



- **Compartir datos:**

Este formato permite utilizar archivos de texto sin formato para compartir datos. Dado que los datos XML se almacenan en formato de texto plano⁽⁶³⁾, los mismos proporcionan una metodología independiente de la aplicación y el hardware⁽⁶⁴⁾ para compartir datos, esto permite que sea mucho más fácil crear datos para aplicaciones diferentes, así como también facilita la extensibilidad de los sistemas.

- **Almacenamiento de datos:**

XML permite utilizar archivos de texto sin formato para almacenar datos, así como también las funcionalidades de escritura y lectura del mismo, utilizando aplicaciones genéricas para la visualización de los datos.

- **Las principales tecnologías derivadas de XML son las siguientes:**

- **XSLT⁽⁶⁵⁾:** Utilizado para crear transformaciones de datos de un formato a otro.
- **XLS-FO:** Utilizado para extender las capacidades de estilos y formatos.
- **XPath⁽⁶⁶⁾:** Lenguaje de rutas XML, es un lenguaje para acceder a partes de un documento XML.

(63) También conocido como textos llanos, son archivos informáticos que están compuesto solo por texto sin ningún tipo de formato.

(64) Se refiere a todas las partes físicas o tangibles de un sistema de información.

(65) Es un estándar, del Consorcio World Wide Web (W3C, por sus siglas en inglés), que presenta como transformar los documentos de Lenguaje de Marcas Extensible (XML, por sus siglas en inglés), en otros formatos.

(66) Lenguaje de rutas XML, es un lenguaje para acceder a partes de un documento XML.



- o **XLink**⁽⁶⁷⁾: Lenguaje de Enlace XML, es un lenguaje que permite insertar elementos en documentos XML para crear enlaces entre recursos del mismo.
 - o **XPointer**⁽⁶⁸⁾: Lenguaje de direccionamiento XML, es un lenguaje que permite el acceso a la estructura interna de un documento XML, esto es, a sus elementos, atributos y contenido.
 - o **XQL**⁽⁶⁹⁾: Lenguaje de Consulta XML, es un lenguaje que facilita la extracción de datos desde documentos del mismo lenguaje, ofreciendo la posibilidad de realizar consultas flexibles para extraer datos de documentos del XML en la web.
- Sintaxis de XML
 - A continuación se presentan las siguientes reglas para crear la sintaxis XML:
 - o Todos los elementos deben contener una etiqueta de cierre.

(67) *Lenguaje de Enlace XML, es un lenguaje que permite insertar elementos en documentos XML para crear enlaces entre recursos del mismo.*

(68) *Lenguaje de direccionamiento XML, es un lenguaje que permite el acceso a la estructura interna de un documento XML, esto es, a sus elementos, atributos y contenido.*

(69) *Es un lenguaje que facilita la extracción de datos desde documentos del mismo lenguaje, ofreciendo la posibilidad de realizar consultas flexibles para extraer datos de documentos del XML en la web.*



Incorrecto:

```
<body>Voy caminando.  
<body>Voy corriendo.
```

Correcto:

```
<body > Voy caminando.</body>  
<body > Voy corriendo.</body>
```

- o Dichas etiquetas son sensibles a mayúsculas y minúsculas.

Incorrecto:

```
<body>Mira cómo corre.</body>
```

Correcto:

```
<body>Mira cómo corre.</body>
```

- o Los elementos XML están debidamente anidados.

Incorrecto:

```
<b><i>Este texto está en negrita y cursiva.</b></i>
```

Correcto:

```
<b><i>Este texto está en negrita y cursiva.</i></b>
```

- o Todos los documentos deben contener un elemento raíz.

```
<root>  
  <child>  
    <subchild>.....</subchild>  
  </child>
```

```
</root>
```

- o Los valores de atributo siempre deben estar entre comillas.

Incorrecto:

```
<?xml version= "1.0" encoding="ISO-8859-1"?>  
<note date=05/05/05>  
<to>Dick</to>  
<from>Jane</from>  
</note>
```

Correcto:

```
<?xml version= "1.0" encoding="ISO-8859-1"?>  
<note date="05/05/05">  
<to>Dick</to>  
<from>Jane</from>  
</note>
```

En el documento incorrecto, el atributo de fecha del elemento note no está entre comillas.

Apartado 4.03.1.3 Formato de valores separados por coma

Los archivos de Valores Separados por Coma⁽⁷⁰⁾ (CSV, por sus siglas en inglés), son un tipo de documento en formato abierto que representa los datos en forma de tabla, en las que las columnas están separadas por comas y las filas por saltos de línea.

Los campos que contengan una coma, un salto de línea o una comilla doble deben ser encerrados entre comillas dobles.

(70) Es un formato de archivo de datos que su contenido está separado por comas.



El formato CSV, no muestra un juego de caracteres concreto, tampoco como están situados los bytes, ni el formato para el salto de línea. Estos puntos deben indicarse al abrir el archivo, por ejemplo, con una hoja de cálculo.

Ejemplo del formato CSV:

```
1,0,0 0,1,0 -1,0,0 0,-1,0 1,0,0  
1,0,0 0,1,0 -1,0,0 0,-1,0 1,0,0
```

Apartado 4.03.1.4

Ficheros de valores separados por tabulaciones

Los ficheros de Valores Separados por Tabulaciones⁽⁷¹⁾ (TSV, por sus siglas en inglés), son un tipo de documento utilizados para representar datos en forma de tabla, donde las columnas se separan por tabulaciones y las filas por saltos de línea.

- A continuación se presentan las siguientes reglas para el formato TSV:
 - Cada entrada del archivo debe contener una única línea.
 - La primera línea del archivo es el encabezado y etiqueta cada campo.
 - Un campo debe contener datos, como un número o texto.
 - Los campos están separados por tabulaciones.
 - Cada línea contiene el mismo número de campos.

(71) Es un formato de texto simple utilizado para el almacenamiento de información en forma de tablas. En este, cada registro de la tabla representa una línea del archivo de texto.



Ejemplo de TSV:

Título<tabulación>Autor<tabulación>Estado<tabulación>Tapa

Orgullo y prejuicio<tabulación>Jane

Austen<tabulación>usado<tabulación>en pasta blanda

Las aventuras de Huckleberry Finn<tabulación>Mark

Twain<tabulación>nuevo<tabulación>tapa dura El arte de la

guerra<tabulación>Sun Tzu<tabulación>usado<tabulación>tapa

dura Ejemplo de cómo queda expresado en el archivo: Título

Autor Tapa Estado Orgullo y prejuicio Jane Austen usado tapa

blanda Las aventuras de Huckleberry Finn Mark Twain nuevo

tapa dura El arte de la guerra Sun Tzu usado tapa dura

SECCIÓN 4.04

DOCUMENTACIÓN DEL SISTEMA

En esta sección se detallan las pautas para la documentación de las aplicaciones o sistemas que se desarrollen en el organismo.

Cabe destacar, que un sistema pobremente documentado carece de valor aunque tenga un funcionamiento correcto y estable. En el caso de programas pequeños y menos importantes que solo se utilizan durante un corto periodo de tiempo, unos cuantos comentarios en el código podrían ser suficientes.

Sub-sección 4.04.1

Esquema documental

Se expresan un número de páginas de referencia con la cual la documentación se presenta de la siguiente manera:



1. Información General:

Describe el problema que se está resolviendo junto con su solución. Esta sección de la documentación no debería contener detalles sobre la estrategia de implementación como tal. Las otras partes de la documentación del sistema no serán de interés para los usuarios, únicamente para los desarrolladores, implementadores, los encargados del mantenimiento y demás personal.

- **Visión general (máximo 2 páginas).** Una explicación del objetivo del sistema y de la funcionalidad del mismo.
- **Especificación.** Especificaciones detalladas del comportamiento del sistema.
- **Manual de usuario (máximo 5 páginas).** Es una descripción detallada sobre cómo el usuario puede utilizar el sistema y qué operaciones puede realizar. Con esta información y la visión general, este manual proporciona toda la información que un usuario del sistema necesita.
- **Requisitos (máximo 1 página).** Requerimientos de hardware y software. Adicionalmente describe las capacidades y conocimientos mínimos que debe tener el usuario para manejar el sistema.
- **Análisis del problema (máximo 10 páginas).** Es una descripción clara del problema que resuelve el sistema. El análisis del problema generalmente incluye uno o más modelos de entidades de negocio que definen las responsabilidades de cada una y las interacciones que tienen entre ellas. La documentación de esto pudiera hacerse en una notación estándar como un **Lenguaje Unificado de Modelado (UML)**⁽⁷²⁾, por sus siglas en inglés).

(72) Es un lenguaje gráfico que visualiza, especifica, construye y documenta un sistema.



2. Diseño:

La documentación de diseño brinda una dirección de lo que es la implementación.

Introducción (máximo 3 páginas). Es una breve explicación de lo que se espera del producto. Se destacan librerías propias, marcos de trabajo y componentes de terceros, así como también tecnologías utilizadas.

Estructura en tiempo de ejecución (máximo 5 páginas). Es una representación de la estructura del estado del programa en ejecución, expresada como un modelo de objeto de código. Éste debería ocultar las representaciones de los tipos de datos abstractos; su propósito consiste en mostrar las relaciones entre objetos. Los modelos de objeto incluyen tanto diagramas como cualquier restricción textual fundamental, y estar claramente expuestos para una correcta legibilidad. Las representaciones de los tipos de datos se explican con sus funciones de abstracción e invariantes de representación.

Estructura del módulo (máximo 5 páginas). Es una descripción de la estructura sintáctica del texto del programa, expresada como un diagrama de dependencia entre módulos. Su **Diagrama de Dependencia entre Módulos (MDD)**, por sus siglas en inglés) debería estar claramente expuesto para una correcta legibilidad.

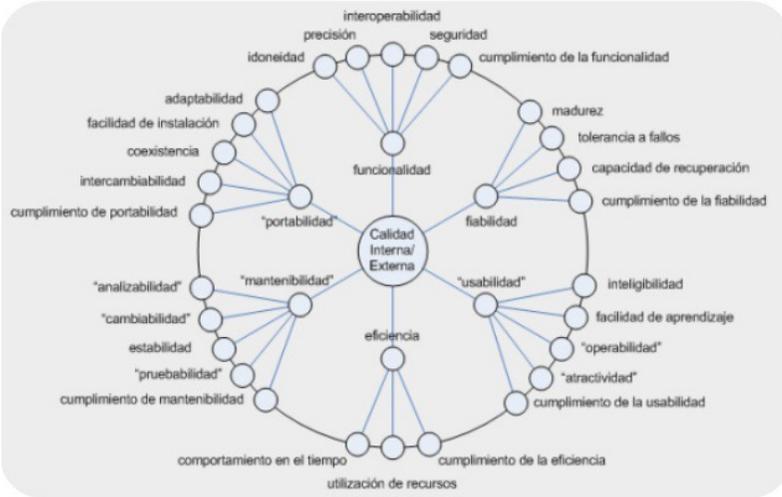
Factores de Calidad de Software [ISO 9126] (máximo 5 páginas). Uno de los aspectos más importante del diseño en la ingeniería de software es el cómo se expresan y se toman en cuenta los lineamientos que pueden garantizar la calidad del software. Pare ello tomar como referencia el estándar ISO 9126, el cual presenta los principios a tomarse en cuenta y así mismo ingresarlos a la documentación de diseño. El estándar identifica 6 atributos principales clave de calidad:



- **Funcionalidad:** El grado en que el software satisface las necesidades indicadas por los siguientes subatributos: idoneidad, corrección, interoperatividad, conformidad y seguridad.
- **Confiabilidad:** Cantidad de tiempo que el software está disponible para su uso. Está referido por los siguientes subatributos: madurez, tolerancia a fallos y facilidad de recuperación.
- **Usabilidad:** Grado en que el software es fácil de usar. Viene reflejado por los siguientes subatributos: facilidad de comprensión, facilidad de aprendizaje y operatividad.
- **Eficiencia:** Grado en que el software hace óptimo el uso de los recursos del sistema. Está indicado por los siguientes subatributos: tiempo de uso y recursos utilizados.
- **Facilidad de mantenimiento:** La facilidad con que una modificación puede ser realizada. Está indicada por los siguientes subatributos: facilidad de análisis, facilidad de cambio, estabilidad y facilidad de prueba.
- **Portabilidad:** La facilidad con que el software puede ser llevado de un entorno a otro. Está referido por los siguientes subatributos: facilidad de instalación, facilidad de ajuste, facilidad de adaptación al cambio.

En el presente diagrama se expresa la jerarquía y relaciones entre factores:

Figura No. 75 Jerarquía y relaciones entre factores.



3. Pruebas:

La documentación de pruebas indica el enfoque que se ha elegido para verificar y validar el sistema. Existen tres niveles de realización de pruebas:

- **Pruebas con el usuario:** Son las actividades de verificación que se realizan con el usuario. Es necesario documentar estas actividades en un formato que exprese los siguientes campos: la fecha, descripción de la actividad, nombre del usuario y resultado (satisfactorio y no satisfactorio).
- **Pruebas de código en tiempo de diseño:** Son las actividades dirigidas a conseguir la mayor calidad del software. Estas pruebas se pueden basar en alguna de las metodologías para estos fines tales como: **Desarrollo Guiado por Pruebas de**



Software (TDD⁽⁷³⁾, por sus siglas en inglés), o **Desarrollo Guiado por Comportamiento (BDD⁽⁷⁴⁾**, por sus siglas en inglés). Se pueden documentar los casos de prueba de cada una de las funciones, en un formato que exprese los siguientes campos: nombre de la función, descripción, código, parámetros, resultado y versión.

- **Pruebas de la solución en tiempo de ejecución:** Son actividades que se realizan utilizando herramientas de “Automatización de Pruebas” que permiten obtener valores estadísticos de ejecución, al mismo tiempo estas herramientas graban las pruebas para que estas puedan ser reproducidas para su solución y finalmente documentadas mediante pantallas tomadas de las pruebas con una breve explicación de las mismas. Algunas herramientas de pruebas recomendadas son: TestCafé, Telerik y Test Studio como aplicaciones comerciales y gratuita Selenium.

4. Retrospectiva:

En la documentación de la retrospectiva o auto evaluación es donde se describe la forma en la que el equipo realizó su trabajo y los acontecimientos que sucedieron, tanto positivos como negativos, lo cual se deriva en un listado de lecciones aprendidas.

- **Evaluación (máximo 1 página).** Listado de éxitos y/o fracasos del desarrollo: problemas de diseño no resueltos, problemas de funcionamiento, limitaciones de integración con otros sistemas no resueltos. Este incluye un listado de

(73) Es una técnica de programación que consiste en desarrollar el código que pruebe una característica o funcionalidad deseada antes del código que implementa dicha funcionalidad.

(74) Es un proceso que amplía las ideas de TDD y las une con otras ideas de diseño de software para proporcionar un proceso a los desarrolladores para que puedan mejorar el desarrollo del software.



lecciones aprendidas de la experiencia presente, así como los factores que causaron problemas, como hitos errados, o errores y limitaciones conocidos.

- **Errores y limitaciones conocidas.** En reconocimiento de lo que no fue solucionado es necesario contribuir a la maduración del desarrollo del software y la forma de trabajo. Para el desarrollo de esta parte se escribe un listado de todos los errores o limitaciones, además de adicionar por cual razón no se pudieron resolver, así cabe destacar el impacto que tendrá corregir esto en una próxima actualización del sistema.

Se contemplan los siguientes campos: Nombre del error, descripción, funcionalidad que lo presenta, fecha, razón humana (de no resolución del error) y razón técnica (de no resolución del error).

SECCIÓN 4.05 MANEJO DE CÓDIGO FUENTE CON GIT

En la siguiente sección se muestran los sistemas utilizados para controlar el versionamiento⁽⁷⁵⁾, los cuales son aplicaciones que tienen como función controlar y administrar los cambios en el desarrollo de cualquier clase de software, como es el caso de Git.

Sub-sección 4.05.1

Instalación de Git

Git puede ser instalado desde código fuente, o instalar un paquete existente para la plataforma.

(75) Es el proceso de administración y gestión de los distintos cambios que experimento un código fuente, producto o elemento.



Apartado 4.05.1.1

Instalación desde código fuente

Cuando se opta por este tipo de instalación se obtiene la versión más reciente. Cada versión de Git tiende a incluir útiles mejoras en la interfaz de usuario⁽⁷⁶⁾, por lo que se recomienda usar la última versión.

Para instalar Git, se necesita de las siguientes librerías: curl, zlib, openssl, expat y libiconv. Por ejemplo, en un sistema que tiene yum (como Fedora⁽⁷⁷⁾) o apt-get (como un sistema basado en Debian⁽⁷⁸⁾), se puede usar estos comandos para instalar todas las dependencias:

```
$ yum install curl-devel expat-devel gettext-devel \  
  openssl-devel zlib-devel  
$ apt-get install libcurl4-gnutls-dev libexpat1-dev gettext \  
  libz-dev libssl-dev
```

Cuando se tenga todas las dependencias necesarias, se puede descargar la versión más reciente de Git visitando su página web:

<http://git-scm.com/download>

Luego de compilar⁽⁷⁹⁾ e instalar, es necesario ejecutar los siguientes comandos:

```
$ tar -zxf git-1.6.0.5.tar.gz  
$ cd git-1.6.0.5  
$ make prefix=/usr/local all  
$ sudo make prefix=/usr/local install
```

(76) Es el medio por el cual el usuario puede interactuar con un dispositivo o computador.

(77) Es una distribución del sistema operativo linux, desarrollado en base a código libre.

(78) Debian es un sistema operativo (S.O) libre, para su computadoras.

(79) Consiste en traducir mediante un software o herramienta las instrucciones de un lenguaje de programación aun lenguaje entendible por la máquina.



También se puede obtener Git, a través del propio Git, para futuras actualizaciones:

```
$ git clone git://git.kernel.org/pub/scm/git/git.git
```

Apartado 4.05.1.2

Instalación en Linux

La instalación se puede hacer a través de la herramienta básica de gestión de paquetes que trae la distribución. Si esta se encuentra en Fedora, se puede usar yum:

```
$ yum install git-core
```

O si es una distribución basada en Debian como Ubuntu⁽⁸⁰⁾, prueba con apt-get:

```
$ apt-get install git
```

Apartado 4.05.1.3

Instalación en Mac

Se puede usar el instalador gráfico de Git, descargandolo desde la página de SourceForge:

<http://sourceforge.net/projects/git-osx-installer/>

Otra alternativa es instalar Git a través de MacPorts, el cual puede ser descargo visitando este enlace:

<http://www.macports.org>

(80) Es un sistema operativo de código abierto que funciona con funcionalidad para trabajar en diferentes plataformas.



Teniendo MacPorts instalado, se procede a instalar Git con:

```
$ sudo port install git-core +svn +doc +bash_completion +gitweb
```

Apartado 4.05.1.4

Instalación en Windows

Descargar el archivo .exe del instalador desde la página de GitHub, y ejecutarlo:

<http://msysgit.github.com/>

Una vez instalado, se tiene tanto la versión de línea de comandos (incluido un cliente SSH que será de utilidad más adelante) como la interfaz gráfica de usuario estándar.

Para el uso en Windows: Se puede usar Git con la shell provista por msysGit (estilo Unix), lo cual permite un mejor uso en las líneas de comando. Si por cualquier razón se necesita usar la shell nativa de Windows, la consola de línea de comandos, se usan las comillas dobles en vez de las simples (para parámetros que contengan espacios) y entrecomillar los parámetros terminándolos con el acento circunflejo (^) si están al final de la línea, ya que en Windows es uno de los símbolos de continuación.

Sub-sección 4.05.2

Configurando Git por primera vez

Git cuenta con una herramienta llamada git config que permite obtener y establecer variables de configuración, que controlan el aspecto y funcionamiento de Git. Estas variables pueden almacenarse en tres sitios distintos:

- Archivo `/etc/gitconfig`: Contiene valores para todos los usuarios del sistema y todos sus repositorios. Seleccionando la opción `~system` a git config, se puede leer y escribir específicamente en este archivo.



- Archivo `~/.gitconfig` file: Específico al usuario. Se puede hacer que Git lea y escriba específicamente en este archivo seleccionando la opción `~global`.
- Archivo **config**: En el directorio de Git (es decir, `.git/config`) del repositorio que se está utilizando actualmente: Cada nivel sobrescribe los valores del nivel anterior, por lo que los valores de `.git/config` tienen preferencia sobre los de `/etc/gitconfig`.

En sistemas Windows, Git busca el archivo `.gitconfig` en el directorio `$HOME` (`%USERPROFILE%` in Windows' environment), que es `C:\Documents and Settings\%USER` para la mayoría de usuarios, dependiendo de la versión (`$USER` es `%USERNAME%` en el entorno Windows). También busca en el directorio `/etc/gitconfig`, aunque esta ruta es relativa a la raíz MSys, que es donde quiera que se decida instalar Git en el sistema Windows, cuando se ejecutó el instalador.

Apartado 4.05.2.1

Personalizando la identidad

Para personalizar el nombre de usuario y dirección de correo electrónico. Esto es importante porque las confirmaciones de cambios (commits) en Git usan esta información, y es introducida de manera inmutable en los commits que se envían:

```
$ git config --global user.name "John Doe"  
$ git config --global user.email johndoe@example.com
```



Solo es necesario hacerlo una vez especificando la opción `--global`, ya que Git siempre usa esta información para todo lo que se realice en ese sistema. En caso de sobrescribir esta información con otro nombre o dirección de correo para proyectos específicos, se puede ejecutar el comando sin la opción `--global` cuando se esté en ese proyecto.

Apartado 4.05.2.2

Personalizando el editor

Para elegir el editor de texto por defecto que se utiliza cuando Git necesite que se introduzca un mensaje. En caso de no indicar nada, Git usa el editor por defecto del sistema, que generalmente es Vi o Vim. Para usar otro editor de texto, como Emacs, se puede hacer lo siguiente:

```
$ git config --global core.editor emacs
```

Apartado 4.05.2.3

Herramienta de diferencias

Es la herramienta de diferencias por defecto, usada para resolver conflictos de unión (merge). En ese caso, para usar `vimdiff`:

```
$ git config --global merge.tool vimdiff
```

Git acepta `kdiff3`, `tkdiff`, `meld`, `xxdiff`, `emerge`, `vimdiff`, `gvimdiff`, `ecmerge`, y `opendiff` como herramientas válidas.

Apartado 4.05.2.4

Comprobando la configuración

Para comprobar la configuración, se puede usar el comando `git config --list` para listar todas las propiedades que Git ha configurado:

```
$ git config --list  
user.name=Scott Chacon  
user.email=schacon@gmail.com
```



```
color.status=auto
color.branch=auto
color.interactive=auto
color.diff=auto
```

Puede que se vean claves repetidas, porque Git lee la misma clave de distintos archivos (/etc/gitconfig y ~/.gitconfig, por ejemplo). En ese caso, Git usa el último valor para cada clave única que ve.

También se puede comprobar qué valor cree Git que tiene una clave específica ejecutando git config {clave}:

```
$ git config user.name
```

Sub-sección 4.05.3

Obtener Ayuda

Para utilizar la ayuda en Git, existen tres formas de ver la página del manual para cualquier comando de Git:

```
$ git help <comando>
$ git <comando> -help
$ man git-<comando>
```

Por ejemplo, se puede ver la página del manual para el comando config ejecutando:

```
$ git help config
```

Apartado 4.05.3.1

Obteniendo un repositorio Git

Para obtener un proyecto Git existen de dos maneras. La primera tomando un proyecto o directorio existente y se importa en Git. La segunda clonar un repositorio Git existente desde otro servidor.



Apartado 4.08.3.2 Inicializando un repositorio en un directorio existente

Para empezar el seguimiento en Git de un proyecto existente, se necesita ir al directorio del proyecto y escribir:

```
$ git init
```

Esto crea un nuevo subdirectorio llamado `.git` que contiene todos los archivos necesarios del repositorio, un esqueleto de un repositorio Git.

Para empezar a controlar versiones de archivos existentes (a diferencia de un directorio vacío), probablemente se debería comenzar el seguimiento de esos archivos y hacer una confirmación inicial. Se puede conseguir con los comandos `git add` para especificar qué archivos se quiere controlar, seguidos de un “commit” para confirmar los cambios:

```
$ git add *.c  
$ git add README  
$ git commit -m 'versión inicial del proyecto'
```

Luego de este proceso se tiene un repositorio Git con archivos bajo seguimiento, y una confirmación inicial.

Apartado 4.05.3.2 Clonando un repositorio existente

Para obtener una copia de un repositorio Git existente, el comando que se necesita es `git clone`. Es una distinción importante, ya que Git recibe una copia de casi todos los datos que tiene el servidor. Cada versión de cada archivo de la historia del proyecto es descargado cuando se ejecuta `git clone`.



Por ejemplo, para clonar la librería Ruby llamada Grit, sería:

```
$ git clone git://github.com/schacon/grit.git
```

Esto crea un directorio llamado “grit”, se inicializando un directorio .git en su interior, descargando toda la información de ese repositorio, y saca una copia de trabajo de la última versión. Cuando se ingresa al nuevo directorio grit, están los archivos del proyecto, listos para ser utilizados.

Para clonar el repositorio a un directorio con otro nombre que no sea grit, se puede especificar con la siguiente opción de línea de comandos:

```
$ git clone git://github.com/schacon/grit.git mygrit
```

Ese comando hace lo mismo que el anterior, pero el directorio destino sería mygrit.. Git permite usar distintos protocolos de transferencia. El ejemplo anterior usa el protocolo git://, pero puede encontrarse con http(s):// o usuario@servidor:/ruta.git, que utiliza el protocolo de transferencia SSH.

Apartado 4.05.3.3 Guardando cambios en el repositorio

En todo momento se tiene un repositorio Git completo, y una copia de trabajo de los archivos de ese proyecto, por lo que es necesario hacer algunos cambios, y confirmar la entrada (cambios preparados para registrar o “commit”) de esos cambios al repositorio, cada vez que el proyecto alcance un estado que se desee grabar.

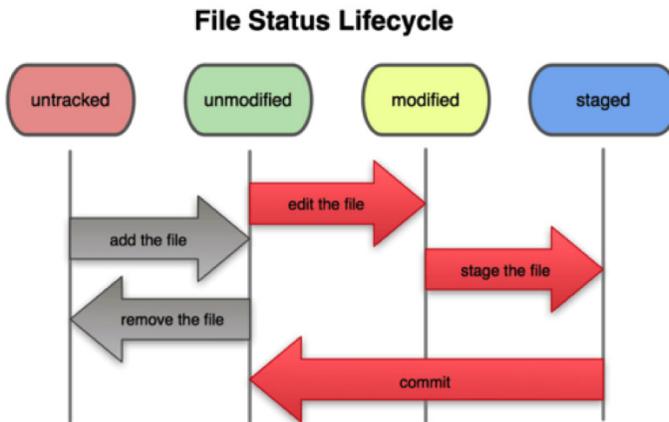
Es importante recordar que cada archivo del directorio de trabajo puede estar en uno de estos dos estados: bajo seguimiento (tracked), o sin seguimiento (untracked).

Los archivos bajo seguimiento son aquellos que existían en la última entrada; pueden estar sin modificaciones, modificados, o preparados.

Los archivos sin seguimiento son todos los demás archivos, cualquier archivo del directorio que no estuviese en la última entrada ni está en el área de preparación. La primera vez que se clona un repositorio, todos los archivos están bajo seguimiento y sin modificaciones, ya que se acaban de copiar y no se han realizado modificaciones.

A medida que se editan los archivos, Git los observa como modificados, porque se han cambiado desde la última confirmación. Se necesita preparar estos archivos modificados y luego confirmar que todos los cambios estén preparados como se visualiza en la **Figura No. 76 El ciclo de vida del estado de los archivos.**

Figura No. 76 El ciclo de vida del estado de los archivos.



Apartado 4.05.3.4 Comprobando el estado de los archivos

La principal herramienta para determinar qué archivos están en qué estado, es el comando **git status**. Para ejecutar este comando justo después de clonar un repositorio, ejecutar las siguientes líneas de comandos:



```
$ git status
# On branch master
nothing to commit, working directory clean
```

Esto significa que el directorio de trabajo está limpio, en otras palabras, no se tienen archivos bajo seguimiento y modificados. Git tampoco se muestra ningún archivo que no esté bajo seguimiento o listado.

Por último, el comando indica en qué rama se está. Por ahora, esa rama siempre es “**master**”, que es la predeterminada. Luego se añade un nuevo archivo al proyecto, un sencillo archivo README. Si el archivo no existía y se ejecuta **git status**, dicho archivo se muestra sin seguimiento de esta manera:

```
$ vim README
$ git status
# On branch master
# Untracked files:
# (use “git add <file>...” to include in what will be committed)
#
# README
nothing added to commit but untracked files present (use “git add”
to track)
```

Se puede observar que el nuevo archivo README aparece bajo la cabecera “Archivos sin seguimiento” (“Untracked files”) de la salida del comando. Sin seguimiento significa básicamente que Git ve un archivo que no estaba en la entrada anterior; Git no puede comenzar a incluirlo en las confirmaciones de sus entradas hasta que se indique explícitamente. Esto lo hace para que no incluya accidentalmente archivos binarios generados u otros archivos que no tenían intención de incluir.



Apartado 4.05.3.5

Seguimiento de nuevos archivos

Para el seguimiento de un nuevo archivo se usa el comando **git add**. Iniciando el seguimiento del archivo **README**, ejecutando esta sentencia:

```
$ git add README
```

De volver a ejecutar el comando **git status**, el archivo **README** se está ahora bajo seguimiento y preparado, como se muestra a continuación:

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#   new file:   README
#
```

Puede ver que está confirmado porque aparece bajo la cabecera “**Changes to be committed**”. De confirmar ahora, la versión del archivo en el momento de ejecutar **git add** es la que se incluya en la entrada.

Es importante recordar que cuando antes se ejecutó **git init**, seguidamente se ejecutó **git add**. Esto fue para iniciar el seguimiento de los archivos del directorio. El comando **git add** recibe la ruta de un archivo o de un directorio; si es un directorio, se añaden todos los archivos que contenga de manera recursiva.



Apartado 4.05.3.6

Preparando archivos modificados

Para modificar el archivo **benchmarks.rb** que estaba bajo seguimiento, ejecutar el comando **status** de nuevo:

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#   new file:   README
#
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#
#   modified:   benchmarks.rb
#
```

El archivo **benchmarks.rb** aparece bajo la cabecera “**Changes not staged for commit**”, esto significa que un archivo bajo seguimiento ha sido modificado en el directorio de trabajo, pero no ha sido preparado todavía. Para prepararlo, ejecutar el comando **git add** (es un comando multiuso; se puede utilizar para empezar el seguimiento de archivos nuevos, para preparar archivos, y para otras cosas como marcar como resueltos los archivos con conflictos de unión).

Ejecutar **git add** para preparar el archivo **benchmarks.rb**, y volver a ejecutar **git status**:

```
$ git add benchmarks.rb
$ git status
# On branch master
# Changes to be committed:
```



```
# (use "git reset HEAD <file>..." to unstage)
#
# new file: README
# modified: benchmarks.rb
#
```

Ambos archivos están ahora preparados y se incluirán en la próxima confirmación. De realizar en este momento una pequeña modificación en **benchmarks.rb** antes de confirmarlo luego de ejecutar nuevamente **git status** se tiene:

```
$ vim benchmarks.rb
$ git status
# On branch master
# Changes to be committed:
# (use "git reset HEAD <file>..." to unstage)
#
# new file: README
# modified: benchmarks.rb
#
# Changes not staged for commit:
# (use "git add <file>..." to update what will be committed)
#
# modified: benchmarks.rb
#
```

Ahora **benchmarks.rb** aparece listado como preparado y como no preparado. Git prepara un archivo tal y como era en el momento de ejecutar el comando **git add**. De hacer **git commit** ahora, la versión de **benchmarks.rb** que se incluye la confirmación es la que se incluyó cuando ejecutó el comando **git add**, no la versión que se está viendo en este momento en el directorio de trabajo. De modificar un archivo después de haber ejecutado **git add**, se tiene que volver a ejecutar **git add** para preparar la última versión del archivo y garantizar que la última versión sea sometida:



```
$ git add benchmarks.rb
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#   new file:   README
#   modified:   benchmarks.rb
#
```

Apartado 4.05.3.7

Ignorando archivos

Es posible que se desee ignorar o no tomar en cuenta algunos archivos a la hora de someterlos al manejo de versiones de Git. Para estos casos se puede crear un archivo llamado **.gitignore**, en el que lista los patrones de nombres que se desea que sean ignorados. Como se muestra en el siguiente ejemplo:

```
$ cat .gitignore
*.oa
*~
```

La primera línea le dice a Git que ignore cualquier archivo cuyo nombre termine en `.o` ó `.a`, archivos objeto que suelen ser producto de la compilación de código. La segunda línea le dice a Git que ignore todos los archivos que terminan en tilde (`~`), usada por muchos editores de texto, como Emacs, para marcar archivos temporales. También puede incluir directorios de log, temporales, documentación generada automáticamente, etc. Configurar un archivo **.gitignore** antes de empezar a trabajar suele ser una buena práctica, para así no confirmar archivos que no se desean en el repositorio Git.

Las reglas para los patrones que pueden ser incluidos en el archivo .gitignore son:



- Las líneas en blanco, o que comienzan por #, son ignoradas.
- Puede usar patrones glob estándar.
- Puede indicar un directorio añadiendo una barra hacia delante (/) al final.
- Puede negar un patrón añadiendo una exclamación (!) al principio.

Los patrones glob son expresiones regulares simplificadas que pueden ser usadas por las shells. Un asterisco (*) reconoce cero o más caracteres; [abc] reconoce cualquier carácter de los especificados entre corchetes (en este caso, a, b o c); una interrogación (?) reconoce un único carácter; y caracteres entre corchetes separados por un guión ([0-9]) reconoce cualquier carácter entre ellos (en este caso, de 0 a 9).

Otro ejemplo de archivo **.gitignore** sería:

```
# a comment - this is ignored
# no .a files
*.a
# but do track lib.a, even though you're ignoring .a files above
lib.a
# only ignore the root TODO file, not subdir/TODO
/TODO
# ignore all files in the build/ directory
build/
# ignore doc/notes.txt, but not doc/server/arch.txt
doc/*.txt
# ignore all .txt files in the doc/ directory
doc/**/* .txt
```



Apartado 4.05.3.8

Viendo los cambios preparados y no preparados

Si el comando `git status` es demasiado impreciso, y se quiere saber exactamente lo que se ha cambiado, no sólo qué archivos fueron modificados, se puede usar el comando `git diff`. Aunque `git status` responde esas preguntas de manera general, `git diff` muestra exactamente las líneas añadidas y eliminadas.

En caso de que se desea editar y preparar el archivo **README** otra vez, y luego editar el archivo **benchmarks.rb** sin prepararlo, se ejecuta el comando `status`:

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#   new file:   README
#
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#
#   modified:   benchmarks.rb
#
```

Para ver lo que ha modificado pero aún no se ha preparado, escribir **git diff**:

```
$ git diff
diff --git a/benchmarks.rb b/benchmarks.rb
index 3cb747f..da65585 100644
--- a/benchmarks.rb
```



```
+++ b/benchmarks.rb
@@ -36,6 +36,10 @@ def main
    @commit.parents[0].parents[0].parents[0]
  end
+   run_code(x, 'commits 1') do
+     git.commits.size
+   end
+
  run_code(x, 'commits 2') do
    log = git.commits('master', 15)
    log.size
```

A partir de la versión 1.6.1 de Git, también se puede usar `git diff --staged`, que puede resultar más sencillo de recordar.

Este comando compara el contenido del directorio de trabajo con contenido del área de preparación. El resultado indica los cambios que se han hecho y que todavía no han preparado.

Para ver los cambios que se han preparado y que irán en su próxima confirmación, se puede usar **`git diff --cached`**.

Este comando compara sus cambios preparados con su última confirmación:

```
$ git diff --cached
diff --git a/README b/README
new file mode 100644
index 0000000..03902a1
--- /dev/null
+++ b/README2
@@ -0,0 +1,5 @@
```



```
+grit
+ by Tom Preston-Werner, Chris Wanstrath
+ https://github.com/libgit2/rugged
+
+Rugged is a library for accessing libgit2 in Ruby.
```

Es importante indicar que **git diff** por sí solo no muestra todos los cambios hechos desde la última confirmación, sólo los cambios que todavía no están preparados.

Otro ejemplo es, preparar el archivo **benchmarks.rb** y después editar, se puede usar **git diff** para ver las modificaciones del archivo que están preparadas, y las que no lo están:

```
$ git add benchmarks.rb
$ echo '# test line' >> benchmarks.rb
$ git status
# On branch master
#
# Changes to be committed:
#
#   modified:   benchmarks.rb
#
# Changes not staged for commit:
#
#   modified:   benchmarks.rb
#
```

Se puede usar **git diff** para ver qué es lo que aún no está preparado:

```
$ git diff
diff --git a/benchmarks.rb b/benchmarks.rb
index e445e28..86b2f7c 100644
```



```
~ a/benchmarks.rb
+++ b/benchmarks.rb
@@ -127,3 +127,4 @@ end
main()
##pp Grit::GitRuby.cache_client.stats
+# test line
Y git diff --cached para ver los cambios que lleva preparados hasta
ahora:
$ git diff --cached
diff --git a/benchmarks.rb b/benchmarks.rb
index 3cb747f..e445e28 100644
~ a/benchmarks.rb
+++ b/benchmarks.rb
@@ -36,6 +36,10 @@ def main
  @commit.parents[0].parents[0].parents[0]
end
+  run_code(x, 'commits 1') do
+    git.commits.size
+  end
+
run_code(x, 'commits 2') do
  log = git.commits('master', 15)
  log.size
```

Apartado 4.05.3.9

Confirmando los cambios

Cuando el área de preparación este lista, se puede confirmar los cambios. Es importante recordar que cualquier contenido que todavía esté sin preparar, cualquier archivo que se haya creado o modificado, y sobre el que no se ejecute git add desde su última edición, no se incluye en esta confirmación.



En este caso, la última vez cuando se ejecutó `git status` se pudo visualizar que estaba todo preparado, por lo que ahora está habitable para confirmar los cambios. La forma más fácil de confirmar es escribiendo `git commit`:

```
$ git commit
```

Al hacerlo, se ejecutará el editor de texto. (Esto se configura a través de la variable de entorno `$EDITOR` del shell normalmente `vim` o `emacs`, aunque se puede configurar usando el comando `git config --global core.editor`).

El editor mostrará el siguiente texto (este ejemplo usa Vim):

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#   new file:   README
#   modified:  benchmarks.rb
~
~
~
".git/COMMIT_EDITMSG" 10L, 283C
```

Como se puede observar en el mensaje de confirmación predeterminado, contiene la salida del comando `git status` comentada. Se puede eliminar los comentarios y escribir el mensaje de confirmación, o simplemente se puede dejar para recordar las modificaciones que ha confirmado. (Para un recordatorio todavía más explícito de lo que se ha modificado, se puede pasar la opción `-v` a `git commit`. Esto provoca que se añadan también las diferencias de los



cambios que se han realizado, para que se observe exactamente lo que hizo).

Al salir del editor, Git se crea la confirmación con el mensaje especificado (omitiendo los comentarios y las diferencias).

Como alternativa, se puede escribir un mensaje de confirmación desde la propia línea de comandos mediante la opción -m:

```
$ git commit -m "Story 182: Fix benchmarks for speed"
[master]: created 463dc4f: "Fix benchmarks for speed"
2 files changed, 3 insertions(+), 0 deletions(-)
create mode 100644 README
```

Se puede ver que el comando **commit** ha dado cierta información sobre la confirmación: a qué rama se ha confirmado (master), cuál es la suma de comprobación SHA-1 de la confirmación (463dc4f), cuántos archivos se modificaron, y estadísticas acerca de cuántas líneas se han añadido y cuántas se han eliminado.

Apartado 4.05.3.10

Saltar el área de preparación

Aunque puede ser extremadamente útil para elaborar confirmaciones exactamente con los requerimientos, el área de preparación es en ocasiones compleja para las necesidades del flujo de trabajo. De no querer el área de preparación, Git proporciona una manera más eficiente. Pasando la opción -a al comando git commit hace que Git prepare todo archivo que estuviese en seguimiento antes de la confirmación, permitiendo obviar toda la parte de **git add**:

```
$ git status
# On branch master
#
```



```
# Changes not staged for commit:
#
#   modified:   benchmarks.rb
#
$ git commit -a -m 'added new benchmarks'
master 83e38c7] added new benchmarks
1 files changed, 5 insertions(+), 0 deletions(-)
```

No se ha tenido que ejecutar `git add` sobre el archivo `benchmarks.rb` antes de hacer la confirmación.

Apartado 4.05.3.11

Eliminando archivos

Para eliminar un archivo de Git, se eliminan de los archivos bajo seguimiento (más concretamente, eliminar del área de preparación), y después confirmar la entrada del cambio. El comando `git rm` se encarga de eso, y también eliminar el archivo del directorio de trabajo, para que no sea visto entre los archivos sin seguimiento.

Si simplemente se elimina el archivo del directorio de trabajo, aparecerá bajo la cabecera “**Changes not staged for commit**” (es decir, sin preparar) de la salida del comando `git status`:

```
$ rm grit.gemspec
$ git status
# On branch master
#
# Changes not staged for commit:
#   (use “git add/rm <file>...” to update what will be committed)
#
#       deleted:   grit.gemspec
#
```



Si entonces se ejecuta el comando **git rm**, se prepara la eliminación del archivo en cuestión:

```
$ git rm grit.gemspec
rm 'grit.gemspec'
$ git status
# On branch master
#
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#   deleted:   grit.gemspec
#
```

La próxima vez que se confirme, el archivo desaparecerá y dejará de estar bajo seguimiento. Si ya había modificado el archivo y se tenía en el área de preparación, se deberá forzar su eliminación con la opción **-f**. Esta es una medida de seguridad para evitar la eliminación accidental de información que no ha sido registrada en una entrada (commit), y que por tanto no podría ser recuperada.

Otra decisión que se puede tomar es que desee mantener el archivo en su directorio de trabajo, pero eliminarlo de su área de preparación. Dicho de otro modo, se puede que desee mantener el archivo en el disco duro, pero interrumpir el seguimiento por parte de Git. Esto resulta particularmente útil cuando se olvida añadir algún cambio a el archivo **.gitignore** y es añadido accidentalmente, como un archivo de log de gran tamaño, o un gran número de archivos. Para realizar esto, usar la opción **--cached**:

```
$ git rm --cached readme.txt
```



El comando **git rm** acepta archivos, directorios, y patrones glob. Es decir, que se puede hacer algo como:

```
$ git rm log/\*.log
```

Se puede observar en la barra hacia atrás (\) antes del *. Es necesaria debido a que Git hace su propia expansión de rutas, además de la expansión que hace el shell. En la consola del sistema de Windows, esta barra debe de ser omitida. Este comando elimina todos los archivos con la extensión .log en el directorio log/. También se puede hacer algo como:

```
$ git rm \*~
```

Este comando elimina todos los archivos que terminan en ~.

Apartado 4.05.3.12

Moviendo archivos

A diferencia de muchos otros VCSs, Git no hace un seguimiento explícito del movimiento de archivos. Si renombra un archivo, en Git no se almacena ningún metadato que indique que lo ha renombrado.

Por tanto, es un poco desconcertante que Git tenga un comando mv. Cuando se desea renombrar un archivo en Git, se puede ejecutar:

```
$ git mv file_from file_to
```

De esta manera funciona perfectamente. Muestra que Git lo considera un archivo renombrado:

```
$ git mv README.txt README
$ git status
# On branch master
```



```
# Your branch is ahead of 'origin/master' by 1 commit.
#
# Changes to be committed:
# (use "git reset HEAD <file>..." to unstage)
#
#   renamed:  README.txt -> README
#
$ mv README.txt README
$ git rm README.txt
$ git add README
```

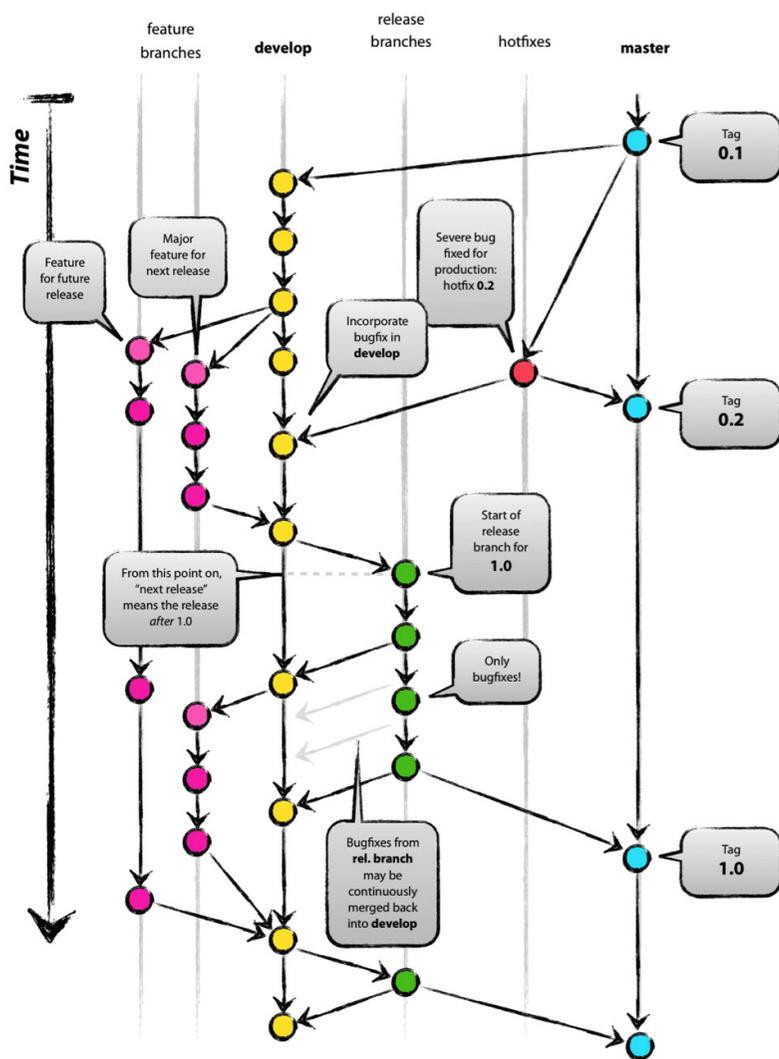
Git se da cuenta de que es un renombrado de manera implícita, así que no importa si renombra un archivo de este modo, o usando el comando `mv`. La única diferencia real es que `mv` es un comando en vez de tres, es más cómodo. Y lo que es más importante, puede usar cualquier herramienta para renombrar un archivo, y preocuparse de los `add` y `rm` más tarde, antes de confirmar.

Sub-sección 4.05.4 Metodología avanzada de manejo de calidad de código con Git

Es una colección de extensiones para Git que proveen una serie de operaciones de alto nivel para utilizar un repositorio Git de manera más fácil.

De manera gráfica el flujo que propone git-flow es el que se muestra en la siguiente **Figura No. 77 Imagen del Flujo Git.**

Figura No. 77 Imagen del Flujo Git.





La descripción de esta gráfica es la siguiente:

- Se tienen por defecto tres branches: master, release y develop. Siendo develop la rama o branch donde ocurre todo y master la branch de producción.
- A partir de develop se crean nuevas ramas con el prefijo feature para la creación de nueva funcionalidad y arreglo de errores que no tienen urgencia.
- En cualquier momento se debe poder pasar de develop a master, lo que quiere decir que no se debe desarrollar en develop si no en las branches ⁽⁸¹⁾que empiezan con feature.
- Para pasar de develop a master se debe usar una branch intermedia llamada release. Esta branch se usa como un punto intermedio en la cual solo se pueden corregir errores que puedan surgir antes de pasar a master.
- Si al pasar a master se descubre un error, se usan branches con prefijo hotfix para corregir errores de producción.
- Cada vez que se pasa a master, se debe crear una etiqueta con el nombre del release.

(81) Es un apuntador móvil dirigido a una de las confirmaciones o “commit”.

Apartado 4.05.4.1

Inicio de Git-Flow

Todo inicia con **git flow init**, como se muestra en la siguiente **Figura No. 78 Imagen del Flujo Git ini:**

Figura No. 78 Imagen del Flujo Git ini.

```
1. Mario@Mario-Mac: ~/mi-super-programa (zsh)
Mario@Mario-Mac:~ » mkdir mi-super-programa && cd $_
Mario@Mario-Mac:~/mi-super-programa » git init
Initialized empty Git repository in /Users/Mario/mi-super-programa/.git/
Mario@Mario-Mac:~/mi-super-programa(master) »
Mario@Mario-Mac:~/mi-super-programa(master) » git flow init
No branches exist yet. Base branches must be created now.
Branch name for production releases: [master]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Mario@Mario-Mac:~/mi-super-programa(develop) » git branch -a
* develop
  master
Mario@Mario-Mac:~/mi-super-programa(develop) »
```

Como se puede ver, git-flow crea todas las branches necesarias. Aquí se pueden subir las nuevas ramas al servidor remoto, como se muestra en la siguiente **Figura No. 79 branches:**

Figura No. 79 branches.

```
1. Mario@Mario-Mac: ~/mi-super-programa (zsh)
Mario@Mario-Mac:~/mi-super-programa(develop) » git remote add origin git@github.com:marioy47/dazzet-blog-gitflow
Mario@Mario-Mac:~/mi-super-programa(develop) » git push -u origin master
Counting objects: 2, done.
Writing objects: 100% (2/2), 179 bytes | 0 bytes/s, done.
Total 2 (delta 0), reused 0 (delta 0)
To git@github.com:marioy47/dazzet-blog-gitflow
 * [new branch] master -> master
Branch master set up to track remote branch master from origin.
Mario@Mario-Mac:~/mi-super-programa(develop) » git branch -a
* develop
  master
  remotes/origin/master
Mario@Mario-Mac:~/mi-super-programa(develop) »
Mario@Mario-Mac:~/mi-super-programa(develop) » git push -u origin develop
Total 0 (delta 0), reused 0 (delta 0)
To git@github.com:marioy47/dazzet-blog-gitflow
 * [new branch] develop -> develop
Branch develop set up to track remote branch develop from origin.
Mario@Mario-Mac:~/mi-super-programa(develop) » git branch -a
* develop
  master
  remotes/origin/develop
  remotes/origin/master
Mario@Mario-Mac:~/mi-super-programa(develop) »
```

Apartado 4.05.4.2

Creación de una nueva funcionalidad o corrección de un error

Para git-flow esto es un nuevo complemento, es decir, se puede ejecutar git flow feature star, como se muestra en la siguiente **Figura No. 80 Imagen del git flow feature star:**

Figura No. 80 Imagen del git flow feature star.

```
1. Mario@Mario-Mac: ~/mi-super-programa (zsh)
Mario@Mario-Mac:~/mi-super-programa(develop) » git branch -a
* develop
  master
  remotes/origin/develop
  remotes/origin/master
Mario@Mario-Mac:~/mi-super-programa(develop) » git flow feature start 1234-7890-un-cambio-pequeno
Switched to a new branch 'feature/1234-7890-un-cambio-pequeno'

Summary of actions:
- A new branch 'feature/1234-7890-un-cambio-pequeno' was created, based on 'develop'
- You are now on branch 'feature/1234-7890-un-cambio-pequeno'

Now, start committing on your feature. When done, use:

    git flow feature finish 1234-7890-un-cambio-pequeno

Mario@Mario-Mac:~/mi-super-programa(feature/1234-7890-un-cambio-pequeno) »
```

Como se puede ver git-flow creó una nueva rama llamada feature/1234-7890-un-cambio-pequeño e hizo un checkout a esta rama automáticamente.

Apartado 4.05.4.3

Nombres de las ramas

Se recomienda que al trabajar con git-flow los branches sigan una convención de nombres que esté fuertemente ligada con el bug traker, ya sea Jira, The Bug Genie, entre otros. IdDelBug-Numero DeComentario-DescripciónDeLaFuncionalidad.

En el ejemplo anterior se tiene:

- Id del bug: 1234

- **Número del comentario:** 7890. Se asume que en un comentario del bug es en el que se define que se puede hacer en el código.
- **Descripción de la funcionalidad:** Idealmente se separa por guiones. Algo como nuevo-comando-de-reset.

Una vez creada la rama con el nuevo complemento, se puede trabajar como se hace normalmente con git. Como se muestra en la siguiente **Figura No. 81 Trabajando con Git:**

Figura No. 81 Trabajando con Git.

```
1. Mario@Mario-Mac: ~/mi-super-programa (zsh)
Mario@Mario-Mac:~/mi-super-programa(feature/1234-7890-un-cambio-pequeno) » echo "# Un programa de ejemplo" >> README.md
Mario@Mario-Mac:~/mi-super-programa(feature/1234-7890-un-cambio-pequeno) » echo "Este programa no tiene nada. Es solo un ejemplo" >> README.md
Mario@Mario-Mac:~/mi-super-programa(feature/1234-7890-un-cambio-pequeno) » git status
On branch feature/1234-7890-un-cambio-pequeno
Untracked files:
  (use "git add <file>..." to include in what will be committed)

  README.md

nothing added to commit but untracked files present (use "git add" to track)
Mario@Mario-Mac:~/mi-super-programa(feature/1234-7890-un-cambio-pequeno) » git add README.md
Mario@Mario-Mac:~/mi-super-programa(feature/1234-7890-un-cambio-pequeno) »
Mario@Mario-Mac:~/mi-super-programa(feature/1234-7890-un-cambio-pequeno) » git commit -m "Agregando readme"
[feature/1234-7890-un-cambio-pequeno 40b163f] Agregando readme
1 file changed, 10 insertions(+)
create mode 100644 README.md
Mario@Mario-Mac:~/mi-super-programa(feature/1234-7890-un-cambio-pequeno) »
```

Apartado 4.05.4.4

Publicando la nueva rama

Git ayuda a trabajo colaborativo, así que es necesario publicar esta nueva rama al servidor. Esto se logra con `git flow feature publish <feature>` como se muestra en la **Figura No. 82 git flow feature publish <feature>**.

Figura No. 82 git flow feature publish <feature>.

```
1. Mario@Mario-Mac: ~/mi-super-programa (zsh)
Mario@Mario-Mac:~/mi-super-programa(feature/1234-7890-un-cambio-pequeno) » git flow featu
re publish 1234-7890-un-cambio-pequeno
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 324 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@github.com:marioy47/dazzet-blog-gitflow
 * [new branch]      feature/1234-7890-un-cambio-pequeno -> feature/1234-7890-un-cambio-pe
queno
Already on 'feature/1234-7890-un-cambio-pequeno'
Your branch is up-to-date with 'origin/feature/1234-7890-un-cambio-pequeno'.

Summary of actions:
- A new remote branch 'feature/1234-7890-un-cambio-pequeno' was created
- The local branch 'feature/1234-7890-un-cambio-pequeno' was configured to track the remot
e branch
- You are now on branch 'feature/1234-7890-un-cambio-pequeno'

Mario@Mario-Mac:~/mi-super-programa(feature/1234-7890-un-cambio-pequeno) »
```

Este comando toma su tiempo, ya que envía también los cambios al servidor remoto.

Apartado 4.05.4.5

Otros colaboradores

Estando el código en el servidor por medio a git flow feature publish <feature>, el resto de desarrolladores ya pueden trabajar en la nueva funcionalidad. Para esto lo que debe hacerse es usar git normalmente para descargar y sincronizar el código:

```
git fetch
git checkout feature/1234-7890-un-pequeno-cambio
... (cambios a los archivos)
git fetch #(se trae cambio remotos)
git rebase origin feature/1234-7890-un-pequeno-cambio
... (mas cambios)
git push
```

Apartado 4.05.4.6

Finalización de la funcionalidad

Cuando se termine de crear la nueva funcionalidad, Se debe ejecutar `git flow feature finish <feature>` y así pasar nuevamente a la rama `develop`, como se muestra en la siguiente **Figura No. 83 nueva funcionalidad:**

Figura No. 83 nueva funcionalidad.

```
1. Mario@Mario-Mac: ~/mi-super-programa (zsh)
Mario@Mario-Mac:~/mi-super-programa(feature/1234-7890-un-cambio-pequeno) » git flow featu
re finish 1234-7890-un-cambio-pequeno
Switched to branch 'develop'
Your branch is up-to-date with 'origin/develop'.
Updating c1c0cae..40b163f
Fast-forward
 README.md | 10 ++++++++
 1 file changed, 10 insertions(+)
 create mode 100644 README.md
 Deleted branch feature/1234-7890-un-cambio-pequeno (was 40b163f).

Summary of actions:
- The Feature branch 'feature/1234-7890-un-cambio-pequeno' was merged into 'develop'
- Feature branch 'feature/1234-7890-un-cambio-pequeno' has been removed
- You are now on branch 'develop'

Mario@Mario-Mac:~/mi-super-programa(develop) » git push origin develop
Total 0 (delta 0), reused 0 (delta 0)
To git@github.com:marioloy47/dazzet-blog-gitflow
 c1c0cae..40b163f develop -> develop
Mario@Mario-Mac:~/mi-super-programa(develop) »
```

Lo que hace este comando es:

- Hacer merge con `develop`.
- Borrar la rama `feature` en la que se estaba trabajando.
- No regresa a la rama `develop`.

En este instante, es permitido hacer “push” al servidor.

Apartado 4.05.4.7

Limpieza

Con la funcionalidad finalizada, esta rama ya no es necesaria, entonces lo mejor es borrar la rama de la funcionalidad del servidor remoto y así no tener miles de ramas, una por funcionalidad.

El comando a ejecutar aquí es directamente de git: git push origin :feature/<feature>, como se muestra en la siguiente **Figura No. 84 Limpieza del servidor:**

Figura No. 84 Limpieza del servidor

```
1. Mario@Mario-Mac: ~/mi-super-programa (zsh)
Mario@Mario-Mac:~/mi-super-programa(develop) » git branch -a
* develop
  master
  remotes/origin/develop
  remotes/origin/feature/1234-7890-un-cambio-pequeno
  remotes/origin/master
Mario@Mario-Mac:~/mi-super-programa(develop) » git push origin :feature/1234-7890-un-cambio-pequeno
To git@github.com:marioy47/dazzet-blog-gitflow
- [deleted]      feature/1234-7890-un-cambio-pequeno
Mario@Mario-Mac:~/mi-super-programa(develop) » git branch -a
* develop
  master
  remotes/origin/develop
  remotes/origin/master
Mario@Mario-Mac:~/mi-super-programa(develop) »
```

Apartado 4.05.4.8

Distribuciones

Una vez terminados todas las funcionalidades que se querían corregir o implementar, se procede a crear una distribución. En términos de git, esto significa hacer un merge a master. Para esto está el comando: git flow release start <release>, como se muestra en la siguiente **Figura No. 85 Creación de la distribución:**

Figura No. 85 Creación de la distribución

```
1. Mario@Mario-Mac: ~/mi-super-programa (zsh)
Mario@Mario-Mac:~/mi-super-programa(develop) » git branch -a
* develop
  master
  remotes/origin/develop
  remotes/origin/master
Mario@Mario-Mac:~/mi-super-programa(develop) » git flow release start 2014-09-20.0
Switched to a new branch 'release/2014-09-20.0'

Summary of actions:
- A new branch 'release/2014-09-20.0' was created, based on 'develop'
- You are now on branch 'release/2014-09-20.0'

Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

    git flow release finish '2014-09-20.0'

Mario@Mario-Mac:~/mi-super-programa(release/2014-09-20.0) » git branch -a
  develop
  master
* release/2014-09-20.0
  remotes/origin/develop
  remotes/origin/master
Mario@Mario-Mac:~/mi-super-programa(release/2014-09-20.0) »
```

El resultado de este comando es la creación de una nueva rama con el nombre `release/2014-09-20.0` que se puede enviar al servidor para que la puedan descargar los colaboradores, como se muestra en la siguiente **Figura No. 86** `release/2014-09-20.0`:

Figura No. 86 `release/2014-09-20.0`.

```
1. Mario@Mario-Mac: ~/mi-super-programa (zsh)
Mario@Mario-Mac:~/mi-super-programa(release/2014-09-20.0) » git push origin release/2014-09-20.0
Total 0 (delta 0), reused 0 (delta 0)
To git@github.com:marioy47/dazzet-blog-gitflow
* [new branch]   release/2014-09-20.0 -> release/2014-09-20.0
Mario@Mario-Mac:~/mi-super-programa(release/2014-09-20.0) »
Mario@Mario-Mac:~/mi-super-programa(release/2014-09-20.0) »
```

En este punto se puede hacer aseguramiento de la calidad (QA, por sus siglas en inglés), es decir, todo lo que tenga que ver con pruebas y aseguramiento de calidad.

Una vez terminado el QA, se finaliza la distribución con `git flow release finish <release>` que pasará todos los cambios a `master`, como se muestra en la siguiente **Figura No. 87** `QA`:

Figura No. 87 `QA`.

```
1. Mario@Mario-Mac: ~/mi-super-programa (zsh)
Mario@Mario-Mac:~/mi-super-programa(release/2014-09-20.0) » git flow release finish 2014-09-20.0
Switched to branch 'master'
Your branch is up-to-date with 'origin/master'.
Merge made by the 'recursive' strategy.
 README.md | 10 ++++++++
 1 file changed, 10 insertions(+)
 create mode 100644 README.md
fatal: no tag message?
Tagging failed. Please run finish again to retry.
Mario@Mario-Mac:~/mi-super-programa(master) » git push origin :release/2014-09-20.0 1 ↵
To git@github.com:marioy47/dazzet-blog-gitflow
 - [deleted]   release/2014-09-20.0
Mario@Mario-Mac:~/mi-super-programa(master) » git push --tags
Everything up-to-date
Mario@Mario-Mac:~/mi-super-programa(master) »
```

Por último se hace limpieza y se envían los tags al servidor.



Apartado 4.05.4.9

Nombres de las distribuciones

Al igual que las funcionalidades, se recomienda usar un formato para nombrar las distribuciones.

El más usado es la fecha en formato AAAA-MM-DD.#, es decir año-mes-día, más el número de distribución por día. Esto en caso de que sea necesario hacer varias distribuciones en el mismo día.

Apartado 4.05.4.10

Corrección de errores

Para hacer es una corrección de errores en producción, usar git flow hotfix start <release> y git flow hotfix finish <release>, como se muestra en la siguiente **Figura No. 88 Corrección de errores y Figura No. 89 Corrección de errores 2:**

Figura No. 88 Corrección de errores

```
1. Mario@Mario-Mac: ~/mi-super-programa (zsh)
Mario@Mario-Mac:~/mi-super-programa(master) » git flow hotfix start 2014-09-20.0
Switched to a new branch 'hotfix/2014-09-20.0'

Summary of actions:
- A new branch 'hotfix/2014-09-20.0' was created, based on 'master'
- You are now on branch 'hotfix/2014-09-20.0'

Follow-up actions:
- Bump the version number now!
- Start committing your hot fixes
- When done, run:

git flow hotfix finish '2014-09-20.0'
Mario@Mario-Mac:~/mi-super-programa(hotfix/2014-09-20.0) »
```

Figura No. 89 Corrección de errores 2

```
1. Mario@Mario-Mac: ~/mi-super-programa (zsh)
Mario@Mario-Mac:~/mi-super-programa(hotfix/2014-09-20.0) » git flow hotfix fini
sh 2014-09-20.0
Switched to branch 'develop'
Your branch is up-to-date with 'origin/develop'.
Already up-to-date!
Merge made by the 'recursive' strategy.
Deleted branch hotfix/2014-09-20.0 (was 817909f).

Summary of actions:
- Latest objects have been fetched from 'origin'
- Hotfix branch has been merged into 'master'
- The hotfix was tagged '2014-09-20.0'
- Hotfix branch has been back-merged into 'develop'
- Hotfix branch 'hotfix/2014-09-20.0' has been deleted

Mario@Mario-Mac:~/mi-super-programa(develop) »
```

Al hacer git flow finish, se hace push tanto a master como a develop. Esto con el fin de no perder en develop los arreglos hechos en master.

Finalmente se envía a master, como se muestra en la siguiente **Figura No. 90 git flow finish:**

Figura No. 90 git flow finish.

```
1. Mario@Mario-Mac: ~/mi-super-programa (zsh)
Mario@Mario-Mac:~/mi-super-programa(develop) » git push origin master
To git@github.com:marioy47/dazzet-blog-gitflow
 ! [rejected]      master -> master (fetch first)
error: failed to push some refs to 'git@github.com:marioy47/dazzet-blog-gitflow'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
Mario@Mario-Mac:~/mi-super-programa(develop) » 1 ↵
Mario@Mario-Mac:~/mi-super-programa(develop) » git push --tags 1 ↵
Counting objects: 2, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 399 bytes | 0 bytes/s, done.
Total 2 (delta 0), reused 0 (delta 0)
To git@github.com:marioy47/dazzet-blog-gitflow
 * [new tag]      2014-09-20.0 -> 2014-09-20.0
Mario@Mario-Mac:~/mi-super-programa(develop) »
```



SECCIÓN 4.06

INTERFAZ DE USUARIO WEB

Actualmente las interfaces de usuario web son fundamentales en el desarrollo de cualquier solución web. Debido a la gran cantidad de dispositivos que existen hoy en día, se vuelve un reto que la solución tecnológica desarrollada sea accesible por todos estos para lograr que el usuario final llegue a utilizarla bajo el criterio de obtener la mejor experiencia posible.

Una interfaz de usuario no solo refiere a un punto completamente estético, en realidad se trata de una disciplina, sobre todo, funcional. Una interfaz de usuario cumple con su objetivo cuando funciona de manera correcta, es capaz de cumplir con lo que promete y ayuda al usuario a encontrar lo que está buscando.

De acuerdo con las estadísticas el 68% de los usuarios que abandonan una web lo hacen porque la experiencia de usuario está mal diseñada, es decir porque no funciona como debería o no tiene los elementos necesarios que el usuario espera.

Sub-sección 4.06.1

Características de una buena interfaz de usuario

Simplicidad

Siempre debe tomarse como prioridad la permanencia de los usuarios en la aplicación o página web que se desarrolle, por lo que es necesario facilitarles la visita haciendo una web sencilla, clara y concisa. Para esto hay que crear una jerarquía que explique a los usuarios lo que pueden hacer. Algunos consejos que pueden seguirse para mejorar una interfaz de usuario son los siguientes:

- Agrupar las funcionalidades similares en vez de fragmentarlas.



- Usar diferentes estilos para los elementos que son cliclables y los que son seleccionables.
- Siempre que sea posible hacer una sola columna en vez de una web multicolumna.

Experiencia

Es una característica de importancia para las interfaces de usuario. Una vez se tenga solucionada la funcionalidad y la simplicidad, es recomendable proveerle a los usuarios funcionalidades por las que realmente terminarían sintiéndose a gusto con dicha interfaz, de manera que estos disfruten su uso.

Una forma sencilla y efectiva de conseguir esta experiencia es mediante la animación de la interfaz de usuario. Hoy en día se tiene a disposición las tecnologías HTML5⁽⁸²⁾ y CSS3 que contienen una cantidad de herramientas para poder realizar animaciones, transiciones, degradaciones dinámicas, entre otros.

Detalles

Es importante enfocarse en pequeños comportamientos o detalles que terminan de complementar una interfaz de usuario rica. Esto se puede lograr siguiendo los pasos a continuación:

- Todos los navegadores de Internet que se utilizan actualmente están basados en “Pestañas⁽⁸³⁾” esto da cabida a aprovechar detalles como crear cambios dinámicos del favicon⁽⁸⁴⁾ para que el usuario identifique lo que está sucediendo en la aplicación, a pesar de encontrarse posicionado en otra “Pestaña”.

(82) Es el lenguaje de programación utilizado para la creación de páginas web. El número 5 que acompaña el término al final hace referencia a su versión 5.

(83) Elemento que pertenece a la interfaz gráfica, el cual le permite al usuario de una aplicación cambiar entre distintos documentos de una forma rápida, normalmente están organizadas en una al lado de otra en fila.

(84) Es una imagen que tiene relación con el portal web en el que se encuentra el usuario.



- Crear transiciones sutiles entre páginas.
- Crear la programación necesaria para que los elementos que se desplieguen se carguen de forma independiente y asincrónica, y de esta forma evitar una percepción de suspensión o lentitud en la página o aplicación mostrándose la pantalla en blanco en cada clic que el usuario realice.

Sub-sección 4.06.2

Marcos de Trabajo para Interfaces de Usuario

Los Marcos de trabajo, comúnmente conocidos como frameworks, facilitan el trabajo en gran medida a la hora de crear interfaces de usuario eficientes y bajo el nuevo concepto de diseño web adaptable.

Para la creación de interfaces de usuario se ha tomado como ejemplo el marco de trabajo Bootstrap⁽⁸⁵⁾, el cual implementa la librería de archivo LESS⁽⁸⁶⁾ especificado en la **NORTIC A4:2014. Sección 5.06 sobre Interoperabilidad web**. Con esto se pretende explicar los conceptos básicos del manejo de los elementos de este marco de trabajo.

Apartado 4.06.2.1

Descargando Bootstrap

Existen varias formas diferentes para empezar a utilizar Bootstrap, cada una de esta se orienta a un tipo de público en función de su nivel técnico. A continuación se especifican algunas formas de las cuales se puede seleccionar la que más se adapte a las circunstancias:

(85) Es un conocido marco de trabajo para el desarrollo de interfaces y sitios web con diseños adaptativos basados en CSS, HTML y Javascript.

(86) Es un lenguaje dinámico de hoja de estilos, el cual extiende el lenguaje CSS, añadiendo características que permiten variables, funciones y muchas otras técnicas con las cuales se pueden realizar CSS más fácil de mantener y extensible.



- Descargar el código CSS y JavaScript compilado: Es la forma más sencilla de empezar a utilizar Bootstrap. La desventaja es que esta versión no incluye ni los archivos originales ni la documentación.
- Esta versión puede ser descargada accediendo al sitio web getbootstrap.com y luego pulsando el botón Download Bootstrap.
- Descargar el código fuente: Esta forma contiene todos los archivos Less, y JavaScript originales de Bootstrap. La desventaja es que requiere un compilador de archivos Less y trabajo de configuración.
- Si se utiliza esta forma, se puede descargar la versión más reciente accediendo en el sitio web github.com/twbs/bootstrap/releases.
- Descargar el código fuente en formato Sass⁽⁸⁷⁾, es una variante de la versión anterior , la cual se ha creado para facilitar la integración de Bootstrap en las aplicaciones Ruby On Rails⁽⁸⁸⁾, Compass⁽⁸⁹⁾ o cualquier otro proyecto basado en Sass.
 - Si se utiliza esta forma, se puede descargar la versión más reciente accediendo en el sitio web github.com/twbs/bootstrap-sass/releases.

(87) Es un metalenguaje de CSS, el cual se utiliza para describir el estilo de un documento de manera limpia y estructurada.

(88) Es un marco de trabajo de código abierto para aplicaciones web cuyo objetivo es simplificar la escritura de aplicaciones del mundo real con menos código que con otros marcos de trabajo.

(89) Es un marco de trabajo de hojas de estilos que hace las hojas de estilo más fáciles de construir y mantener. Con este marco se pueden escribir las hojas de estilo en Sass en vez de CSS plano.



- Vía Red de entrega de Contenidos (CDN⁽⁹⁰⁾, por sus siglas en inglés): Esta forma permite a los usuarios de Bootstrap alojar los archivos CSS y JavaScript en un servidor multi-localización. Para utilizar estos archivos, se deben modificar los siguientes dos enlaces en el HTML de las páginas:

```
<!-- Versión compilada y comprimida del CSS de Bootstrap  
--><link rel="stylesheet" href="//netdna.bootstrapcdn.com/bootstrap/3.3.0/css/bootstrap.min.css">  
<!-- Tema opcional --><link rel="stylesheet" href="//netdna.bootstrapcdn.com/bootstrap/3.1.0/css/bootstrap-theme.min.css">  
<!-- Versión compilada y comprimida del JavaScript de Bootstrap --><script src="//netdna.bootstrapcdn.com/bootstrap/3.3.0/js/bootstrap.min.js"></script>
```

Para utilizar este código debe cambiarse el valor 3.3.0 del versionamiento por la versión específica de Bootstrap que se vaya a utilizar.

Apartado 4.06.2.2

Requerimientos para trabajar con Bootstrap

A la hora de trabajar con Bootstrap, los requerimientos de entorno son los siguientes:

- Un editor de preferencia, puede ser Sublime, Notepad++, entre otros.
- Twitter Bootstrap 2.3.2
- jQuery 2.1.3

(90) Es una red de computadores en la cual se disponen copias de datos alojados en diferentes lugares de la red, con el objetivo de que los usuarios puedan tener un acceso más rápido a dichos datos.



Apartado 4.06.2.3

Instalación de Bootstrap

1. Primero debe descargarse la versión .zip de Twitter Bootstrap de la web oficial. Este presenta el siguiente contenido:

```
bootstrap/  
├── css/  
│   ├── bootstrap.css  
│   ├── bootstrap.min.css  
│   ├── bootstrap-theme.css  
│   └── bootstrap-theme.min.css  
├── js/  
│   ├── bootstrap.js  
│   └── bootstrap.min.js  
└── fonts/  
    ├── glyphicons-halflings-regular.eot  
    ├── glyphicons-halflings-regular.svg  
    ├── glyphicons-halflings-regular.ttf  
    └── glyphicons-halflings-regular.woff
```

Son tres carpetas: los CSS, los archivos JS y las imágenes en sus versiones de desarrollo y producción.

2. Luego se crea una página básica que incluya lo siguiente:

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Mi primera web con Bootstrap</title>  
    <meta name="viewport" content="width=device-width, initial-  
scale=1.0"/>  
    <link href="css/bootstrap.min.css" rel="stylesheet" media="screen"/>  
    <link href="css/bootstrap-responsive.min.css" rel="stylesheet"  
media="screen"/>
```



```
</head>
<body>
  <h1>Mi primera web con Bootstrap</h1>
  <script src="js/jquery-2.1.3.min.js"></script>
  <script src="js/bootstrap.min.js"></script>
</body>
</html>
```

Apartado 4.06.2.4 Sistema de cuadrícula para la web

El sistema de cuadrículas⁽⁹¹⁾, también conocido en inglés como Grid, debe definirse correctamente, ya que una vez se tengan definidas las zonas de la web a utilizar, será más sencillo añadir el contenido.

Hay que tener muy en cuenta que un gran número de usuarios se conectan a Internet desde dispositivos móviles, los cuales tienen diferentes resoluciones y tamaños de pantalla, por lo que, siempre se debe tener en cuenta que los sitios web deben verse bien en múltiples dispositivos.

(91) Es un sistema referencia el cual se utiliza para realizar localizaciones mediante un plano constituido por cuadrículas.



Apartado 4.06.2.5

Componentes básicos

Twitter Bootstrap cuenta con un catálogo de componentes muy amplio con todo lo necesario para crear interfaces web. Alguno de los componentes más comunes son los especificados a continuación:

- Tablas:

```
<table class="table table-striped">
  <thead>
    <tr>
      <th>#</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Username</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Mark</td>
      <td>Otto</td>
      <td>@mdo</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Jacob</td>
      <td>Thornton</td>
      <td>@fat</td>
    </tr>
    <tr>
      <td>3</td>
      <td>Larry</td>
```



```
<td>the Bird</td>
<td>@twitter</td>
</tr>
</tbody>
</table>
```

- Formularios:

```
<form class="form-horizontal">
  <div class="control-group">
    <label class="control-label" for="inputUser">Usuario:</label>
    <div class="controls">
      <input type="text" id="inputUser" placeholder="usuario">
    </div>
  </div>
  <div class="control-group">
    <label class="control-label" for="inputPassword">Contraseña</
label>
    <div class="controls">
      <input type="password" id="inputPassword"
placeholder="contraseña">
    </div>
  </div>
  <div class="control-group">
    <div class="controls">
      <label class="checkbox">
        <input type="checkbox"> Recordarme
      </label>
      <button type="submit" class="btn">Entrar</button>
    </div>
  </div>
</form>
```



- Pestañas:

```
<div class="tabbable">
  <ul class="nav nav-tabs">
    <li class="active"><a href="#tab1" data-toggle="tab">Pestaña 1</a></li>
    <li><a href="#tab2" data-toggle="tab">Pestaña 2</a></li>
    <li><a href="#tab3" data-toggle="tab">Pestaña 3</a></li>
  </ul>
  <div class="tab-content">
    <div class="tab-pane active" id="tab1">
      <p>Sección 1</p>
    </div>
    <div class="tab-pane" id="tab2">
      <p>Sección 2</p>
    </div>
    <div class="tab-pane" id="tab3">
      <p>Sección 3</p>
    </div>
  </div>
</div>
```

- Barra de navegación:

```
<div class="navbar">
  <div class="navbar-inner">
    <div class="nav-collapse collapse">
      <ul class="nav">
        <li><a href="#">Quiénes somos</a></li>
        <li><a href="#">Formación</a></li>
        <li class="dropdown">
          <a href="#" class="dropdown-toggle" data-toggle="dropdown">Tutoriales <b class="caret"></b></a>
          <ul class="dropdown-menu">
```



```
<li><a href="#">Spring</a></li>
<li><a href="#">BPM</a></li>
<li><a href="#">Maven</a></li>
</ul>
</li>
</ul>
</div>
</div>
</div>
```

- Alertas:
 - Éxito

```
<div class="alert alert-success">
  <h4>Success</h4>
</div>
```

- Advertencia

```
<div class="alert alert-block">
  <h4>Warning</h4>
</div>
```

- Error

```
<div class="alert alert-error">
  <h4>Error</h4>
</div>
```

- Información

```
<div class="alert alert-info">
  <h4>Info</h4>
</div>
```



- Barra de progreso:

```
<div class="progress progress-striped active">  
  <div class="bar" style="width: 70%;"></div>  
</div>
```

Apartado 4.06.2.6

Componentes avanzados

Algunos componentes más avanzados que también se utilizan con frecuencia en las aplicaciones web son los indicados a continuación:

- Ventana modal:

```
<a href="#myModal" role="button" class="btn" data-  
toggle="modal">Abrir ventana modal</a>  
  <div id="myModal" class="modal hide fade" tabindex="-1"  
role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">  
  <div class="modal-header">  
    <button type="button" class="close" data-dismiss="modal" aria-  
hidden="true">x</button>  
    <h3 id="myModalLabel">Alerta</h3>  
  </div>  
  <div class="modal-body">  
    <p>Cuerpo de la ventana modal</p>  
  </div>  
  <div class="modal-footer">  
    <button class="btn" data-dismiss="modal" aria-  
hidden="true">Cerrar</button>  
  </div>  
</div>
```



- Descripción emergente

```
<a id="link-tooltip" data-toggle="tooltip" data-placement="bottom" data-original-title="Mensaje adicional">Texto con tooltip</a>
<script language="javascript" type="text/javascript">
  $(document).ready(function () {
    var opciones = {
      placement: "right", // Posicionado a la derecha
      delay: 200 // Tiempo de espera 200 milésimas de segundo
    };

    // Se seleccionan los elementos con clase btn
    $("#link-tooltip").tooltip(opciones);
  });
</script>
```

- Acordeón

```
<div class="accordion" id="accordion2">
  <div class="accordion-group">
    <div class="accordion-heading">
      <a class="accordion-toggle" data-toggle="collapse" data-parent="#accordion2" href="#collapseOne">
        Panel 1
      </a>
    </div>
    <div id="collapseOne" class="accordion-body collapse in">
      <div class="accordion-inner">
        Contenido del panel 1
      </div>
    </div>
  </div>
</div>
```

```
<div class="accordion-group">
  <div class="accordion-heading">
    <a class="accordion-toggle" data-toggle="collapse" data-
parent="#accordion2" href="#collapseTwo">
      Panel 2
    </a>
  </div>
  <div id="collapseTwo" class="accordion-body collapse">
    <div class="accordion-inner">
      Contenido del panel 2
    </div>
  </div>
</div>
</div>
```

- Carrusel

```
<div id="myCarousel" class="carousel slide"
style="overflow:auto;width:300px;">
  <ol class="carousel-indicators">
    <li data-target="#myCarousel" data-slide-to="0" class="active"></
li>
    <li data-target="#myCarousel" data-slide-to="1"></li>
    <li data-target="#myCarousel" data-slide-to="2"></li>
  </ol>

  <div class="carousel-inner" style="margin-left:4rem;">
    <div class="active item">
      
    </div>

    <div class="item">
```



```

</div>

<div class="item">
    
</div>

<div class="item">
    
</div>

</div>

<a class="carousel-control left" href="#myCarousel" data-slide="prev"></a>
<a class="carousel-control right" href="#myCarousel" data-slide="next"></a>
</div>
```



GLOSARIO DE TÉRMINOS

Arreglo:

Es una estructura de datos, el cual sirve para almacenar y manejar un número “n” de elementos, todos del mismo tipo.

Bootstrap:

Es un conocido marco de trabajo para el desarrollo de interfaces y sitios web con diseños adaptativos basados en CSS, HTML y Javascript.

Branches / ramas:

Es un apuntador móvil dirigido a una de las confirmaciones o “commit”.

Caché:

Es un tipo de memoria de acceso rápido, la cual almacena la última información de manera temporal.

Cadena:

Es una secuencia ordenada, compuesta de caracteres ligados entre sí, las cuales pueden estar asignadas a una variable.

Capa de Conexión Segura (SSL):

Es un protocolo de seguridad para conexiones de transmisión de información, el cual emplea autenticación y cifrado de datos.

Certificado digital:

Es un documento digital que permite garantizar la identidad de una persona en la red, a través de una firma electrónica. Se utiliza como una forma segura de garantizar la autenticación, integridad y confidencialidad de la información.



Código:

Es un conjunto de instrucciones redactas en base a las reglas sintácticas de un lenguaje de programación para desarrollar un software determinado.

Comandos:

Son cada una de las sentencias u órdenes que a través de una interfaz el usuario le pasa al sistema con el fin de realizar una tarea o función específica.

Compass:

Es un marco de trabajo de hojas de estilos que hace las hojas de estilo más fáciles de construir y mantener. Con este marco se pueden escribir las hojas de estilo en Sass en vez de CSS plano.

Compilar:

Consiste en traducir mediante un software o herramienta las instrucciones de un lenguaje de programación aun lenguaje entendible por la máquina.

Consola:

Es el medio por el cual el usuario interactúa con el sistema mediante la ejecución de comandos.

Consorcio World Wide Web (W3C):

Es una entidad internacional que produce pautas y recomendaciones para la usabilidad y accesibilidad de la web.

Correo electrónico:

Es un servicio de mensajería en red que permite el intercambio de mensajes, a través de sistemas de comunicación electrónicos.



Cortafuegos:

Es un equipo diseñado para bloquear los accesos no permitidos a una red.

Datos abiertos:

Son datos que están disponibles para cualquier persona sin restricciones, los cuales pueden ser utilizados, reutilizados y redistribuidos libremente.

Datos:

Hace referencia a un valor íntegro sobre un elemento determinado, el cual por si solo carece de importancia y a través del procesamiento adecuado logra convertirse en información útil.

Debian:

Debian es un sistema operativo (S.O) libre, para su computadoras.

Desarrollo Guiado por Comportamiento (BDD):

Es un proceso que amplía las ideas de TDD y las une con otras ideas de diseño de software para proporcionar un proceso a los desabolladores para que puedan mejorar el desarrollo del software.

Desarrollo Guiado por Pruebas de Software (TDD):

Es una técnica de programación que consiste en desarrollar el código que pruebe una característica o funcionalidad deseada antes del código que implementa dicha funcionalidad.

Diseño web adaptable (Responsive):

Es una técnica de diseño y desarrollo que permite la adaptación de una página web a diferentes dispositivos para su correcta visualización.



Distribución:

Son cada una de las diferentes versiones funcionales que se desarrollan de un sistema.

Fávicón:

Es una imagen que tiene relación con el portal web en el que se encuentra el usuario.

Fedora:

Es una distribución del sistema operativo linux, desarrollado en base a código libre.

Formas normales:

Las formas normales son teorías que proporcionan criterios para determinar qué tan vulnerable puede ser una tabla de una base de datos a anomalías e inconsistencias.

Formato de Transformación Unicode de 8-Bit (UTF-8):

Es un formato estándar de codificación de caracteres capaz de representar cualquier carácter. Por sus características es recomendado para ser utilizado en la codificación de correos electrónicos y páginas web.

Grafos:

Son un conjunto de elementos enlazados entre sí mediante arcos.

Hardware:

Se refiere a todas las partes físicas o tangibles de un sistema de información.

Hipertexto:

También conocido como hipervínculos. Es una estructura de enlace asociativo en Internet, que permite compartir, agregar y enlazar recursos de diferentes fuentes.



Hoja de Estilo en Cascada:

Es un lenguaje de programación para la web destinado a dar estilo visual.

Identificador Uniforme de Recursos (URI):

Es una dirección exacta y precisa que permite ubicar un recurso en el internet o en una red de cómputos.

Interfaz de Programación de Aplicaciones (API):

Son los procedimientos, funciones o servicios que se encuentran disponibles en determinadas librerías y/o bibliotecas para ser utilizados por otro software.

Interfaz de usuario:

Es el medio por el cual el usuario puede interactuar con un dispositivo o computador.

Internet:

Es una red mundial que conecta computadoras de todos los lugares del mundo para que los usuarios puedan interactuar entre sí sin importar el lugar.

Intérprete Órdenes Seguras (SSH):

Es un protocolo y aplicación por el cual se accede remotamente a una computadora a través de una red de comunicación.

Lenguaje de Consulta Estructurado (SQL):

También conocido como lenguaje de consulta estructurado, es un lenguaje para acceder a bases de datos relacionales que permite la especificación de acciones y operaciones.



Lenguaje de Consulta XML (XQL):

Es un lenguaje que facilita la extracción de datos desde documentos del mismo lenguaje, ofreciendo la posibilidad de realizar consultas flexibles para extraer datos de documentos del XML en la web.

Lenguaje de Hoja de Estilo (LESS):

Es un lenguaje dinámico de hoja de estilos, el cual extiende el lenguaje CSS, añadiendo características que permiten variables, funciones y muchas otras técnicas con las cuales se pueden realizar CSS más fácil de mantener y extensible.

Lenguaje de Hoja de Estilo Extensible para el Formato de Objeto (XSL-FO):

Lenguaje que especifica cómo se presentará en pantalla y en otros medios el formato de unos datos.

Lenguaje de Marcas de Hipertexto (HTML):

Es el lenguaje de programación utilizado para la creación de páginas web.

Lenguaje de Marcas de Hipertexto, versión 5 (HTML5):

Es el lenguaje de programación utilizado para la creación de páginas web. El número 5 que acompaña el término al final hace referencia a su versión 5.

Lenguaje de Marcas Extensible (XML):

Es un lenguaje desarrollado por el Consorcio World Wide Web (W3C) para almacenar datos en forma legible y utilizada para el intercambio de información entre diferentes plataformas.

Lenguaje de Punteros XML (XPointer):

Lenguaje de direccionamiento XML, es un lenguaje que permite el acceso a la estructura interna de un documento XML, esto es, a sus elementos, atributos y contenido.



Lenguaje de rutas XML (XPath):

Lenguaje de rutas XML, es un lenguaje para acceder a partes de un documento XML.

Lenguaje de Transformaciones de Etiquetado Extensible de Hojas de Estilo (XSLT):

Es un estándar, del Consorcio World Wide Web (W3C, por sus siglas en inglés), que presenta como transformar los documentos de Lenguaje de Marcas Extensible (XML, por sus siglas en inglés), en otros formatos.

Lenguaje de vínculos XML (XLink):

Lenguaje de Enlace XML, es un lenguaje que permite insertar elementos en documentos XML para crear enlaces entre recursos del mismo.

Lenguaje Unificado de Modelado (UML):

Es un lenguaje gráfico que visualiza, especifica, construye y documenta un sistema.

Localizador de Recursos Uniforme (URL):

Se utiliza para especificar la dirección exacta de un recurso dentro del portal web.

Mac:

Sistema operativo de disco de Microsoft (MSDOS).

Manejador de Servicio de Información de Internet (IISM):

Es una interfaz gráfica para configurar los grupos de aplicaciones o los sitios Web, FTP, SMTP o NNTP.

Marco de Descripción de Recursos (RDF):

Es un modelo estándar del Consorcio World Wide Web (W3C), diseñado para almacenar datos en forma legible e intercambio de datos en la web.



Marco de Descripción de Recursos en Atributos (RDFa):

Es una técnica que permite proporcionar un conjunto de atributos de marcas para aumentar la información visual en la web.

Marcos de trabajos:

Comúnmente conocido en inglés como Framework, es un concepto que engloba a las características que debe tener un sitio o aplicación web para adaptarse a cualquier dispositivo de forma dinámica.

Metadatos:

Son un conjunto de datos de información que describe las características de otra información. Es “datos sobre datos”.

MySQL:

Es una base de datos para la gestión de base de datos relacional, multihilo y multiusuario.

Navegador Web:

Es un tipo de software utilizado para acceder de forma gráfica a los recursos disponibles en una red o Internet.

Notación de Objetos de JavaScript (JSON):

Es un formato ligero usado como alternativa al XML para intercambio de datos.

Notación Turtle:

Conocido también como Lenguaje de Notación 3 o “N3”, es un lenguaje utilizado para sintaxis XML de RDF.

Objeto:

Es un elemento de datos que consta de un estado y comportamiento el cual se le asigna funcionalidades o tareas realizables.



Openssh:

Es una herramientas utilizada para conectividad remota a través de Internet.

Oracle:

Es una base de datos para la gestión de base de datos objeto-relacional. Es multiplataforma y es conocida por su gran estabilidad y desempeño.

Páginas web:

También llamada página electrónica; no es más que un documento electrónico o una página que presenta cierta información de interés.

Pestañas:

Elemento que pertenece a la interfaz gráfica, el cual le permite al usuario de una aplicación cambiar entre distintos documentos de una forma rápida, normalmente están organizadas en una al lado de otra en fila.

Píxeles:

Es la unidad más pequeña que compone una imagen.

Protocolo de Control de Transmisión (TCP):

Es un protocolo orientado a la conexión de extremo a extremo diseñado para encajar en una jerarquía de capas de protocolos que soportan aplicaciones multi-red.

Protocolo de Seguridad de la Capa de Transporte (TLS):

Se encarga de proporcionar privacidad e integridad de datos entre dos aplicaciones que se comunican a través del internet.

Protocolo de Transferencia de Archivos Seguro (SFTP):

Es un protocolo de red de utilizado para acceder y manejar archivos de manera remota utilizando métodos de encriptación.



Protocolo de Datagrama de Usuario (UDP):

Es un protocolo de transporte de datos, específicamente datagramas, el cual opera en la capa de transporte del modelo OSI.

Protocolo Seguro de Transferencia de Hipertexto (HTTPS):

Es un protocolo utilizado para realizar la transferencia de texto o hipervínculos de manera segura mediante el uso de cifrado, a través de la web.

Protocolo para la transferencia de Archivos (FTP):

Es un protocolo utilizado para la transferencia de archivos.

Protocolo para la Transferencia de Archivos sobre SSL (FTPS):

Es un protocolo utilizado para permitir la negociación de mecanismos de seguridad por medio de la autenticación, confidencialidad e integridad de los mensajes.

Protocolo para la Transferencia de Hipertexto (HTTP):

Es un protocolo utilizado para realizar la transferencia de texto o hipervínculos, a través de la web.

Proxies:

Es un servidor o aplicación utilizado como intermediario para el envío de información en una red de datos.

Red de entrega de Contenidos (CDN):

Es una red de computadores en la cual se disponen copias de datos alojados en diferentes lugares de la red, con el objetivo de que los usuarios puedan tener un acceso más rápido a dichos datos.

Redes Privadas:

Es una red de datos con un alcance geográficamente limitado.



Ruby On Rails:

Es un marco de trabajo de código abierto para aplicaciones web cuyo objetivo es simplificar la escritura de aplicaciones del mundo real con menos código que con otros marcos de trabajo.

Samba:

Samba es una implementación de servicios y protocolos libre.

Sass:

Es un metalenguaje de CSS, el cual se utiliza para describir el estilo de un documento de manera limpia y estructurada.

Servicio de Información de Internet (IIS):

Es un servidor web que permite compartir información con usuarios en Internet, en una intranet o en una extranet.

Servicios criptográficos:

Ayuda a crear medios de comunicación seguros sobre canales que, de otro modo, serían inseguros, proporcionando a su vez integridad de datos y autenticación.

Sintaxis:

Es la instrucción gramática que determina la regla para la implementación de un código.

Sistema de cuadrículas:

Es un sistema referencia el cual se utiliza para realizar localizaciones mediante un plano constituido por cuadrículas.

Sistema de Organización del Conocimiento (KOS):

Este término engloba diferentes esquemas que permiten crear y organizar la información, y promover la administración del conocimiento.



Sistema operativo:

Es un software utilizado en los sistemas de información para gestionar y administrar los recursos de los dispositivos o computadores.

Software:

Se refiere a todos los componentes lógicos o intangibles de un sistema de información, tales como programas, aplicaciones, sistemas operativos, entre otros.

Solicitud de Certificación (CSR):

Es la forma los servicios y sitios web en el Internet obtienen una validación para el cifrado de los datos que se envían por la red.

SparQL:

Es un lenguaje de consulta para grafos estandarizados en RDF.

SQL Server:

SQL Server es un sistema creado por Microsoft creado para la administración y gestión de bases de datos relacionales.

Tabla hash:

Es una estructura de datos donde se asocia llaves o claves con valores.

Texto plano:

También conocido como textos llanos, son archivos informáticos que están compuesto solo por texto sin ningún tipo de formato.

Ubuntu:

Es un sistema operativo de código abierto que funciona con funcionalidad para trabajar en diferentes plataformas.

UNIX:

Es un sistema operativo multitarea desarrollado por Laboratorios Bell.



Valores Separados por Comas (CSV):

Es un formato de archivo de datos que su contenido está separado por comas.

Valores Separados por Tabulaciones (TSV):

Es un formato de texto simple utilizado para el almacenamiento de información en forma de tablas. En este, cada registro de la tabla representa una línea del archivo de texto.

Vectores:

También conocido como matriz, es una zona de almacenamiento continuo, donde se presentan una serie de elementos de la misma clase.

Versionamiento:

Es el proceso de administración y gestión de los distintos cambios que experimenta un código fuente, producto o elemento.

Vocabulario para Catálogo de Datos (DCAT):

Es un estándar definido por el Consorcio World Wide Web (W3C) y diseñado para facilitar la interoperabilidad entre catálogos de datos publicados en la web.

ZIP:

Es un formato de compresión de archivos sin pérdida que comprime cada uno de los archivos de forma separada.



ABREVIATURAS Y ACRÓNIMOS

NO.	ABREVIATURAS Y ACRÓNIMOS	INGLÉS	ESPAÑOL
1	OPTIC	N/D	Oficina Presidencial de Tecnologías de la Información y Comunicación
2	ENAT	N/D	Estandarización, Normativas y Auditoría Técnica
3	NIU	N/D	Número de Identificación Único
4	CIEN	N/D	Comité Interno para Evaluación de las Normas
5	COETIC	N/D	Comité de Estándares de Tecnologías de la Información y Comunicación
6	TIC	N/D	Tecnología de la Información y la Comunicación
7	SFTP	SSH File Transfer Protocol	Protocolo de Transferencia de Archivos Seguro
8	MAP	N/D	Ministerio de Administración Pública
9	DCAT	Data Catalog	Catálogo de Datos
10	URI	Uniform Resource Identifier	Identificador de Recursos Uniforme



11	KOS	Key Organization System	Valores del Sistema de Organización
12	FTP	File Transfer Protocol	Protocolo de Transferencia de Archivos
13	HTTP	Hypertext Transfer Protocol	Protocolo de Transferencia de Hipertexto
14	HTTPS	Hypertext Transfer Protocol Secure	Protocolo Seguro de Transferencia de Hipertexto
15	FTPS	File Transfer Protocol Secure	Protocolo de Transferencia de Archivos Seguro
16	TLS	Transport Layer Security	Protocolos de Seguridad de la Capa de Transporte
17	SSL	Secure Sockets Layer	Capa de Conexión Segura
18	SSH	Secure Shell	Intérprete de Órdenes Segura
19	IP	Internet Protocol	Protocolo de Internet
20	MS-DOS	MicroSoft Disk Operating System	Sistema operativo de disco de Microsoft
21	URL	Uniform Resource Locator	Localizador de recursos uniforme
22	CSR	Certificate Signing Request	Solicitud de Certificación
23	IIS	Internet Information Services	Servicio de Información de Internet de Microsoft



24	SQL	Structured Query Language	Lenguaje de Consulta Estructurado
25	ID	Unique Identifier	Identificador Único
26	PK	Primary Keys	Claves Primarias
27	FK	Foreign Keys	Claves Foráneas
28	I	Index	Índice
29	DDL	Data Definition Language	Lenguaje de Definición de Datos
30	DML	Data Manipulation Language	Lenguaje de Manipulación de Datos
31	ASCII	American Standard Code for Information Interchange	Código Estándar Estadounidense para el Intercambio de Información
32	ISO	Organización Internacional de Normalización	Organización Internacional de Normalización
33	UTF-8	8-bit Unicode Transformation Format	Formato de Transformación Unicode de 8-bit
34	JSON	JavaScript Object Notation	Formato de Notación de Objetos de JavaScript
35	XML	eXtensible Markup Language	Lenguaje de Marcas Extensible
36	API	Application Programming Interface	Interfaz de Programación de Aplicaciones
37	B2B	Business-to-business	Negocio a Negocio



38	CSV	Comma-Separated Values	Formato de Valores Separados por Coma
39	TSV	Tab-separated values	Valores Separados por Tabulaciones
40	UML	Unified Modeling Language	Lenguaje Unificado de Modelado
41	MDD	Managing Deliberate Defaulters	Diagrama de Dependencia entre Módulos
42	TDD	Test-Driven Development	Desarrollo Guiado por Pruebas de Software
43	BDD	Behavior-Driven Development	Desarrollo Guiado por Comportamiento



BIBLIOGRAFÍA

- Comisión Económica para América Latina y el Caribe (CEPAL). (2007). Libro blanco de interoperabilidad de gobierno electrónico para América Latina y el Caribe. División de Desarrollo Productivo y Empresarial de la CEPAL.
- Consorcio World Wide Web (W3C). (2008). Sintaxis XML Signature y Procesamiento. Segunda Edición. New York.
- Criado, J. I., Mila Gascó, & Carlos E. Jiménez. (2010). Bases para una Estrategia Iberoamericana de Interoperabilidad. Documento para la consideración de la XII Conferencia Iberoamericana de Ministros de Administración Pública y Reforma del Estado.
- European Telecommunications Standards Institute. (2003). ETSI TR 102 272. Electronic Signatures and Infrastructures (ESI); ASN.1 format for signature policies. República Francesa.
- European Telecommunications Standards Institute. (2009). ETSI TS 101 903. XML Advanced Electronic Signatures (XAdES). República Francesa.
- Gobierno Bolivariana de Venezuela. (2010). Marco de Interoperabilidad. Marco de Interoperabilidad para el Estado Venezolano V 1.0. República Bolivariana de Venezuela: Publicsol 50 C.A.
- Gobierno Bolivariano de Venezuela. (2012). Avances en la Implementación de la Interoperabilidad en Venezuela. República Bolivariana de Venezuela: Centro Nacional de Tecnología de información (CNTI).



- Gobierno Bolivariano de Venezuela. (2013). Interoperabilidad en Venezuela. Integrando los Servicios del Estado. República Bolivariana de Venezuela.
- Gobierno de España. (2010). Real Decreto 4/2010, de 8 de enero, por el que se regula el Esquema Nacional de Interoperabilidad en el ámbito de la Administración Electrónica. España.
- Gobierno de España. (2011). Guía de Aplicación de la Norma Técnica de Interoperabilidad. Digitalización de Documentos. Madrid: Dirección General para el Impulso de la Administración Electrónica.
- Gobierno de España. (2012). Catálogo De Estándares. Guía de aplicación de la Norma Técnica de Interoperabilidad. España: Ministerio de Hacienda y Administraciones Públicas Secretaría General Técnica.
- Gobierno de la Republica Dominicana . (2012). Ley No. 1-12. Ley Orgánica de la Estrategia Nacional de Desarrollo de la República Dominicana 2030. Santo Domingo, Republica Dominicana: Congreso de la República Dominicana.
- Grupo de Trabajo de la Red de S. Josefsson, Ed. (2003). El Base16, Base32, y Base64 de datos Codificaciones.
- Moreno, J. L. (s.f.). Gestión estratégica de la interoperabilidad. Necesidades normativas en la adaptación a los cambios sociales y tecnológicos.
- Network Working Group. (2008). The Transport Layer Security (TLS) Protocol).



- Oficina Nacional de Gobierno (ONGEI). (s.f.). Interoperabilidad en el Estado Peruano. República del Perú: Oficina Nacional de Gobierno Electrónico e Informática.
- Oficina Presidencial de Tecnología de la Información y Comunicación . (2013). NORTIC A2. República Dominicana.
- Oficina Presidencial de Tecnología de la Información y Comunicación. (2014). NORTIC A3. República Dominicana.

Organización Internacional de Normalización (ISO). (1986). ISO 8879. Procesamiento de la información - texto y de oficina - sistemas Generalizado Estándar Markup Language (SGML).

- Organización Internacional de Normalización (ISO). (1994). ISO/IEC 7498-1. Information technology ~ Open Systems Interconnection ~ Basic Reference Model: The Basic Model.
- República de Colombia. (2010). Marco de Interoperabilidad para Gobierno en línea. Manual para la Interoperabilidad del Gobierno en línea. República de Colombia.



ANEXOS

Anexo A. Acta de constitución.

NOMBRE DEL PROYECTO O SERVICIO	Implementación de Sistema de CRM de Atención al Ciudadano
PARTES INVOLUCRADAS EN EL PROYECTO	OPTIC: Luis Sánchez - Desarrollador / Alberto Méndez - Técnico en Infraestructura
DIRECTOR DEL PROYECTO O SERVICIO	Ing. José Pérez - Gerente de aplicaciones
DESCRIPCIÓN DEL PROYECTO O SERVICIO	
Implementación de sistema de Sugar CRM en Linux. Automatizar los procesos mediante formularios digitales que se ejecutan sobre flujos de trabajos. Se creará un reporte estadísticos para medir la eficiencia de respuesta de los casos.	
REQUISITOS DEL PROYECTO O SERVICIO	
I. Requisitos del Cliente:	
<ul style="list-style-type: none"> a. Servidor Linux con al menos los requerimientos mínimos indicados por el producto de CRM. b. Disponibilidad del personal involucrado. 	
II. Requisitos Legales y/o reglamentarios:	
<ul style="list-style-type: none"> a. Cumplimiento de NORTIC A1:2015. b. Cumplimiento de NORTIC A4:2014. 	
III. Otros Requisitos:	
<ul style="list-style-type: none"> a. N/A 	
CRITERIOS DE ACEPTACIÓN DEL PROYECTO O SERVICIO	
<ul style="list-style-type: none"> 1. Plataforma de Sugar CRM correctamente instalada. 2. Formularios con los campos requeridos y validaciones desarrolladas correctamente. 3. Información que muestra Reporte de Resolución de Incidentes con datos correctos. 4. Personal completamente adiestrado en la herramienta y en el proceso de Registro de Incidentes. 	



RIESGOS INICIALES	
<ol style="list-style-type: none"> 1. Que el hardware no esté disponible. 2. Personal indispuerto. 3. Cambio de Personal durante el proyecto. 4. Retrasos en la creación de las cuentas de usuario necesarias para el acceso a los servidores. 	
OBJETIVOS GENERALES DEL PROYECTO	OBJETIVOS ESPECÍFICOS DEL PROYECTO
Automatizar el proceso de atención al ciudadano y hacerlo eficiente.	<ul style="list-style-type: none"> • Disminuir el tiempo de respuesta de los casos de los ciudadanos. • Tener una plataforma tecnológica que permita relacionar la información del ciudadano y sus casos históricos. • Tener la capacidad de enviar notificaciones automáticas sobre el estatus de los casos abiertos de los ciudadanos.

RESUMEN DE HITOS	PLAZO
Culminación de Levantamiento de información.	2 días laborables
Entrega de Documento de Alcance.	1 semanas
Culminación de diseño y desarrollo de formulario de Información General del ciudadano.	2 semanas
Culminación de diseño y desarrollo de formulario de Registro de Caso del ciudadano.	4 semanas
Culminación de diseño y desarrollo de Reporte Resumen de Incidentes.	1 semanas
Finalización de Entrenamiento de Usuarios.	2 semanas
Finalización de Entrenamiento Técnico.	5 semanas
Entrega de Documentación Técnica.	1 semanas
Finalización y Entrega formal del proyecto.	6 semanas

Tiempo:

Total estimado:	22 semanas y 2 días
-----------------	---------------------



PRESUPUESTO ESTIMADO

RD\$ 2,000,000.00

NIVELES DE ESCALAMIENTO

Nivel	Nombre	Cargo/Institución	Teléfono	Correo electrónico
1	Ing. José Pérez	Gerente de Aplicaciones / OPTIC	809-111-6699	jp@optic.gob.do
2	Carlos Rosa	Manejador de Proyectos / OPTIC	829-777-1111	cr@optic.gob.do
3	Armando García	Director de la OPTIC / OPTIC	809-222-4444	ag@optic.gob.do

APROBACIÓN DEL ACTA DE CONSTITUCIÓN DE PROYECTO O SERVICIO

[Especificar nombre del Project Manager o Director de Departamento]
 [Especificar Cargo]
 [Especificar Institución]

José Pérez
 Gerente de Aplicaciones
 Oficina Presidencial de Tecnologías de la Información y Comunicación (OPTIC)

Anexo B. Acta de cierre.

NOMBRE DEL PROYECTO	Implementación de Sistema de CRM de Atención al Ciudadano
PARTES INVOLUCRADAS EN EL PROYECTO	OPTIC: Luis Sánchez - Desarrollador / Alberto Méndez - Técnico en Infraestructura
DIRECTOR DEL PROYECTO O SERVICIO	Ing. José Pérez - Gerente de aplicaciones
DESCRIPCIÓN DEL PROYECTO	
Implementación de sistema de Sugar CRM en Linux. Automatizar los procesos mediante formularios digitales que se ejecutan sobre flujos de trabajos. Se creará un reporte estadísticos para medir la eficiencia de respuesta de los casos.	



INFORMACIÓN OBJETIVOS LOGRADOS/ACUERDO DE CIERRE (CRITERIOS DE ACEPTACIÓN)

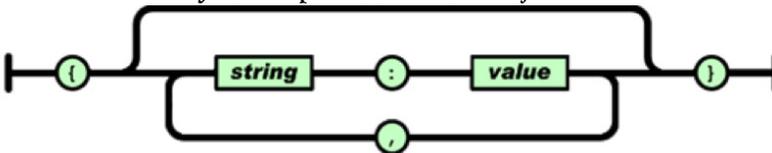
1. Plataforma de Sugar CRM correctamente instalada.
2. Formularios con los campos requeridos y validaciones desarrolladas correctamente.
3. Información que muestra Reporte de Resolución de Incidentes con datos correctos.
4. Personal completamente adiestrado en la herramienta y en el proceso de Registro de Incidentes

APROBACIÓN DEL ACTA DE CIERRE DEL PROYECTO

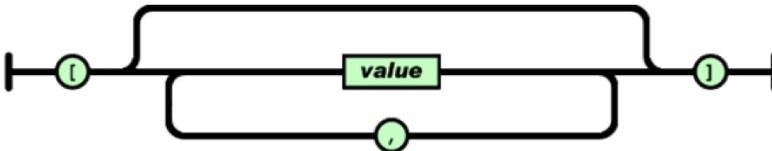
[Especificar nombre del Project Manager
o Director de Departamento]
[Especificar Cargo]
[Especificar Institución]

José Pérez
Gerente de Aplicaciones
Oficina Presidencial de Tecnologías de la
Información y Comunicación (OPTIC)

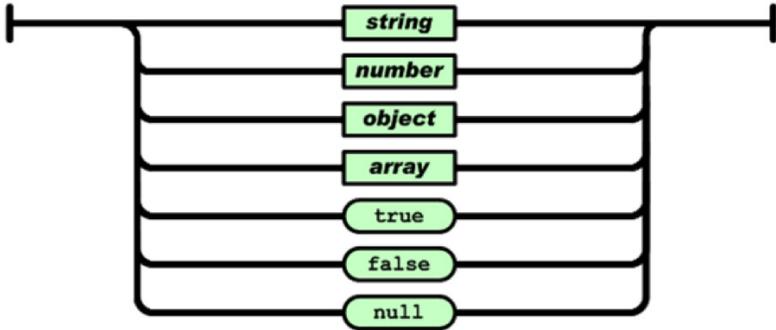
Anexo C. Diagrama de la sintaxis de JSON representada en “Objeto”.



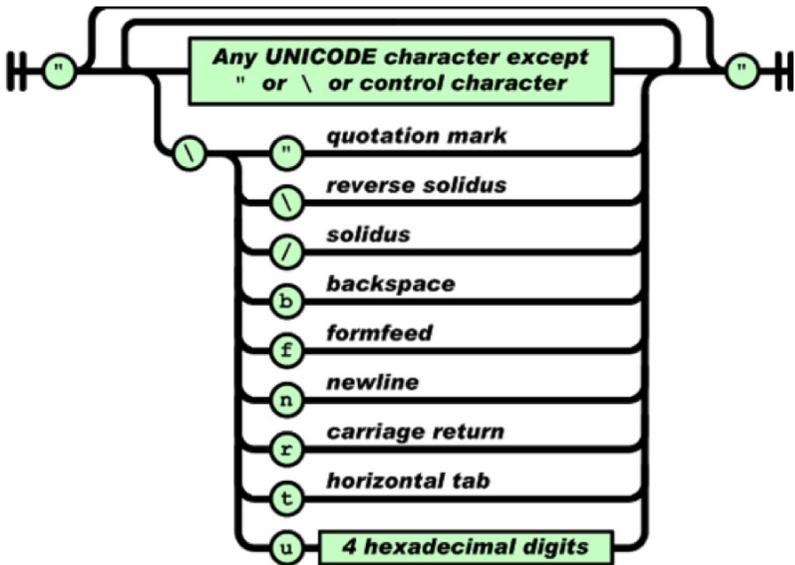
Anexo D. Diagrama de la sintaxis de JSON representada en “Arreglo”.



Anexo E. Diagrama de la sintaxis de JSON representada en “Valor”.

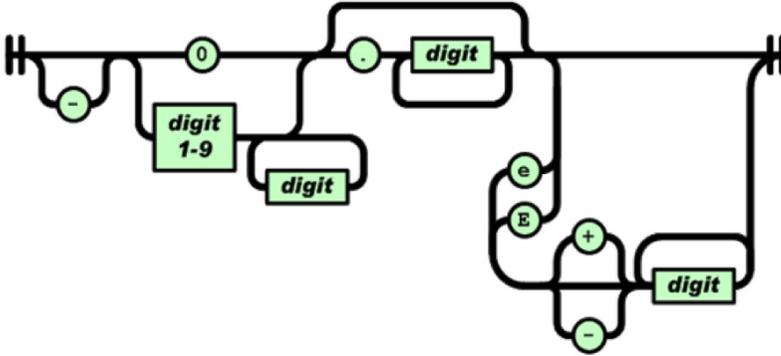


Anexo F. Diagrama de la sintaxis de JSON representada en “Cadena”.



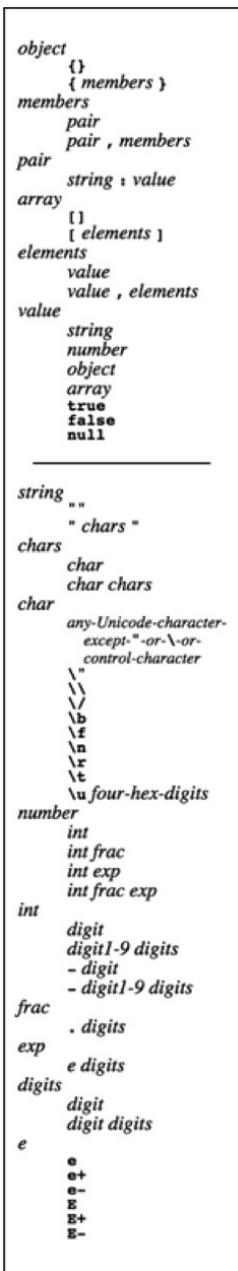


Anexo G. Diagrama de la sintaxis de JSON representada en “Numero”.





Anexo H. Diagrama de la sintaxis de JSON representada en “Notación”.





EQUIPO DE TRABAJO

Dirección General

Armando García, Director General

Departamento de Estandarización, Normativas y Auditoría Técnica

Elvyn Peguero, Gerente

Shalem Pérez, Auditor de Estándares NORTIC

Winner Núñez, Auditor de Estándares NORTIC

Ginsy Aguilera, Consultora de Estándares y Normativas

Ariel Acosta, Consultor de Estándares y Normativas

Hamlet Durán, Analista de Estándares y Normativas

Comité Interno para Evaluación de las Normas - Equipo OPTIC

José Luis Liranzo, Director de Estudios, Investigación y Estrategia de
Gobierno Electrónico

Pedro Gomez, Director de Planificación y Desarrollo

Charli Polanco, Director de TIC

Miguel Guerra, Gerente Multimedia

Asesor

Luis Guzmán, experto en TIC

Colaboradores

Miguel Rodríguez, OPTIC

Eliaguín Encarnación, OPTIC



Para descarga este
documento leer este
codigo.

Av.27 de Febrero #419, Santo Domingo, R.D.
Tel.:+ 809.286.1009 info@optic.gob.do
www.optic.gob.do www.dominicana.gob.do



OpticRD