



UNIVERSIDAD CATÓLICA DE SALTA

Facultad de Ingeniería

Carrera: Ingeniería en Telecomunicaciones

PROYECTO FINAL DE GRADO

Diseño de Red Escalable para E-Government

Alumna: Cecilia Inés Castro

Director: Ing. Pablo Sebastián Vargas

SALTA - 2019



INGENIERÍA EN TELECOMUNICACIONES

Director: **Ing. Pablo Sebastián Vargas**

Firma:

Tribunal Evaluador:

.....

Firma:

.....

Firma:

.....

Firma:

Fecha de Exposición del trabajo:.....

AGRADECIMIENTOS

Quiero dar gracias a Dios y a la Virgen María por estos años de carrera, por todo lo vivido y aprendido.

Dar gracias a mi familia, por todo el apoyo, el esfuerzo y la confianza que depositaron en mí y el acompañamiento a lo largo de todos estos años.

Quiero agradecer a todos aquellos profesores que fueron parte de mi formación, especialmente al Ing. Pablo Vargas, director de esta tesis, y al Ing. Roberto Breslin, jefe de carrera. No me quiero olvidar de Pato y Edith por su disposición para resolver mis dudas y problemas administrativos.

También quiero agradecer a mis compañeros, sobre todo a Marcos, Nicolás y Rodolfo, con los que compartí desde el primer día y seguimos juntos a la par hasta el final de esta carrera.

A Uds., quiero dedicarles este proyecto con el que concluyo esta etapa de mi vida.

Cecilia Inés Castro

INDICE GENERAL

	Pág.
Abstract	10
CAPÍTULO 1: INTRODUCCION	
1.1 – El problema, su importancia	11
1.2 - metodología de trabajo	11
CAPÍTULO 2: ELECTRONIC-GOVERNMENT	
2.1 – Aproximación a la conceptualización de Electronic-Government	12
2.2 – Inicios de e-Gov	13
2.3 – Modelos de Internación de e-Gov	14
2.4 – Objetivos de un e-Gov	14
2.5 – Modelos de desarrollo de e-Gov	15
<i>Modelo de Gartner</i>	15
<i>Fases que promueven la participación ciudadana en el e-Gov</i>	16
2.6 – Requisitos para la implementación del e-Gov	16
2.7 – Beneficios del e-Gov	17
CAPÍTULO 3: ESTADO ACTUAL DEL E-GOV EN EL MUNDO. CASOS DE ESTUDIO	
3.1 – Estado actual del e-Gov en el mundo	18
<i>El EGDI</i>	18
<i>El EPI</i>	18
<i>Ranking mundial según el EGDI</i>	19
3.2 - El Grupo D, de los Gobiernos Digitalmente más avanzados	21
<i>Principios del Desarrollo Digital</i>	21
3.3 – Caso de Estudio N° 1: el e-Gov de Estonia	22
<i>El Portal de e-Gov</i>	23
3.4 – Aspectos tecnológicos del e-Gov de Estonia	24
<i>Servicios de Interoperabilidad con Tecnología X-Road</i>	24
<i>Servicios de Seguridad y Encriptación con Tecnología KSI</i>	25
3.5 – Caso de Estudio N° 2: el e-Gov de Uruguay	26
<i>El Portal de e-Gov</i>	26
3.6 – Arquitectura Integrada de Gobierno de Uruguay	27
<i>Modelo de Referencia Tecnológico (MRT)</i>	29
3.7 – Tecnologías de la Plataforma de e-Gov de Uruguay	30
<i>Plataforma de Interoperabilidad</i>	31
<i>Tecnología SOA como base de la PDI</i>	31
<i>Tecnología REDuy en la PDI</i>	32
<i>E-Gov de Uruguay en la Nube: IaaS y PaaS</i>	32
3.8 – Caso de Estudio N° 3: El e-Gov en Argentina	33
<i>Sus inicios y evolución a la fecha</i>	33
<i>El portal de e-Gov de Argentina</i>	34
<i>Mediciones de e-Gov de Argentina</i>	35
3.9 – Las redes sociales digitales y el e-Gov	36
<i>Uso de las redes sociales digitales a nivel mundial y en la Argentina</i>	37
CAPÍTULO 4: ARQUITECTURA DE SERVICIOS WEB	
4.1 – Definiendo Servicio Web	39
4.2 – Componentes y roles de un servicio web	40
4.3 – Funcionamiento de un servicio web	42
4.4 – Ventajas de los servicios web	43
4.5 – El Web Service	43
<i>Plataforma de servicios web de Primera Generación</i>	44
<i>Plataforma de servicios web de Segunda Generación (extensiones WS- *)</i>	44

CAPÍTULO 5: ESTANDARES BASICOS

5.1 – Introducción	45
5.2 – XML: eXtensible Markup Language	45
XML Schema	46
DTD y XSD	46
5.3 – WSDL: Lenguaje de Descripción de servicios web	47
Estructura de un documento WSDL	47
5.4 – Estándar UDDI	48
Registro UDDI	48
5.5 – Protocolo HTTP	49
Atención de una solicitud	49
Mensaje HTTP	50
Seguridad con HTTPS	50
Versión mejorada con HTTP/2	51
5.6 – Protocolo SOAP	51
SOAP y su interacción con otras plataformas	51
Formato del mensaje SOAP	52
Comunicación cliente-servidor mediante SOAP	53
5.7 – La tecnología REST	53
Descripción de REST	53
Esquema de REST	54
Comparación entre SOAP y REST	54
Principios de diseño	55

CAPÍTULO 6: ARQUITECTURA ORIENTADA A SERVICIOS

6.1 – Conceptualizando SOA	56
Algunas definiciones de SOA	56
Evolución de la arquitectura	57
Características de SOA	58
6.2 – Los servicios	58
Clasificación de los servicios	59
6.3 – Modelo conceptual para SOA	60
6.4 – Diseño y desarrollo de aplicaciones SOA	62
Principios de diseño	62
Metodología de análisis y diseño orientado a servicios	63
6.5 – Tecnologías y Estándares en SOA	64
WS-CDL	64
WS-BPEL	65
6.6 – SOA y los Microservicios	65
6.7 – Beneficios de la implementación de SOA	66

CAPÍTULO 7: ESCALABILIDAD

7.1 – Conceptualizando la escalabilidad	67
Balance de carga	68
7.2 – Internet, una Red Escalable	68

CAPÍTULO 8: REDES DE SIGUIENTE GENERACION

8.1 – Red NGN	72
SDH	72
Evolución hacia una red de Transporte de Próxima Generación	73
8.2 – Arquitectura NGN	74
8.3 – Aplicación de ASON	75
8.4 – IP/MPLS	76
MPLS Generalizado	78
8.5 – Protocolo IMS	79

Arquitectura IMS	79
8.6 - Softswitch	80
Arquitectura de Servicios del Softswitch	81
CAPÍTULO 9: INFRAESTRUCTURAS HIPERCONVERGENTES	
9.1 – Computación en la nube	82
9.2 – Tecnologías hiperconvergentes	82
La infraestructura hiperconvergente, para organizaciones en crecimiento	83
IaaS	83
PaaS	83
SaaS	84
CAPÍTULO 10: DISEÑO DE RED ESCALABLE PARA E-GOV	
10.1 – Etapas del diseño	85
10.2 - Diseño	86
Etapa 1: Definición del ámbito de e-Gov para esta propuesta	86
Etapa 2: Selección de los modelos de comunicación del E-GovG a adoptar	86
Etapa 3: Definición del tipo de Escalabilidad a lograr en E-GovG	87
Etapa 4: Provisión de la infraestructura de la red	88
Etapa 5: Diseño de la Arquitectura General de la red E-GovG	90
Etapa 6: Diseño de la accesibilidad para la plataforma de E-GovG	92
Etapa 7: Implantación del modelo SOA con servicios web	92
10.3 - Ventajas del diseño E-GovG	96
CONCLUSIÓN	98
ACRÓNIMOS	100
BIBLIOGRAFÍA	104

Índice de Tablas

	Pág.
<i>Tabla 2.1: Modelos de desarrollo de e-Gov. Fuente Reporte de las Naciones Unidas</i>	15
<i>Tabla 3.1: Ranking de los 15 mejores EGDI 2018. Fuente UN-E-Government Knowledgebase</i>	20
<i>Tabla 3.2: Ranking de los 6 mejores EDGI 2018 de América. Fuente UN-E-Government Knowledgebase</i>	20
<i>Tabla 3.3: Performance jurisdiccional según subíndice y jurisdicción. Fuente IARAF</i>	36
<i>Tabla 5.1: Comparación entre SOAP y REST</i>	55

Índice de Ilustraciones

	Pág.
<i>Figura 3.1: EGDI 2018. Fuente: UN-E-Government Knowledgebase, 2018.</i>	19
<i>Figura 3.2: El D9 en la cumbre del 21 y 22 de noviembre de 2018 celebrada en Jerusalén. Fuente: Portal AGESIC, 2018.</i>	21
<i>Figura 3.3: EGDI y EPI 2018 de Estonia. Fuente: UN-E-Government Knowledgebase, 2018.</i>	22
<i>Figura 3.4: Portal de e-Gov de Estonia.</i>	24
<i>Figura 3.5: EGDI y EPI 2018 de Uruguay. Fuente: UN-E-Government Knowledgebase, 2018.</i>	26
<i>Figura 3.6: Portal Web de Uruguay.</i>	26
<i>Figura 3.7: AIG de Uruguay basada en TOFAG. Fuente: AGESIC.</i>	27
<i>Figura 3.8: MRN, MRA y MRS de AIG de Uruguay. Fuente: AGESIC.</i>	28
<i>Figura 3.9: MRD de AIG de Uruguay. Fuente: AGESIC.</i>	28
<i>Figura 3.10: MRS y MRDe de AIG de Uruguay. Fuente: AGESIC.</i>	29
<i>Figura 3.11: MRT de e-Gov de Uruguay. Fuente: AGESIC, 2018.</i>	29
<i>Figura 3.11: Esquema de integración de la PGE de Uruguay. Fuente: AGESIC, 2018.</i>	30
<i>Figura 3.12: PDI. Fuente: AGESIC, 2018.</i>	31
<i>Figura 3.14: REDuy. Fuente: AGESIC, 2018.</i>	32
<i>Figura 3.15: Tecnologías hiperconvergentes.</i>	32
<i>Figura 3.16: EGDI y EPI 2018 de Argentina. Fuente: UN-E-Government Knowledgebase, 2018.</i>	33
<i>Figura 3.17: Plataforma e-Gov de Argentina</i>	35
<i>Figura 3.18: Componentes de ISAGEL. Fuente: Informe Económico N° 387 de IARAF.</i>	35
<i>Figura 3.19: Ranking ISAGEL 2016. Estimación periodo julio-agosto 2016. Fuente: IARAF.</i>	36
<i>Figura 3.20: Distribución de portales web gubernamentales de redes sociales según orientación. Fuente: Informe Económico 387 de IARAF.</i>	37
<i>Figura 3.21: Distribución de portales web gubernamentales de redes sociales según Área de Gobernación. Fuente: Informe Económico 387 de IARAF.</i>	37
<i>Figura 4.1: Ventajas del uso de servicios web. Fuente: Desarrolloconsoa, 2014.</i>	40
<i>Figura 4.2: Roles de un servicio web: a) Un servicio web como proveedor de servicios; b) partes de un servicio web que actúan como consumidor de servicios; c) Transición de un servicio web a través de los roles del proveedor a consumidor de servicios. Fuente: Thomas Erl, 2005.</i>	42
<i>Figura 4.3: Esquema de funcionamiento de un servicio web. Fuente: Voorman H., 2006.</i>	42
<i>Figura 4.4: Componentes de un Servidor de servicios web.</i>	43
<i>Figura 5.1: Estándares básicos.</i>	45
<i>Figura 5.2: XML, su relación con SGML.</i>	46
<i>Figura 5.3: HTML describe formato; XML describe estructura y semántica.</i>	46
<i>Figura 5.4: Acceso a un servicio web usando archivos WSDL. Fuente: Desarrolloconsoa, 2014.</i>	47
<i>Figura 5.5: Estructura de un archivo WSDL.</i>	48
<i>Figura 5.6: Repositorio UDDI con archivos WSDL.</i>	49
<i>Figura 5.7: a) HTTP en el esquema cliente-servidor; b) respuesta exitosa; c) respuesta no exitosa.</i>	49
<i>Figura 5.8: Estructura de un mensaje HTTP.</i>	50
<i>Figura 5.9: Comunicación segura con HTTPS, usando encriptación. Fuente: OSI-españa, 2014.</i>	51
<i>Figura 5.10: SOAP usando XML, sobre HTTP.</i>	52
<i>Figura 5.11: SOAP y su interacción con otras plataformas. Fuente: Cubo Velázquez,</i>	52

2018.	
Figura 5.12: Mensaje SOAP en la comunicación cliente-servidor.	52
Figura 5.13: Esquema de comunicación cliente-servidor mediante SOAP.	53
Figura 5.14: REST en SOA.	54
Figura 5.15: Comunicación en REST.	55
Figura 6.1: Enfoque SOA. Fuente: Consultoría CTP, 2007.	57
Figura 6.2: Evolución de SOA. Fuente: Escobar y otros, 2006.	57
Figura 6.3: a) Integración punto a punto; b) integración de aplicaciones empresariales; c) integración orientada a servicios. Fuente: Consultoría CTP, 2007.	58
Figura 6.4: Documentos del contrato de un servicio web.	59
Figura 6.5: Modelo conceptual de SOA. Fuente: Rojas D., 2009.	61
Figura 6.6: Componentes del Modelo conceptual de SOA. Fuente: www.soagenda.com .	62
Figura 6.7: Principios de Diseño SOA. Fuente: Antonio Barco, 2006.	63
Figura 7.1: Escalabilidad vertical y horizontal.	68
Figura 7.2: Nivel 1 – los ISP1 y el backbone de Internet. Fuente: CCNA, 2015.	69
Figura 7.3: Nivel 2 – los ISP2. Fuente: CCNA, 2015.	69
Figura 7.4: Nivel 2 – los ISP2. Fuente: CCNA, 2015.	70
Figura 7.5: El DNS. Fuente: CCNA, 2015.	70
Figura 7.6: La conectividad distribuida. Fuente CCNA, 2015.	70
Figura 7.7: La conectividad entre pares. Fuente CCNA, 2015.	71
Figura 8.1: Trama con SDH. Fuente: ADSLFAQS, 2018.	73
Figura 8.2: Evolución de las redes de transporte.	74
Figura 8.3: Arquitectura NGN. Fuente: Rico-Bautista, 2017.	75
Figura 8.4: Esquema de aplicación de ASON.	76
Figura 8.5: Esquema de comunicación MPLS.	77
Figura 8.6: Esquema de transporte con MPLS.	77
Figura 8.7: Jerarquías de las interfaces de control.	78
Figura 8.8: Esquema de integración IP/MPLS.	79
Figura 8.9: Esquema de la arquitectura de IMS.	80
Figura 9.1: IaaS, PaaS, SaaS. Fuente: Bitec, 2018.	82
Figura 10.1: E-GovG, un e-Gov para cualquier ámbito.	85
Figura 10.2: Esquema de interacción con E-GovG.	86
Figura 10.3: Nodos T y F para la escalabilidad horizontal.	86
Figura 10.4: Nodos para la escalabilidad en la nube.	87
Figura 10.5: Escalabilidad horizontal para expandir la funcionalidad de E-GovG.	87
Figura 10.6: Escalabilidad Vertical para una mejorar la infraestructura y capacidad computacional.	88
Figura 10.7: E-GovG con recursos propios y E-Gov con recursos alquilados.	89
Figura 10.8: Core de E-GovG.	89
Figura 10.9: Mapa de red del gobierno de la República Argentina. Fuente: Castillo, Echenique y García, 2017.	89
Figura 10.10: Mapa de red del gobierno de la República Argentina – Zona Noroeste. Fuente: Castillo, Echenique y García, 2017.	90
Figura 10.11: Arquitectura general de la red E-GovG.	91
Figura 10.12: Capa de acceso a la red E-GovG.	92
Figura 10.13: Modelo de Protocolo de E-GovG.	93
Figura 10.14: Implantación del Modelos SOA en Servicios Web. Paso 1 y 2.	93
Figura 10.15: Implantación del Modelos SOA en Servicios Web. Paso 3.	94
Figura 10.16: Implantación del Modelos SOA en Servicios Web. Paso 4.	94
Figura 10.17: Implantación del Modelos SOA en Servicios Web. Paso 5.	95
Figura 10.18: Implantación del Modelos SOA en Servicios Web. Paso 6 y 7.	95

ABSTRACT

En el presente trabajo se propuso diseñar una red escalable para e-Gov. Para ello se organizaron las diferentes actividades en tres etapas.

En la primera etapa se llevaron a cabo tareas de relevamiento, investigación y estudio de casos sobre gobiernos electrónicos actuales, tanto a nivel mundial, como nacional y provincial. Se estudiaron casos particulares que se destacan como el e-Gov de Estonia y de Uruguay, por ser considerados “países altamente digitalizados”. Así también se abordó la situación de e-Gov de la República Argentina y sus provincias. Toda el relevamiento se documentó en los capítulos 2 y 3.

En la segunda Parte se estudiaron las tecnologías emergentes para la configuración de redes escalables, como arquitectura de servicios web, estándares básicos, paradigma SOA, escalabilidad, redes NGN, tecnologías convergentes, entre otras. Toda la fundamentación teórica necesaria se documentó en los capítulos 4, 5, 6, 7, 8 y 9.

En la tercera Parte se planteó el diseño de la red escalable para e-Gov integrando todos los fundamentos teóricos propuestos en los capítulos anteriores. El diseño se encuentra en el capítulo 10.

La tesis debe contribuir decisivamente a la formación del candidato, que lo enriquezca para el resto de su carrera; por otra parte, debe ser una aportación considerable al esclarecimiento de alguna cuestión interesante.

Julián Marías
Filósofo español católico (1914-2005)

CAPITULO 1: INTRODUCCION

1.1 - El problema, su importancia

E-Government (e-Gov) o Gobierno Electrónico puede definirse como una nueva forma de interacción entre el gobierno y sus respectivos ciudadanos. Esta nueva forma consiste en la implementación, desarrollo y aplicación de las Tecnologías de la información y la comunicación (TICs), en busca de una mejora en la eficacia y eficiencia de los procesos gubernamentales internos y en las vinculaciones con la sociedad. Para tal fin usa diversas tecnologías (telefonía, sistemas de vigilancia, identificación por sistemas de radiofrecuencia e incluso la televisión y la radio) para automatizar las respuestas por parte de la administración ante una creciente demanda de una sociedad cada vez más informatizada. Este servicio debe caracterizarse por garantizar la celeridad, transparencia, inclusión, información y creación de nuevos medios de diálogo e intercambio.

Ante una progreso tecnológico que fluye con tanta rapidez, y que genera nuevas configuraciones en la sociedad y nuevos paradigmas en la comunicación, resulta necesario que los gobiernos se adapten a estos cambios y adopten las nuevas TICs para “gobernar” una sociedad tecnológicamente emergente: la sociedad de la información y el conocimiento.

Ante este escenario, resulta imprescindible **incursionar, investigar, estudiar y plantear diseños de sistemas escalables, que permitan desarrollar sistemas de e-Gov en forma incremental**, es decir que posibiliten expandir la funcionalidad y el alcance de las operaciones gubernamentales a fin de facilitar el acceso a todos los ciudadanos de un país.

1.2 – Metodología de trabajo

Para realizar este trabajo se ejecutaron varias etapas:

Etapas 1: Búsqueda de antecedentes a nivel mundial, en América y en la Argentina.

Etapas 2: Búsqueda Bibliográfica sobre infraestructura y estándares de telecomunicaciones, e-Gov, Servicios web, Redes NGN, otros.

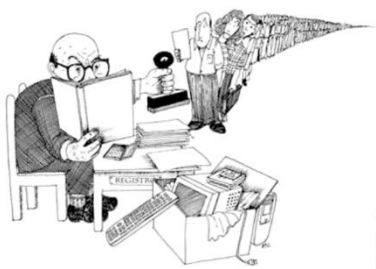
Etapas 3: Estudio de casos prácticos: Se estudió los casos de e-Gov de Estonia y Uruguay, por ser el primer país digital a nivel mundial y el primero a nivel América Latina respectivamente. También se investigó sobre el estado actual de e-Gov en Argentina.

Etapas 4: Análisis de toda la información.

Etapas 5: Diseño de la red escalable, definiendo requisitos e infraestructuras esenciales y esquema de Red propuesto.

Etapas 6: Redacción del Trabajo final.

Estas etapas fueron cumplimentadas, con el criterio de manejar las tecnologías actuales y las implementaciones más recientes de e-Gov en el mundo.



“ E-Gov es la utilización por parte de las agencias de gobierno de Tecnologías de Información y Comunicación (TICs) que tienen la habilidad de transformar relaciones con los ciudadanos, empresarios y otros brazos del gobierno ”.
(Banco Mundial)

CAPÍTULO 2: ELECTRONIC-GOVERNMENT

2.1 - Aproximación a la conceptualización de Electronic-Government

Según el Diccionario de Oxford el término **gobernar** se define como “ejercer la dirección, la administración y el control de un Estado, ciudad o colectividad;” el término **gobierno** se define como la “acción de gobernar”, y también como el “conjunto de personas y organismos que gobiernan o dirigen una división político-administrativa (estado, autonomía, provincia, municipio, departamento, etc)”.

El término **e-Gov**, también denominado **Electronic-Government** (en inglés), E-Government, o e-Government, tiene varias definiciones a nivel mundial, y están construidas sobre la base de las TICs y la Sociedad de la Información (SI):

- Según la Ley 27.078 de Argentina Digital, las TICs son “el conjunto de recursos, herramientas, equipos, programas informáticos, aplicaciones, redes y medios que permitan la compilación, procesamiento, almacenamiento y transmisión de información, como por ejemplo voz, datos, texto, video e imágenes, entre otros.
- La SI es aquella en la cual las tecnologías facilitan la creación, distribución y manipulación de la información y juegan un papel esencial en las actividades sociales, culturales y económicas”.

Según la **Government Publishing Office** de EEUU, en su acta de diciembre de 2002 [GPO, 2002] el término e-Gov hace referencia al “uso que hace el Gobierno de aplicaciones de Internet basadas en la web y otras tecnologías de la información, combinadas con procesos que implementa estas tecnologías a fin de

- mejorar el acceso y la entrega de información y servicios gubernamentales,
- llevar a cabo mejoras en las operaciones gubernamentales que pueden incluir efectividad, eficacia, calidad de servicio o transformación”.

Por su parte, la **Unión Europea** [EUR-Lex, 2003] caracteriza al e-Gov como “el uso de las TICs en las administraciones públicas, combinado con cambios organizativos y nuevas aptitudes, con el fin de mejorar los servicios públicos y los procesos democráticos y reforzar el apoyo a las políticas públicas”.

En la **IX Conferencia Iberoamericana de Ministros de Administración Pública y Reforma del Estado**, realizada en Pucón-Chile, el 31 de mayo y 1º de junio de 2007 [CLAD, 2007] se utilizan las expresiones de “**e-Gov**” y de “**Administración Electrónica**” como sinónimas. Y se define al e-Gov como “el uso de las TIC en los órganos de la Administración para mejorar la información y los servicios ofrecidos a los ciudadanos, orientar la eficacia y eficiencia de la gestión pública e incrementar sustantivamente la transparencia del sector público y la participación de los ciudadanos. Todo ello, sin perjuicio de las denominaciones establecidas en las legislaciones nacionales”.

Según la **Organización de los Estados Americanos** [OEA, 2018], el e-Gov es “la aplicación de las TICs al funcionamiento del sector público, con el objetivo de brindar mejores servicios al ciudadano e incrementar la eficiencia, la transparencia y la participación ciudadana”.

Para el **Banco Mundial**, el e-Gov se refiere al “uso de tecnologías de información por parte de las agencias gubernamentales que tienen la habilidad de transformar las relaciones entre los ciudadanos, los negocios y otros brazos del gobierno”.

La **Organización para la Cooperación y el Desarrollo Económicos** (OCDE) define al e-Gov como “el replanteamiento de la política gubernamental, el programa y la prestación de servicios a través de habilitadores digitales, como parte integral de las estrategias de modernización de los gobiernos, para crear valor público. El gobierno digital depende de un ecosistema que conecta a gobiernos, organizaciones no gubernamentales, empresas, asociaciones de ciudadanos e individuos para respaldar la producción y el acceso a datos, servicios y contenido a través de las interacciones con el gobierno” [Portal de Canadá, 2017].

Para el **Grupo Gartner**¹ [Gartnet, 2000] el e-Gov es “la innovación continua de los servicios, la participación de los ciudadanos y la forma de gobernar mediante la transformación de las relaciones externas e internas a través de la tecnología, Internet y las nuevas formas de comunicación”.

Las definiciones de e-Gov son diversas. Sin embargo, es posible identificar dos elementos comunes en todas ellas: el uso de las TICs y la gestión electrónica de servicios públicos. También es importante considerar las expresiones sinónimas que utilizan en diferentes partes del mundo para el nombrar el e-Gov, como por ejemplo **Administración Electrónica**, **Gobierno Digital** o **Gobierno en línea**. Así mismo se habla de **Gobierno Móvil** o **m-Government**, refiriéndose al uso de la tecnología inalámbrica y móvil. Se puede considerar como una extensión de la **Administración Electrónica**, por ofrecer aplicaciones móviles a tal fin.

En la actualidad se están imponiendo nuevas formas de vida y de trabajo que se sustentan en la información y en las relaciones de comunicación por medio de las TICs; estas nuevas concepciones se extienden cada vez más, avanzan sobre contextos gubernamentales y buscan facilitar la relación gobierno-ciudadano de un país, dando lugar al e-Gov.

2.2 - Inicios del e-Gov

La implementación del e-Gov implica grandes cambios culturales en una sociedad y una transformación profunda en la administración pública. Uno de los primeros antecedentes se dio en **Dinamarca**, en la década de los 70, cuando diversos municipios se organizaron para compartir información mediante sistemas informáticos. Luego esta experiencia novedosa fue replicada en varios países europeos.

En la década del 90, con el surgimiento de Internet, se implementó en **Estados Unidos** creando políticas gubernamentales para regir el desarrollo y la operación del e-Gov, en busca de transparencia y mejoras en las comunicaciones hacia la ciudadanía. Una de las primeras iniciativas de reforma gubernamental con inclusión de las TICs fueron la National Performance Review (NPR) y la The National Information Infrastructure (NII) en 1993.

¹ **Gartner Group** es una empresa consultora y de investigación de las Tecnologías de la Información con sede en Stamford, Connecticut, Estados Unidos. También tiene una sede en Fort Myers Florida y otras oficinas ubicadas en Reino Unido, Asia e Hispanoamérica. Después del año 2001 pasó a llamarse Gartner Inc.

Entre 1994 y 1995 se llevaron a cabo innumerables acciones para e-Gov. Se sumaron países como Canadá, Japón, Australia, Singapur. El G-7 (Estados Unidos, Japón, Canadá, Francia, Italia, Alemania y el Reino Unido) se reunió para discutir la propuesta de implementar la **Infraestructura de Información Global** (GII: Global Information Infrastructure, en inglés), sobre los principios de competencia abierta, estándares técnicos comunes e interoperabilidad de sistemas. En el Foro se manifestó el deseo de cooperación con entidades internacionales como la ONU² y de la OCDE.

En 2004, El Organismo de las Naciones Unidas publicó un estudio sobre los e-Gov's a nivel Mundial en su “**E-Government Readiness Report**” [Reporte UN, 2004]. En él se destacan los diez países con mayor desarrollo en esta área: Estados Unidos (0,9132), Dinamarca, Reino Unido, Suecia, Corea del Sur, Australia, Canadá, Singapur, Finlandia y Noruega (0,8178). De América Latina, solamente figura Chile con una puntuación de 0,6835, un índice bastante importante para ese tiempo.

A partir de todas esas iniciativas y como consecuencia de la utilización creciente de las TICs en los procesos gubernamentales surge el concepto de e-Gov.

2.3 - Modelos de interacción de e-Gov

Implementar un e-Gov no consiste solo en imponer nuevas tecnologías sobre los modelos de gobierno existentes, sino que incluye **un proceso sistemático y paulatino de adaptación del sector público al nuevo entorno**. Resulta sumamente importante definir la dinámica de interacción entre los diferentes grupos de usuarios de la administración pública: los ciudadanos, los negocios y el mismo gobierno [Ruelas y Arámburo, 2006; Rama Prasad y RaghavaRao, 2012]. Algunos modelos de interacción de e-Gov:

- G2C-Government to Citizens: Gobierno a ciudadanos, un modelo centrado en la entrega de información y servicios gubernamentales en ámbitos como educación, impuestos, seguridad social, elecciones, etc. Desde un mismo portal, los ciudadanos pueden realizar trámites que involucren agencias múltiples, sin necesidad del acceso físico a las oficinas respectivas ni pérdidas de tiempo en largas esperas.
- G2B-Government to Business: Gobierno a Negocios, centrado en la entrega de información y servicios gubernamentales a empresas, como acceso a información, subvenciones, obligaciones legales, pago de impuestos, pago de patentes, licitaciones públicas, etc.
- G2G-Government to Government: Gobierno a Gobierno, que representa los sistemas internos y procedimientos que conforman la estructura de las organizaciones públicas. Incluye transacciones electrónicas, compartir datos entre actores gubernamentales, interacciones intra e inter-ministerios, otros.

2.4 - Objetivos de un e-Gov

Un e-Gov persigue los siguientes objetivos [Rama Prasad y RaghavaRao, 2012]:

- Construir una sociedad informada: solo desde esta condición podrá tomar decisiones adecuadas para su país y su comunidad.
- Incrementar la interacción entre el gobierno y los ciudadanos: de tal manera que el gobierno pueda conocer la realidad concreta de cada ciudadano,
- gobernar con transparencia, con rendiciones de cuentas claras,
- reducir los grandes costos por la administración y tiempos administrativos, mejorando los tiempos de respuestas para el ciudadano.

² ONU: Organización de las Naciones Unidas

2.5 - Modelos de desarrollo de e-Gov

El desarrollo del e-Gov es un proceso evolutivo, en el cual se pueden distinguir diferentes fases independientes. Cada una de ellas tiene un objetivo distinto y distintas exigencias con respecto a los costos, necesidades cognitivas y nivel de uso de las TICs. Los gobiernos han trabajado para formular modelos de desarrollo que integren estrategia, personal, procesos y tecnología al igual que planes de acción efectivos para trazar una implementación exitosa.

Los modelos propuestos en la Tabla 2.1 tienen elementos en común. Por lo general las fases iniciales implican la publicación o la catalogación de información en la Web. Luego, los sistemas internos se conectan a interfaces en línea y así los ciudadanos se conectan electrónicamente con el gobierno e interactúan en diferentes niveles. En la etapa final de la evolución, desaparece la frontera de secciones gubernamentales distintas y el usuario verá al gobierno como una sola entidad.

Propuesto por	Fase 1	Fase 2	Fase 3	Fase 4	Fase 5	Fase 6
Gartner (2000)	Presencia	Interacción	Transacción	Transformación	----	----
ONU (2001)	Presencia Emergente	Presencia Mejorada	Presencia Interactiva	Presencia Transaccional	Integración Total	----
Deloitte Research (2001)	Publicación de Información /distribución	Transacciones oficiales bidireccionales	Portales multi propósitos	Personalización del portal	Agrupamiento de servicios comunes	Integración completa y Transformación Empresarial
Elmargamid & Melver	Comunicación en un solo sentido	Comunicación bidireccional	Transacciones Complejas	Integración a través de la administración gubernamental	----	----
Layne & Lee	Catálogo	Transacción	Integración Vertical	Integración Horizontal	----	----
Watson & Mundy	Iniciación	Infusión	Adecuación (customization)	----	----	----
Hiller, Belanger, Moon (2001/2)	Diseminación unidireccional	Comunicación bidireccional	Servicios y transacciones financieras	Integración vertical y horizontal	Participación Política	----
Siau and Yong (2005)	Presencia Web	Interacción	Transacción	Transformación	Democracia electrónica	----

Tabla 2.1: Modelos de desarrollo de e-Gov. Fuente: Reporte de las Naciones Unidas.

El criterio que guía el desarrollo de los portales gubernamentales es brindar un punto de acceso único donde los ciudadanos puedan configurarlos de acuerdo a características seleccionadas por ellos mismos. Con esto, un gobierno puede establecer relaciones de uno a uno con los ciudadanos así como canales de comunicación bidireccionales.

Modelo de Gartner

El grupo Gartner analiza la evolución del e-Gov y reconstruye un modelo de 4 fases [Gartner, 2000]:

- La primera fase es **Presencia**, que implica poner a disposición de los ciudadanos y de las empresas aquella información básica de manera rápida, directa y sin interacción utilizando Internet difundiendo los portales institucionales o sitios web.

- La segunda fase es **Interacción**, donde se abren canales para una comunicación más directa entre ciudadanos y organismo con la posibilidad de brindar información y recibir opiniones. Tales canales de comunicación pueden ser correo electrónico, envío de formularios, de comentarios o de foros.
- En la tercera fase, de **Transacción**, los sitios web o portales de los organismos están preparados para ofrecer trámites en línea (realización, seguimiento y finalización) y ofrecer servicios como un complemento de la atención frente a frente en las oficinas físicas. Por lo tanto, los gobiernos remodelan sus procedimientos de forma que a futuro el resultado implica un ahorro en costos, tiempo y mejora de la productividad.
- La cuarta fase es **Transformación**, donde se modifica la forma de actuar del gobierno en su estructura, su estilo de atención al público y el flujo de trámites y servicios que ahora se pueden realizar en forma electrónica por Internet (sitios web o portales integradores de organismos). Se incentiva y se promueve el uso de tecnologías para una comunicación activa entre los ciudadanos y los organismos de la que resulte la formación, discusión y evaluación de políticas públicas.

Fases que promueven la participación ciudadana en el e-Gov

Otros modelos incluyen una quinta y sexta fase que reciben distintos nombres: **Integración total**, **Administración relacional**, **Agrupamiento de servicios**, **Democracia electrónica** [Consortio del proyecto Atlántico, 2003], **Participación Democrática** [Cardona, 2002]. Estas fases representan el nivel *más sofisticado de las iniciativas en línea del gobierno*. A través de características interactivas tales como blogs, foros y otros, el gobierno solicita activamente opiniones y participación a los ciudadanos y los integra en el proceso interno de toma de decisiones. Estimula al ciudadano a que se involucre en la definición de las acciones gubernamentales [Tricas, 2007].

2.6 - Requisitos para la implementación del e-Gov

Para implementar un e-Gov se requieren cambios estructurales tanto en la administración como en la ciudadanía. Se requiere:

- Apropiaada infraestructura tecnológica y física
- Recursos humanos capacitados en nuevas tecnologías y herramientas administrativas
- Estabilidad política y económica, porque la implementación de un e-Gov es un proyecto a largo plazo
- Sistema estable, seguro, que genere confianza en la prestación de servicios
- Redes gubernamentales con diseño adecuado, intercomunicados y coordinados.
- Ciudadanía alfabetizada tecnológicamente, con fácil disponibilidad a recursos como internet y otras redes.

Según un artículo de la OCDE [OCDE, 2003], las cuatro principales barreras que impiden el desarrollo del gobierno electrónico son:

- Barreras legislativas y regulatorias
- Infraestructura tecnológica obsoleta
- Brecha digital
- Restricciones presupuestarias.

Si se tiene una visión gubernamental global, estratégica y sostenible, un diseño incremental con objetivos claros, resultados escalables, infraestructura adecuada, recursos humanos competentes y ciudadanía alfabetizada tecnológicamente se puede vislumbrar el éxito de un e-Gov.

2.7 - Beneficios del e-Gov

La implementación del e-Gov genera grandes beneficios, tanto para el gobierno como para la ciudadanía:

- Para los ciudadanos:
 - Mayor amplitud de canales, mejor disponibilidad de la información
 - Mejoras en la atención a través de servicios más ágiles, personalizados, con transparencia, en menor tiempo
 - Mayor participación ciudadana en la toma de decisiones
 - Reducción de costos operativos.
- Para las empresas:
 - Interacciones más rápidas
 - Gestiones transparentes
 - Disminución y/o eliminación de aspectos normativos
 - Disminución de costos operativos.
- Para las instituciones públicas:
 - Mayor rapidez, eficacia y eficiencia
 - Disminución de la burocratización en los procesos administrativos
 - Mejor utilización de inversiones tecnológicas
 - Uso eficiente de recursos compartidos (como información, infraestructura)
 - Auditabilidad de la gestión pública
 - Disminución de costos operativos.



*“Estonia es considerado el primer país digital del mundo: el 99% de los servicios públicos están disponibles en línea 24/7 (24 horas del día de los 7 días de la semana).
e-estonia.com*

CAPÍTULO 3: ESTADO ACTUAL DEL E-GOV EN EL MUNDO. CASOS DE ESTUDIO

3.1 - Estado actual del e-Gov en el mundo

Las Naciones Unidas han implementado una herramienta de evaluación del desarrollo del e-Gov aplicada a los Estados miembros³, denominada **Base de Datos de Desarrollo de Gobierno Electrónico de las Naciones Unidas (UNeGovDD)**. La misma ofrece una instantánea interactiva del desarrollo del e-Gov de cada país desde una perspectiva regional y global. Está construida con informes y encuestas elaborados por la **División de Administración Pública y Gestión del Desarrollo (DPADM)** del **Departamento de Asuntos Económicos y Sociales de las Naciones Unidas (DESA)**.

La Encuesta bianual evalúa la preparación y el nivel de participación electrónica en e-Gov de cada Estado miembro [UN-E-Government Knowledgebase, 2018] mediante dos índices:

- **EGDI: E-Government Development Index**, es decir, el **Índice de Desarrollo de Gobierno Electrónico**
- **EPI: Electronic Participation Index**, es decir, **Índice de Participación Electrónica**.

El EGDI

El **EGDI** es una medida compuesta por tres dimensiones importantes del e-Gov: Provisión de servicios en línea, Conectividad de telecomunicaciones y Capacidad humana. Matemáticamente, el EGDI es un promedio ponderado de índices de cada dimensión: Índice **OSI** (Online Service Index), Índice **TII** (Telecommunication Infrastructure Index) e Índice **HCI** (Human Capital Index).

Para obtener el EGDI se realizan estudios bianuales de la presencia en línea de los 193 Estados: se evalúa los sitios web nacionales y cómo se aplican las políticas y estrategias de e-Gov para la prestación de servicios esenciales. Los resultados se tabulan y combinan con un conjunto de indicadores que incorporan la capacidad de un país para participar en la sociedad de la información, sin los cuales los esfuerzos de desarrollo del e-Gov tienen un uso inmediato limitado. Este índice no está diseñado para capturar el desarrollo del gobierno electrónico en un sentido absoluto; más bien, su objetivo es dar una *calificación de desempeño de los gobiernos nacionales en relación con los demás*.

El EPI

Promover la participación electrónica de la ciudadanía es la piedra angular de un e-Gov socialmente inclusivo. Para ello se debe trabajar en la mejora del acceso a la información y los servicios públicos por parte de los ciudadanos; como también generar

³ Las Naciones Unidas reconocen 193 países soberanos (con autogobierno y completamente independientes).

canales de participación para la toma de decisiones públicas que afecte el bienestar de la sociedad, en general, y del individuo, en particular.

El **EPI** se deriva como un índice complementario de la Encuesta de e-Gov de las Naciones Unidas. Amplía la dimensión de la encuesta al centrarse en el **uso de los servicios en línea** para facilitar:

- el suministro de información por parte de los gobiernos a los ciudadanos,
- la interacción con las partes interesadas,
- y la participación en los procesos de toma de decisiones,

generando el siguiente marco de participación electrónica:

- **Información electrónica:** permitir la participación al proporcionar a los ciudadanos información pública y acceso a la información sin necesidad o bajo demanda.
- **Consulta electrónica:** participación de los ciudadanos en las contribuciones y deliberaciones sobre políticas y servicios públicos.
- **Toma de decisiones electrónicas:** empoderamiento de los ciudadanos a través del co-diseño de la opción de política y la coproducción de los componentes del servicio y las modalidades de entrega.

Ranking mundial según el EGDI

Según la encuesta de las Naciones Unidas sobre el e-Gov 2018, lanzada el 19 de julio de 2018, se pudo obtener información importante acerca de las tendencias en el desarrollo en países de todo el mundo (Fig. 3.1). En general, los países de América del Norte, Europa, Norte de Asia, Australia y Nueva Zelanda tienen un buen rendimiento del e-Gov.

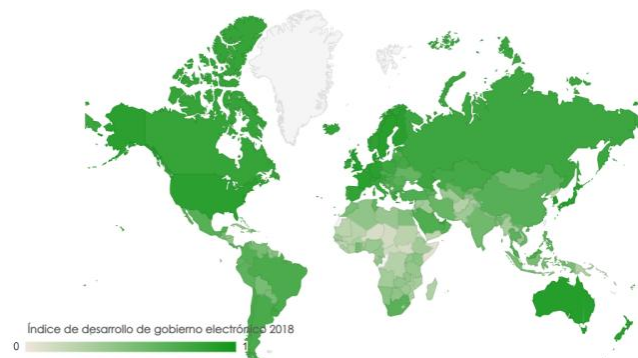


Figura 3.1: EGDI 2018. Fuente: UN-E-Government Knowledgebase, 2018.

Algunas conclusiones a nivel mundial:

- Los países europeos lideran el desarrollo del e-Gov con Dinamarca en el primer lugar.
- Los demás pertenecen a Oceanía y Asia (Tabla 3.1).
- En América y Asia el desarrollo de e-Gov es lento pero notable.
- En general existe una correlación positiva entre el nivel de ingreso del país y su clasificación de e-Gov.
- Existe una alta disparidad de desarrollo de e-Gov entre los países de África; lo mismo sucede con Oceanía. El 93% de los países de África y el 86% de Oceanía obtuvieron un puntaje EDGI inferior al promedio mundial de 0,55; los mismos tienen bajos ingresos y posiblemente enfrentan limitaciones para asignar recursos para el desarrollo de e-Gov.
- Por primera vez en 2018, el principal contribuyente de los puntajes de mejora de EDGI en todos los grupos es la **Provisión de servicios en línea (PSL)**.

- Todos los países tienen portales nacionales y sistemas de back-end para automatizar tareas administrativas básicas.
- El 73% proporcionan al menos un servicio transaccional en línea: pago de servicios públicos, presentación de impuestos de renta, registros de nuevos negocios.
- Aumentó la cantidad de países que ofrecen servicios mediante correo electrónicos, actualizaciones de fuentes SMS/RSS, aplicaciones móviles y formularios descargables.

Rango 2018	País	EGDI 2018
1	Dinamarca	0,9150
2	Australia	0,9053
3	República de Corea	0,9010
4	Reino Unido de Gran Bretaña e Irlanda del Norte	0,8999
5	Suecia	0,8882
6	Finlandia	0,8815
7	Singapur	0,8812
8	Nueva Zelanda	0,8806
9	Francia	0,8790
10	Japón	0,8783
11	Estados Unidos de América	0,8769
12	Alemania	0,8765
13	Países Bajos	0,8757
14	Noruega	0,8557
15	Suiza	0,8520

Tabla 3.1: Ranking de los 15 mejores EGDI 2018. Fuente: UN-E-Government Knowledgebase.

Para América, algunas conclusiones son:

- Estados Unidos y Canadá tienen los puntajes más altos.
- Uruguay es el país de América Latina con puntaje EGDI alto (Tabla 3.2).
- El 43% de los países⁴ tienen un EGDI promedio mundial superior a 0,55.
- Argentina ocupa el 43° lugar en el ranking mundial con un puntaje de 0,7335.

Rango 2018	País	EGDI 2018
11	Estados Unidos	0,8769
23	Canadá	0,8258
34	Uruguay	0,7858
42	Chile	0,7350
43	Argentina	0,7335
44	Brasil	0,7327

Tabla 3.2: Ranking de los 6 mejores EGDI 2018 de América. Fuente: UN-E-Government Knowledgebase.

⁴ América tiene 35 países

El significado preciso de estos valores varía de una edición de la Encuesta a la siguiente en la medida que se comprende el potencial de los cambios generados por el e-Gov. A nivel mundial, hay un progreso constante en la mejora de la prestación de servicios públicos y de e-Gov en línea.

3.2 - El Grupo D, de los Gobiernos Digitalmente más avanzados

El Grupo “D”, de los países digitales más avanzados, fue creado en 2014 y reúne a líderes en e-Gov. Inicialmente se denominó D5 y lo integraban Corea del Sur, Estonia, Israel, Nueva Zelanda y Reino Unido. En Febrero del 2018, pasó a denominarse D7 por la incorporación de Uruguay y Canadá. En Noviembre del mismo año pasó a denominarse D9 con la inclusión de México y Portugal [AGESIC-Novedades, 2018]. El objetivo del Grupo es compartir experiencias y aprendizajes, ayudarse mutuamente con la misión de lograr servicios más eficientes para la población de los países.

Durante la cumbre del D9 celebrada en Jerusalén, Uruguay asumió la presidencia del grupo y, por consiguiente, la organización de la cumbre del próximo año (Fig. 3.2).



Figura 3.2: El D9 en la cumbre del 21 y 22 de noviembre de 2018 celebrada en Jerusalén. Fuente: Portal AGESIC, 2018.

Principios del Desarrollo Digital

Los principios orientadores apuntan al desarrollo de la conectividad, la ciudadanía digital, la programación desde la niñez, el gobierno abierto, los estándares y códigos abiertos y el gobierno centrado en las personas. Se debe desarrollar el e-Gov atendiendo a:

- Necesidades del usuario: mediante un diseño de servicios públicos para el ciudadano.
- Estándares abiertos: La tecnología requiere interoperabilidad, por lo que se necesita un claro compromiso con una política de estándares abiertos libres.
- Código abierto: Los futuros sistemas gubernamentales, manuales y estándares se crean como código abierto y se pueden compartir entre los miembros.
- Mercados abiertos: En la contratación pública, crean una verdadera competencia para las empresas independientemente de su tamaño. Fomentar y apoyar una cultura de start-up y promover el crecimiento económico a través de mercados abiertos.
- Gobierno abierto (transparencia): Ser miembro de la Alianza para el Gobierno Abierto y usar licencias abiertas para producir y consumir datos abiertos.
- Conectividad: Habilitar una población en línea a través de una infraestructura digital integral y de alta calidad.
- Enseñar a los niños a programar: Compromiso de ofrecer a los niños la oportunidad de aprender a programar y desarrollar la próxima generación de habilidades.

- Asistencia digital: Un compromiso para ayudar a todos los ciudadanos a acceder a los servicios digitales.
- Compromiso de compartir y aprender: Los miembros se comprometen a trabajar juntos para ayudar a resolver los problemas de los demás.

3.3 - Caso de Estudio N° 1: el e-Gov de Estonia

Estonia, se encuentra ubicada al norte de Europa, con una población de 1.315.321 y un valor de ingreso per cápita de 18.190 dólares. Su EGDI es de 0,8486, por lo que ocupa el 16° lugar a nivel mundial. Su EPI es de 0.9101 ocupando el puesto N° 27 (Fig. 3.3). **Es el primer país que declara que el acceso a internet es un derecho humano.** Por ello el uso del Internet es completamente gratuito y con el mejor ancho de banda en todo su territorio.

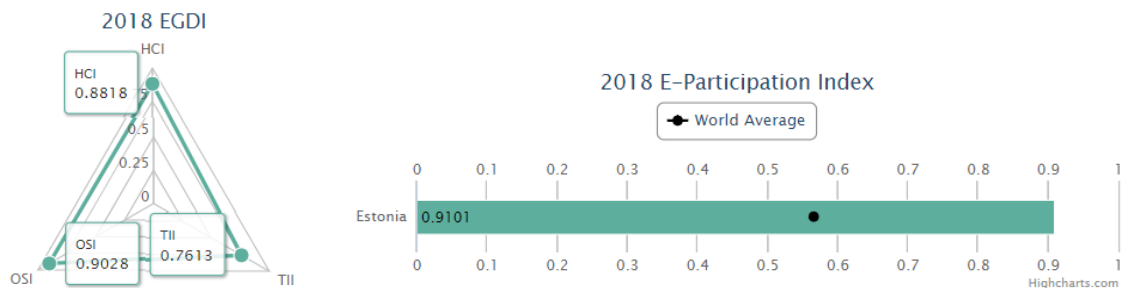


Figura 3.3: EGDI y EPI 2018 de Estonia. Fuente: UN-E-Government Knowledgebase, 2018.

En 1991 se independizó de la Unión Soviética, su economía estaba devastada, el nivel de pobreza era marginal y el nivel de desarrollo tecnológico era nulo. Después de casi 3 décadas, Estonia es considerado el primer país digital del mundo. **Su transformación fue una cuestión de supervivencia.** Los pilares fundamentales del desarrollo de su E-Gov fueron:

- La educación
- Internet como derecho humano, con mucha libertad
- La identificación digital
- La gestión pública como un servicio al ciudadano, con la máxima transparencia
- La ciberseguridad.

Algunos datos interesantes [El País, 2018]:

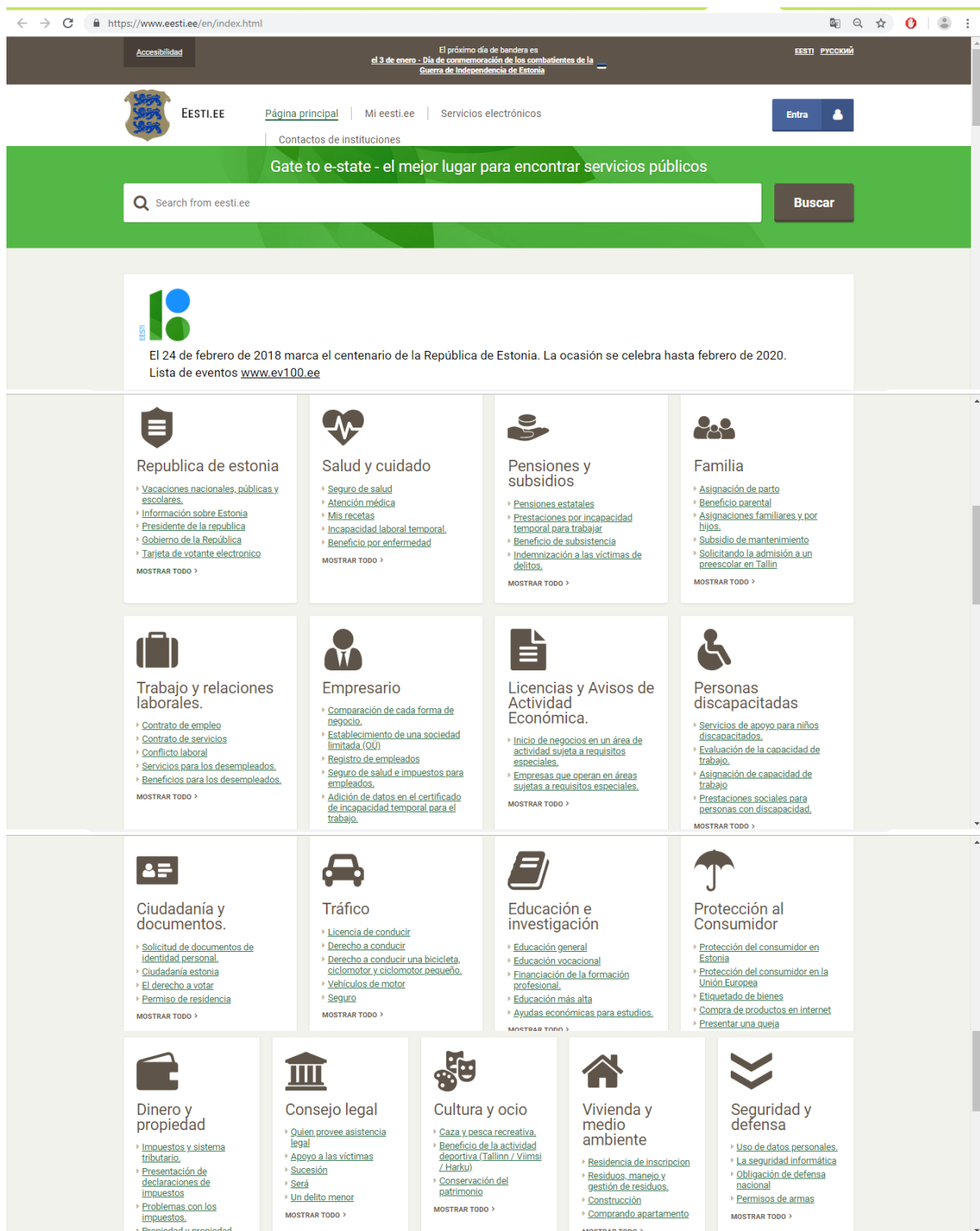
- El 88% de los estonios navega por Internet a diario.
- El 87% de la población entre los 16 y los 74 años se conecta al portal gubernamental.
- El 99% de los trámites oficiales están digitalizados, pueden realizarse en cualquier momento.
- Los ciudadanos tan solo necesitan una conexión a Internet para: Votar, renovar su licencia de conducir, consultar recetas médicas, hacer la declaración de la renta, impugnar una multa de tráfico, cambiar la dirección de su domicilio, registrar una empresa, firmar documentos, ver las notas de sus hijos y comunicarse con los profesores, acceder a su historial médico, etc.
- En el sistema estonio, **los ciudadanos son los únicos propietarios de sus datos 'online'**, consultarlos sin razón constituye un delito.
- Sus gobernantes predicán con el ejemplo: desde el año 2000 no usan papel en sus reuniones, utilizan firma digital en una pantalla para que las leyes entren en vigor.

En el paradigma digital de Estonia la tecnología está completamente entrelazada con la vida cotidiana [La Nación, 2018]. Se habla de e-gobierno, e-Estado, e-escuela, e-salud, e-voto, e-policía, e-tierra, e-ambulancia. De hecho, en Estonia los bebés ya nacen como e-

ciudadanos. La registraci3n se puede hacer desde cualquier computadora port3til, por los propios padres, usando sus respectivas identificaciones electr3nicas. Los ni1os comienza a programar desde primer grado, y tambi3n estudiar ciberseguridad y rob3tica. Complementan su formaci3n con arte, deportes y autodisciplina.

El Portal de e-Gov

El portal gubernamental est3 abierto las 24 horas de los siete d3as de la semana, ofrece 1789 tr3mites oficiales digitalizados (Figuras 3.4) desde 17 3reas (ventanas). Solamente 3 tr3mites necesitan de la presencia f3sica: casarse, divorciarse y operaciones inmobiliarias.



The screenshot shows the homepage of the EESTI.EE portal. At the top, there is a navigation bar with the EESTI.EE logo, a search bar, and links to 'Página principal', 'Mi eesti.ee', and 'Servicios electr3nicos'. Below the navigation bar, there is a green banner with the text 'Gate to e-state - el mejor lugar para encontrar servicios p3blicos' and a search bar. The main content area is divided into a grid of 17 categories, each with an icon and a list of services:

- Republica de estonia**: Vacaciones nacionales, p3blicas y escolares; Informaci3n sobre Estonia; Presidente de la rep3blica; Gobierno de la Rep3blica; Tarjeta de votante electr3nico.
- Salud y cuidado**: Seguro de salud; Atenci3n m3dica; Mis recetas; Incapacidad laboral temporal; Beneficio por enfermedad.
- Pensiones y subsidios**: Pensiones estatales; Prestaciones por incapacidad temporal para trabajar; Beneficio de subsistencia; Indemnizaci3n a las v3ctimas de delitos.
- Familia**: Asignaci3n de parto; Beneficio parental; Asignaciones familiares y por hijos; Subsidio de mantenimiento; Solicitando la admisi3n a un preescolar en Tallin.
- Trabajo y relaciones laborales**: Contrato de empleo; Contrato de servicios; Conflicto laboral; Servicios para los desempleados; Beneficios para los desempleados.
- Empresario**: Comparaci3n de cada forma de negocio; Establecimiento de una sociedad limitada (OÜ); Registro de empleados; Seguro de salud e impuestos para empleados; Adici3n de datos en el certificado de incapacidad temporal para el trabajo.
- Licencias y Avisos de Actividad Econ3mica**: Inicio de negocios en un 3rea de actividad sujeta a requisitos especiales; Empresas que operan en 3reas sujetas a requisitos especiales.
- Personas discapacitadas**: Servicios de apoyo para ni1os discapacitados; Evaluaci3n de la capacidad de trabajo; Asignaci3n de capacidad de trabajo; Prestaciones sociales para personas con discapacidad.
- Ciudadanía y documentos**: Solicitud de documentos de identidad personal; Ciudadanía estonia; El derecho a votar; Permiso de residencia.
- Tr3fico**: Licencia de conducir; Derecho a conducir; Derecho a conducir una bicicleta, ciclomotor y ciclomotor peque1o; Vehículos de motor; Seguro.
- Educaci3n e investigaci3n**: Educaci3n general; Educaci3n vocacional; Financiación de la formaci3n profesional; Educaci3n m3s alta; Ayudas econ3micas para estudios.
- Protecci3n al Consumidor**: Protecci3n del consumidor en Estonia; Protecci3n del consumidor en la Uni3n Europea; Etiquetado de bienes; Compra de productos en internet; Presentar una queja.
- Dinero y propiedad**: Impuestos y sistema tributario; Presentaci3n de declaraciones de impuestos; Problemas con los impuestos; Propiedad y propiedad.
- Consejo legal**: Quien provee asistencia legal; Apoyo a las v3ctimas; Sucesi3n; Ser3; Un delito menor.
- Cultura y ocio**: Caza y pesca recreativa; Beneficio de la actividad deportiva (Tallinn / Viimsi / Harku); Conservaci3n del patrimonio.
- Vivienda y medio ambiente**: Residencia de inscripci3n; Residuos, manejo y gesti3n de residuos; Construcci3n; Comprando apartamento.
- Seguridad y defensa**: Uso de datos personales; La seguridad inform3tica; Obligaci3n de defensa nacional; Permisos de armas.



Figura 3.4: Portal de e-Gov de Estonia.

3.4 - Aspectos tecnológicos del e-Gov de Estonia

La clave del desarrollo de “e-Estonia” o “Estonia Electrónica” ha sido:

- Desarrollo en pequeñas iteraciones incrementales (para periodos menores a un año)
- Uso de estándares abiertos
- Desarrollo abierto y transparencia de procesos
- Uso de software de código abierto.

El e-Gov de Estonia se desarrolla en colaboración entre el **Gobierno de Estonia**, representado por la **Fundación de Información y Comunicación del Estado (RIKS)**, y un consorcio de empresas del sector privado como **Cybernetica**, **Dell EMC**, **Ericsson**, **OpenNode** y **Telia**. Cada uno de ellos es responsable de áreas específicas:

- RIKS: centros de datos y gestión de las relaciones con los clientes
- Cybernetica: proceso de certificación ISKE
- Dell EMC: hardware
- Ericsson: gestión de proyectos, hardware y software;
- OpenNode: liderazgo técnico y organizador de nubes (portal de autoservicio y tienda de servicios);
- Telia: operaciones, productos y servicios relacionados con la nube.

RIKS fue creada a fines del año 2000. Se trata de una fundación no comercial administrada por el Ministerio de Asuntos Económicos y Comunicaciones que proporciona servicios relacionados con la comunicación para instituciones públicas y otras instituciones estatales presupuestadas y comunicación operativa de radio, marítima y telefónica.

Servicios de Interoperabilidad con Tecnología X-Road

X-Road es la columna vertebral de e-Estonia, permite que las diversas bases de datos de servicios electrónicos del sector público y privado de la nación se conecten y funcionen en forma coordinada.

El entorno de solución electrónica de Estonia incluye una gama completa de servicios para el público en general, y como cada servicio tiene sus propias bases de datos, todos usan X-Road. **Para garantizar transferencias seguras, todos los datos salientes de X-Road se firman y se cifran digitalmente, y todos los datos entrantes se autentican y se registran.**

X-Road se diseñó teniendo en cuenta el crecimiento, por lo que se puede ampliar a medida que los nuevos servicios electrónicos y las nuevas plataformas se ponen en línea.

Originalmente, esta tecnología se utilizaba solamente para enviar consultas a diferentes bases de datos. Ahora se ha convertido en una herramienta que también puede:

- escribir en múltiples bases de datos,
- transmitir grandes conjuntos de datos,
- y realizar búsquedas en varias bases de datos simultáneamente.

Hoy en día, X-Road también se implementa en Finlandia, Azerbaiyán, Namibia y las Islas Feroe. X-Road es también la primera plataforma de intercambio de datos en el mundo que permite el intercambio automático de datos entre países. Desde junio de 2017, se ha establecido la capacidad de intercambio automático de datos entre Estonia y Finlandia [e-Estonia, 2018]. Más de 900 organizaciones y empresas en Estonia utilizan X-Road diariamente.

Servicios de Seguridad y Encriptación con Tecnología KSI

Estonia ha desarrollado una amplia experiencia en esta seguridad cibernética, convirtiéndose en uno de los expertos internacionales en seguridad cibernética más reconocidos y valorados. Después de la experiencia de los ciberataques de 2007, el gobierno desarrolló la tecnología KSI, que actualmente es usada en más de 180 países.

KSI es una **tecnología de cadena de bloques escalable** para **garantizar** la integridad de los datos almacenados en los repositorios del gobierno y para **proteger** los datos contra amenazas internas; se garantiza que las redes, los sistemas y los datos estén libres de compromisos y mantengan su privacidad.

Un **blockchain** es un libro contable público distribuido, una base de datos con un conjunto de reglas predefinidas sobre cómo el libro contable se distribuye entre los participantes en el sistema. Debido a su propiedad ampliamente observada, la tecnología de cadena de bloques también hace **imposible cambiar los datos que ya se encuentran en la cadena de bloques**.

Con **KSI Blockchain** implementado en las redes del gobierno de Estonia, **nadie puede reescribir la historia y se puede probar matemáticamente la autenticidad de los datos electrónicos**. Significa que nadie, ni piratas informáticos, ni administradores de sistemas, ni siquiera el propio gobierno, puede manipular los datos [e-Estonia, 2018]. Estas soluciones son proporcionadas por las siguientes empresas e instituciones estonias:

- Guardtime KSI Blockchain, e-Law
- Sistema de información del tribunal de **Net Group**
- **Nortal** Data Protection, Data Governance
- **Roksnet** Private Blockchain
- **Helmes** Court Information System, Public e-File
- Sistema de información de la corte de **Aktors** (Kurdistán)
- **Centro de registros y sistemas de información** e-File, Public e-File, Court Information System
- **CGI Estonia** Digital File.

Estonia se convirtió en sede del Centro de Excelencia de Defensa Cibernética Cooperativa de la OTAN⁵ y de la agencia europea de TI. Los estonios son **especialistas mundiales en seguridad informática y exportan profesionales de esta disciplina a todo el mundo**. Respetando los protocolos diplomáticos, crearon una embajada de datos en Luxemburgo: un **backup de todo** fuera de sus fronteras, por si algo falla.

⁵ OTAN: Organización del Tratado del Atlántico Norte; en inglés: North Atlantic Treaty Organization (NATO)

3.5 - Caso de Estudio N° 2: el e-Gov de Uruguay

En el ranking mundial, Uruguay ocupa la posición 34°, con un EGDI de 0,7858 y un EPI de 0,9157 (Fig. 3.5). Actualmente lidera la presidencia del grupo D9.

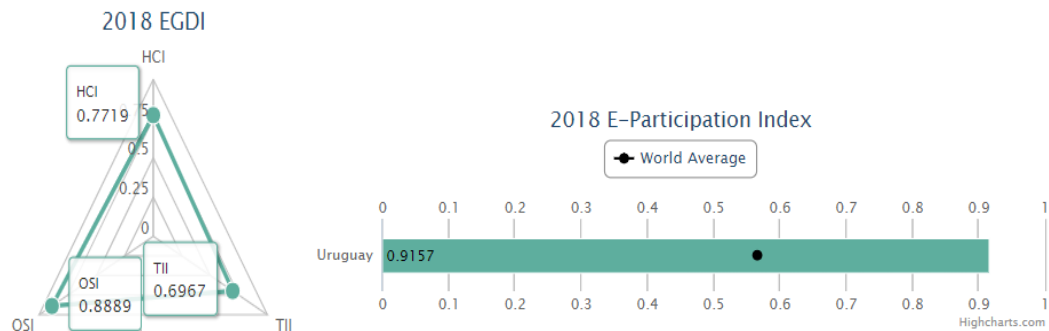


Figura 3.5: EGDI y EPI 2018 de Uruguay. Fuente: UN-E-Government Knowledgebase, 2018.

AGESIC es la **Agencia para el Desarrollo del Gobierno de Gestión Electrónica y la Sociedad de la Información y del Conocimiento**; es un organismo que depende de la Presidencia de la República y funciona con autonomía técnica. Su objetivo es **procurar la mejora de los servicios al ciudadano, utilizando las posibilidades que brindan las Tecnologías de la Información y las Comunicaciones** [AGESIC, 2018].

El Portal de e-Gov

Su **Plataforma de E-Gov (PGE)** fue creada por el artículo 72 de la Ley N° 17.930 de 19 de diciembre de 2005; proporciona el ambiente normativo legal y técnico, así como los instrumentos informáticos necesarios para facilitar la instalación de aplicaciones y servicios de e-Gov de Uruguay (Fig. 3.6).



Figura 3.6: Portal Web de Uruguay.

Se estima que para el 2020 el 100% de los trámites podrá ser online como se prevé en el Plan de Gobierno Digital 2020 [El País, 2018]. El mismo se estructura en cuatro pilares: políticas sociales e inclusión, desarrollo económico sustentable, gestión de gobierno y gobierno para la sociedad de la información.

3.6 - Arquitectura Integrada de Gobierno de Uruguay

La **Arquitectura Integrada de Gobierno** (AIG) está basada en TOGAF, un framework (Marco) de Arquitectura Empresarial que proporciona un enfoque para el diseño, planificación, implementación y gobierno de una arquitectura empresarial de gobierno [AGESIC, 2018].

Para el desarrollo de la **AIG de Uruguay** se adaptó dicho framework a las necesidades específicas del Estado, agregando los dominios de **Seguridad** y **Desempeño** como dominios transversales y adjuntando a los principios generales de la arquitectura el marco normativo asociado a interoperabilidad y tecnologías de información. Ese marco de referencia está compuesto por (Fig. 3.7):

- **Modelos de referencia:** Contienen los elementos mínimos para la definición de la arquitectura en un dominio en particular (negocio, datos, aplicaciones y Tecnología).
- **Repositorio:** Orientado favorecer el uso eficiente de las TICs, y poniendo a disponibilidad documentación de arquitecturas, componentes y soluciones.
- **Continuo de Arquitectura:** Define los procesos que apoyan a la mejora continua tanto de la arquitectura de gobierno, como su instanciación en las arquitecturas empresariales.

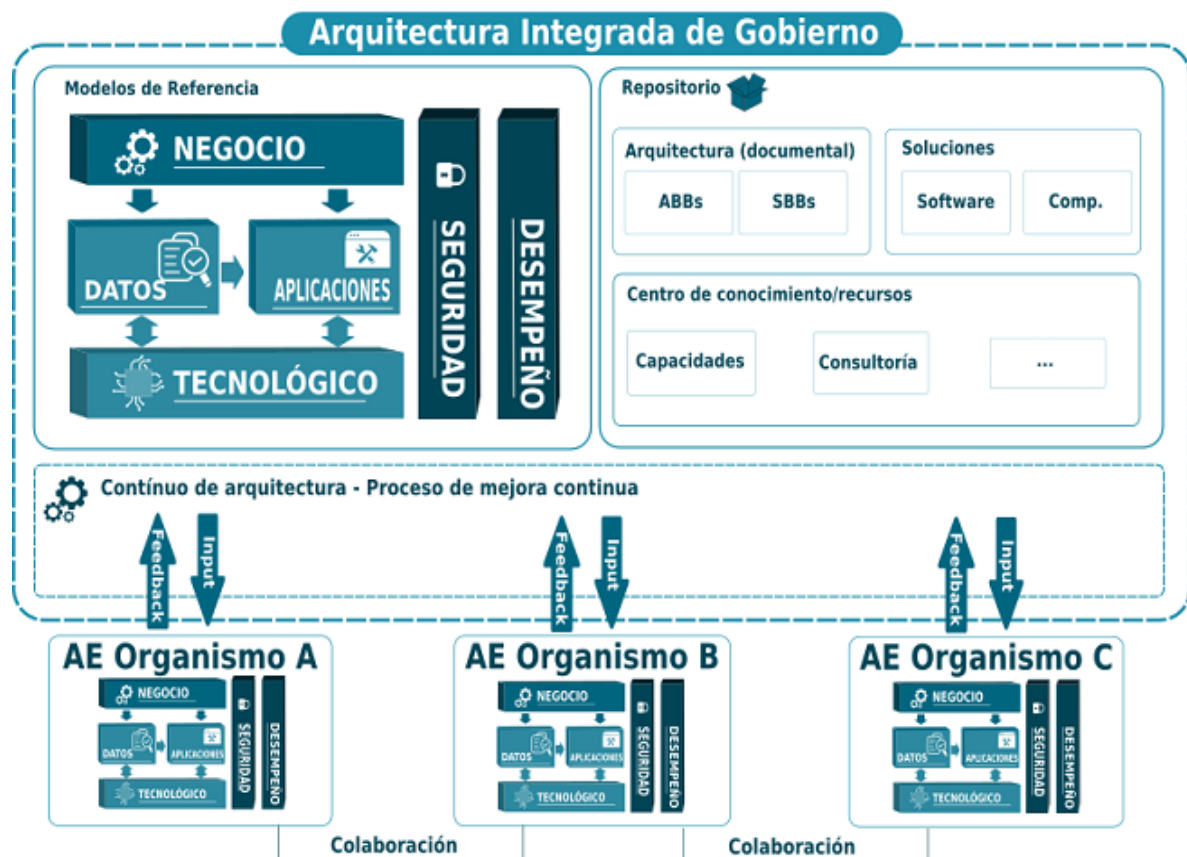


Figura 3.7: AIG de Uruguay basada en TOGAF. Fuente: AGESIC.

Por otro lado, la **AIG de Uruguay** sigue un **modelo federado** en el cual **cada organismo instancia el marco de referencia para la implementación de su arquitectura empresarial de gobierno**. Los procesos de mejora continua ayudan a llevar a adelante una retroalimentación entre la Arquitectura de Gobierno y las arquitecturas empresariales instanciadas por cada organización.

Los dominios desarrollados en la AIG de Uruguay son:

- Dominio de **Negocio**: describe una organización a través de la identificación de su misión, visión y principales productos y servicios de apoyo. Se trabaja con un Modelo de Referencia de Negocio (**MRN**), referenciada en la Fig. 3.8.
- Dominio de **Datos**: describe la estructura de los activos de datos físicos y lógicos, así como los recursos de gestión de datos. Se trabaja con un Modelo de Referencia de Datos (**MRD**), referenciada en la Fig. 3.9.
- Dominio de **Aplicaciones**: describe las aplicaciones de software que soportan los procesos, productos y servicios de negocio. Se trabaja con un Modelo de Referencia de Aplicaciones (**MRA**), referenciada en la Fig. 3.8.
- Dominio **Tecnológico**: describe la infraestructura tecnológica que da soporte a los componentes físicos y lógicos de datos y aplicaciones. Se trabaja con un Modelo de Referencia Tecnológico (**MRT**).
- Dominio de **Seguridad**: presenta un marco de ciberseguridad de referencia de mejores prácticas, organizado y con referencias a estándares internacionales. Se trabaja con un Modelo de Referencia de Seguridad (**MRS**), referenciada en la Fig. 3.8.
- Dominio de **Desempeño**: tiene por objetivo facilitar la medición en forma transversal de distintos elementos comprendidos en cada dominio de la arquitectura empresarial de la organización. Se trabaja con un Modelo de Referencia de Desempeño (**MRDe**), referenciada en la Fig. 3.10.

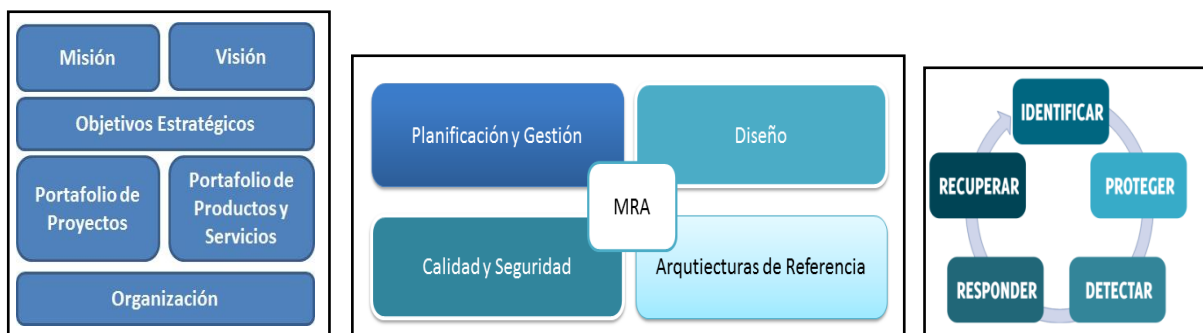


Figura 3.8: MRN, MRA y MRS de AIG de Uruguay. Fuente: AGESIC



Figura 3.9: MRD de AIG de Uruguay. Fuente: AGESIC.

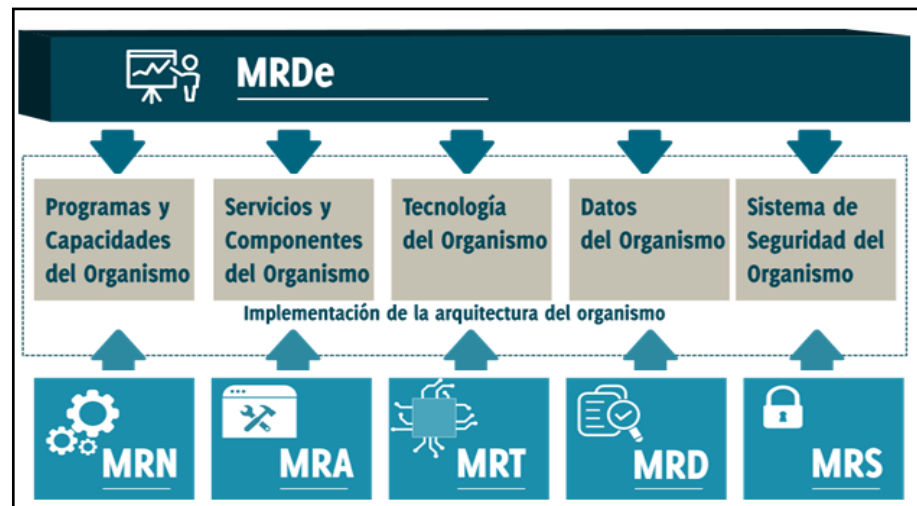


Figura 3.10: MRS y MRDe de AIG de Uruguay. Fuente: AGESIC.

Modelo de Referencia Tecnológico (MRT)

El MRT es una guía para la documentación y gestión de las arquitecturas desde el punto de vista del dominio de tecnología y está estructurada en varias capas (Fig. 3.11).

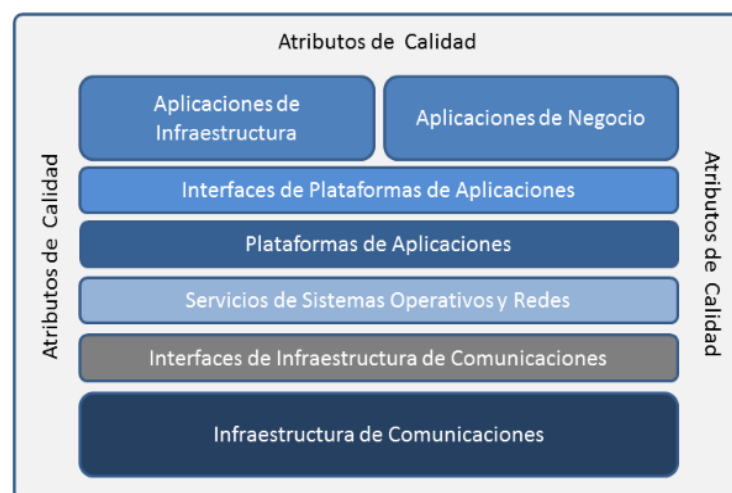


Figura 3.11: MRT de e-Gov de Uruguay. Fuente: AGESIC, 2018.

La capa de **Aplicaciones de Software** constituye un **input fundamental para el desarrollo de la arquitectura tecnológica al definir requerimientos para los servicios de infraestructura**, particularmente la base para los estándares a utilizar en las Plataformas de Aplicaciones (de infraestructura y de negocios). La capa de **Interfaces de Plataformas de Aplicaciones** define la interfaz de comunicación entre la **capa de aplicaciones** y las **plataformas de aplicaciones**: APIs, protocolos de comunicación, puntos de acceso, etc.

La capa **Plataformas de Aplicaciones** es conjunto de componentes tecnológicos de hardware y software que provee los servicios para soportar los requerimientos de la capa **Aplicaciones de Software**. Dichas plataformas pueden ser categorizadas como sigue:

- Gestión de datos. Ej.: diccionarios y repositorios de datos, administradores de bases de datos (DBMS), sistemas de archivos, etc.
- Intercambio de datos. Ej.: Plataforma de Interoperabilidad (PDI)
- Servicios de Ubicación y Directorio. Ej.: motor de búsqueda de contenidos de los portales de la organización
- Procesamiento Transaccional. Ej.: gestor transaccional de un sistema bancario.

- Servicios de Seguridad. Ej.: plataformas de autenticación centralizada, sistemas de control de acceso, sistemas de registro de auditorías, etc.
- Ingeniería de Software. Ej.: entornos de desarrollo, frameworks utilizados, generadores de interfaces gráficas, etc.
- Gestión de Sistemas y Redes. Ej.: sistemas para la gestión de usuarios, sistemas de monitoreo para la medición del desempeño, sistemas de respaldos, herramientas de gestión de redes, etc.

La capa de **Servicios de Sistemas Operativos y de Redes** brinda **apoyo** para el funcionamiento de las distintas plataformas y **soporte** para interoperabilidad y acceso a datos de aplicaciones distribuidas en ambientes heterogéneos, como ser:

- Comunicaciones. Ej.: transferencia de archivos, log-in remoto, ejecución de procesos remoto, etc.
- Notificaciones electrónicas. Ej.: a través de correo electrónico.
- Datos distribuidos. Ej.: servicios de acceso a bases de datos.
- Archivos distribuidos. Ej.: servicios de cache, bloqueo de archivos, etc.

La capa **infraestructura de comunicaciones** provee los servicios básicos para la interconexión de sistemas y provee mecanismos básicos para facilitar la transferencia de información. Contiene los elementos de hardware y software que componen la red y los enlaces físicos de comunicaciones utilizados por los sistemas.

3.7 - Tecnologías de la Plataforma de e-Gov de Uruguay

La PGE provee infraestructura (hardware y software) y servicios utilitarios, que **reducen la complejidad de implementar servicios al público y/o accesibles dentro del Estado**. Asimismo, **aporta los mecanismos técnicos** idóneos para implementar servicios compuestos, basados en los ofrecidos por diferentes organismos, normalizando e integrando la información proveniente de éstos. La PGE apunta a ser una **plataforma de integración**, promueve el desarrollo técnico de los organismos en forma coordinada, para que los avances institucionales contribuyan al desarrollo del e-Gov a nivel nacional (Fig. 3.12).

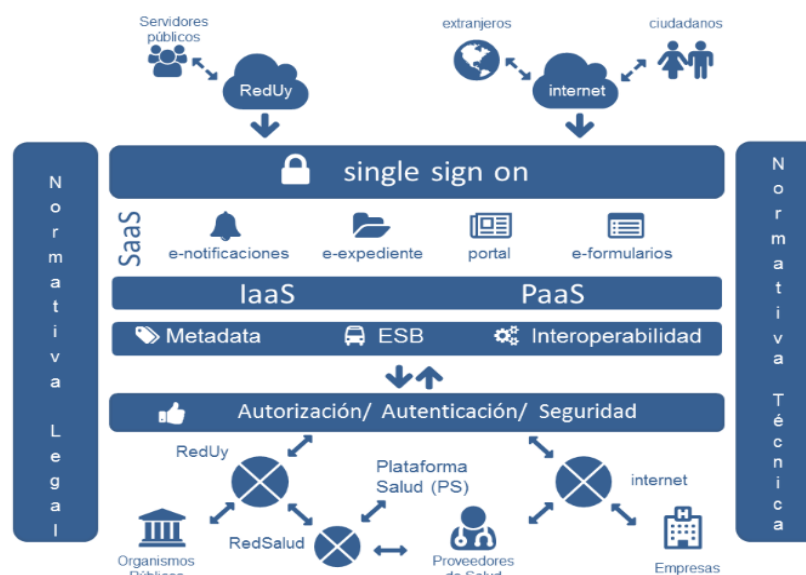


Figura 3.11: Esquema de integración de la PGE de Uruguay. Fuente: AGESIC, 2018.

A nivel tecnológico, la PGE **posibilita que los organismos provean sus funcionalidades de negocio a través de servicios de software de forma independiente a la plataforma en**

la que fueron implementados. Esto corresponde a la **implementación de una Arquitectura Orientada a Servicios (SOA) a nivel del Estado**, en la cual los servicios ofrecidos por los organismos son descriptos, publicados y descubiertos, invocados y combinados a través de interfaces y protocolos estandarizados. De esta forma, al facilitarse la **reutilización de servicios**, se promueve la construcción de nuevos servicios en base a otros ya existentes, reduciéndose los tiempos de implementación de nuevos requerimientos. Por otro lado, el “acoplamiento débil” entre servicios promovido por la SOA, permite la evolución autónoma de los servicios de software implementados en los organismos [AGESIC, 2018].

Plataforma de Interoperabilidad

La Plataforma de Interoperabilidad (PDI) forma parte de la PGE de AGESIC y tiene como objetivo general facilitar y promover la implementación de servicios de e-Gov en Uruguay. Para esto, la PDI brinda **mecanismos que apuntan a simplificar la integración entre los organismos del Estado** y a **posibilitar un mejor aprovechamiento de sus activos** (Fig. 3.12). Como consecuencia se reduce la cantidad de pasos que un ciudadano tiene que ejecutar al momento de realizar un trámite estatal en determinado Organismo. El objetivo final es no pedirle al ciudadano información que el Estado, como un todo, ya posee.

La PDI **brinda un marco legal y técnico a todas las transacciones que por allí pasan**. A su vez, **establece un único punto de acceso** hacia los servicios que son expuestos a través de ella, y todas las comunicaciones son a través REDuy.

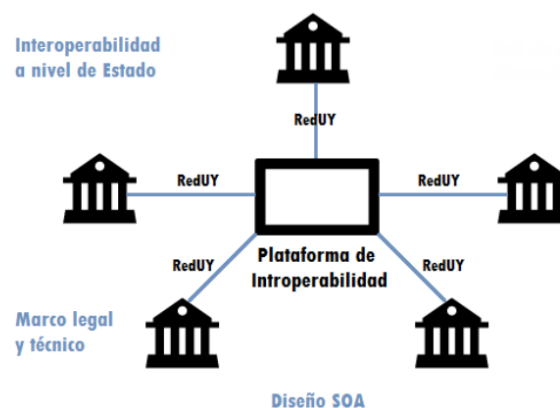


Figura 3.12: PDI. Fuente: AGESIC, 2018.

Tecnología SOA como base de la PDI

A nivel tecnológico, la PDI implementa SOA. La reutilización de servicios estatales lleva a una reducción de tiempos de desarrollo, costos, y duplicidad de la información, entre otros. Tres grandes sistemas son los que conforman la PDI:

- **Sistema de control de acceso:** formado por varios productos para autenticación y autorización del consumo de servicios basados en XML, y es la puerta de entrada a la Plataforma.
- **Sistema de gestión de metadatos:** provee una especificación de alto nivel de los conceptos relativos a servicios públicos, de forma de evitar o resolver ambigüedades en el manejo de estos conceptos por parte de los organismos.
- **Middleware:** cuenta con mecanismos para facilitar el desarrollo, despliegue e integración de servicios y aplicaciones. Los organismos pueden utilizar esta plataforma para publicar y descubrir servicios, así como utilizar las diferentes capacidades de mediación, las cuales permiten desacoplar clientes y servicios.

El Middleware (herramienta de intermediación) está integrado por dos **buses de servicios Empresarial** (ESB-Enterprise Service Bus): uno de tecnología **Microsoft**, otro de tecnología **Java**, con el fin de obtener lo mejor de ambos y ampliar el espectro de posibilidades en cuanto a los métodos de conexión.

Tecnología REDuy en la PDI

La REDuy es una red de alta velocidad de fibra óptica, diseñada y desplegada como una Intranet a nivel país. La **arquitectura de la red** genera una malla sin la necesidad de tener enlaces punto a punto. Es decir, cada organismo está interconectado con todos los demás solo con un enlace, con ancho de banda de 10 Mbps o 100Mbps (Fig. 3.14).

Se trata de una red privada y segura, que cuenta con la supervisión y el control expresos del CERTuy (Centro de Respuesta a Incidentes de Seguridad Informática del Uruguay) de AGESIC. El tráfico desde los organismos hacia la REDuy está permitido libremente. Sin embargo, el tráfico hacia los organismos desde la REDuy, debe ser habilitado en los firewalls por el equipo de soporte de AGESIC.

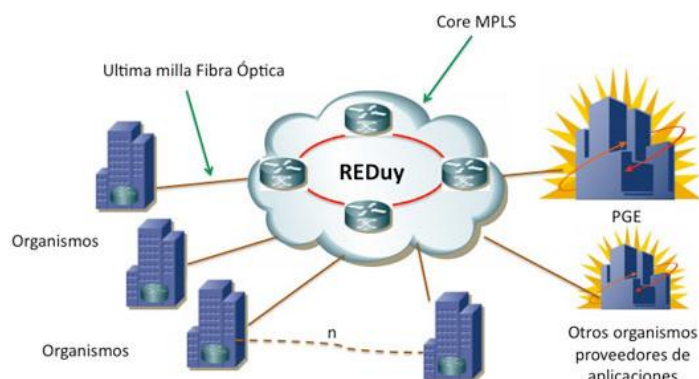


Figura 3.14: REDuy. Fuente: AGESIC, 2018.

En este contexto, la REDuy se convierte en el **instrumento clave para la obtención de un Gobierno en Red**. La infraestructura de conectividad de la misma posibilita que los organismos trabajen de forma integrada, bajo un marco técnico seguro para el intercambio de información. Además, permite racionalizar los recursos humanos y económicos.

E-Gov de Uruguay en la Nube: IaaS y PaaS

En función del crecimiento de los últimos años y acompañando las futuras necesidades, se trabajó en la incorporación de **Infraestructura hiperconvergente** (Fig. 3.15). En la PGE de Uruguay se incorporó **IaaS** y **PaaS**.



Figura 3.15: Tecnologías hiperconvergentes.

La **Plataforma como servicio (PaaS)** tiene la finalidad de **ampliar los servicios en la Nube** y **ofrecer ambientes que faciliten los tiempos de implantación**. La **Infraestructura como servicio (IaaS)** tiene la finalidad de **optimizar la gestión**, mejorando la calidad de los servicios en base a la aplicación de estándares exigentes de la industria, alojándose en un

datacenter diseñado específicamente para brindar estos servicios y respaldado en normas de organismos internacionales como **ISO**⁶, **TIA**⁷, **BICSI**⁸, **ICREA**⁹ entre otros.

3.8 - Caso de Estudio N° 3: El e-Gov en Argentina

Según la encuesta mundial realizada por las Naciones Unidas en julio del 2018, Argentina obtuvo un EGDI es de 0,7335, por lo que ocupa el 43° lugar a nivel mundial. Su EPI es de 0.6236 ocupando el puesto N° 87 (Fig. 3.16). Tiene una población de aproximadamente 44.000.000 habitantes y un valor de ingreso per cápita de 13.040 dólares (Según el Banco Mundial).

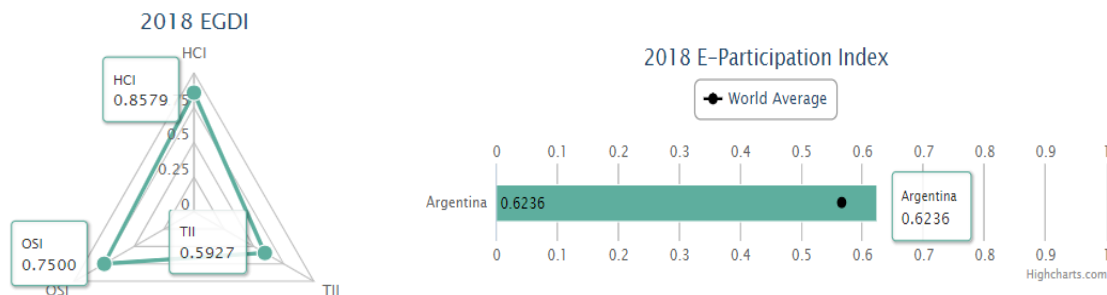


Figura 3.16: EGDI y EPI 2018 de Argentina. Fuente: UN-E-Government Knowledgebase, 2018.

Sus inicios y evolución a la fecha

En Argentina, se comenzó a implementar el e-Gov sólo en Capital Federal a partir del año 2009, en donde se ha pasado de una burocracia centrada en el papel a tener la totalidad de los trámites operativos en forma digital [Diario DPI, 2016].

Hacia el Año 2016, por Decreto Nacional 434/2016 se puso en marcha el **Plan de Modernización del Estado**, mediante el cual se definieron los ejes centrales, las prioridades y los fundamentos para promover las acciones necesarias orientadas a convertir al Estado en el principal garante del bien común [Boletín Oficial, 2016]. Se **incorporan las TICs** para simplificar procedimientos, propiciar reingenierías de procesos y ofrecer al ciudadano la posibilidad de mejorar el acceso por medios electrónicos a información personalizada, coherente e integral. Los ejes propuestos son:

- Plan de Tecnología y Gobierno Digital
- Gestión Integral de los Recursos Humanos
- Gestión por Resultados y Compromisos Públicos
- Gobierno Abierto e Innovación Pública
- Estrategia País Digital.

En agosto del año 2018, se puso en vigencia el Decreto Nacional 733/2018 por el cual se busca la **migración a soporte digital**, desterrando por completo el papel de la administración pública y permitiendo que todas las interacciones con el Estado sean de manera digital, más rápidas y más simples. Se busca evitar que al momento de realizar un trámite en un organismo del Estado o una empresa de servicios públicos, los ciudadanos tengan que presentar

⁶ ISO: International Organization for Standardization: Organización Internacional de Normalización

⁷ TIA: Telecommunications Industry Association: Asociación de la Industria de Telecomunicaciones

⁸ BICSI: Building Industry Consulting Service International: Servicio Internacional de Consultoría de la Industria de la Construcción

⁹ ICREA: International Computer Room Experts Association: Asociación Internacional de Expertos en Sala de Computación

información en papel con datos que son conocidos y que fueron generados por el propio Estado, es decir ya están digitalizados. Los objetivos son:

- Contar con un marco de interoperabilidad que permita el intercambio directo entre organismos de documentación y datos.
- Lograr la interconexión y operación simultánea.
- Permitir la consulta automática entre bases de datos informáticas y aprovechar las iniciativas ya desarrolladas para evitar que el ciudadano tenga que aportar información ya obrante en la Administración.

El decreto 733/2018 establece que se debe brindar a los ciudadanos una **herramienta gratuita de firma digital**. También indica que los documentos, comunicaciones, expedientes, actuaciones y procedimientos en general, deberán instrumentarse en el sistema de **Gestión Documental Electrónica (GDE)**, el cual está funcionando en todos los ministerios nacionales y 160 organismos públicos [Infobae, 2018]. En el caso de la Administración Nacional y los organismos descentralizados, esta digitalización comenzó el 15 de agosto de este año (2018), mientras que para las empresas y sociedades del Estado será desde enero de 2019.

El portal de e-Gov de Argentina

La plataforma e-Gov de Argentina permite el acceso a trámites básicos (Fig. 3.17). Actualmente cada provincia y la gran mayoría de los municipios disponen de su propio portal habilitando servicios básicos para el ciudadano, dentro del ámbito municipal o provincial.

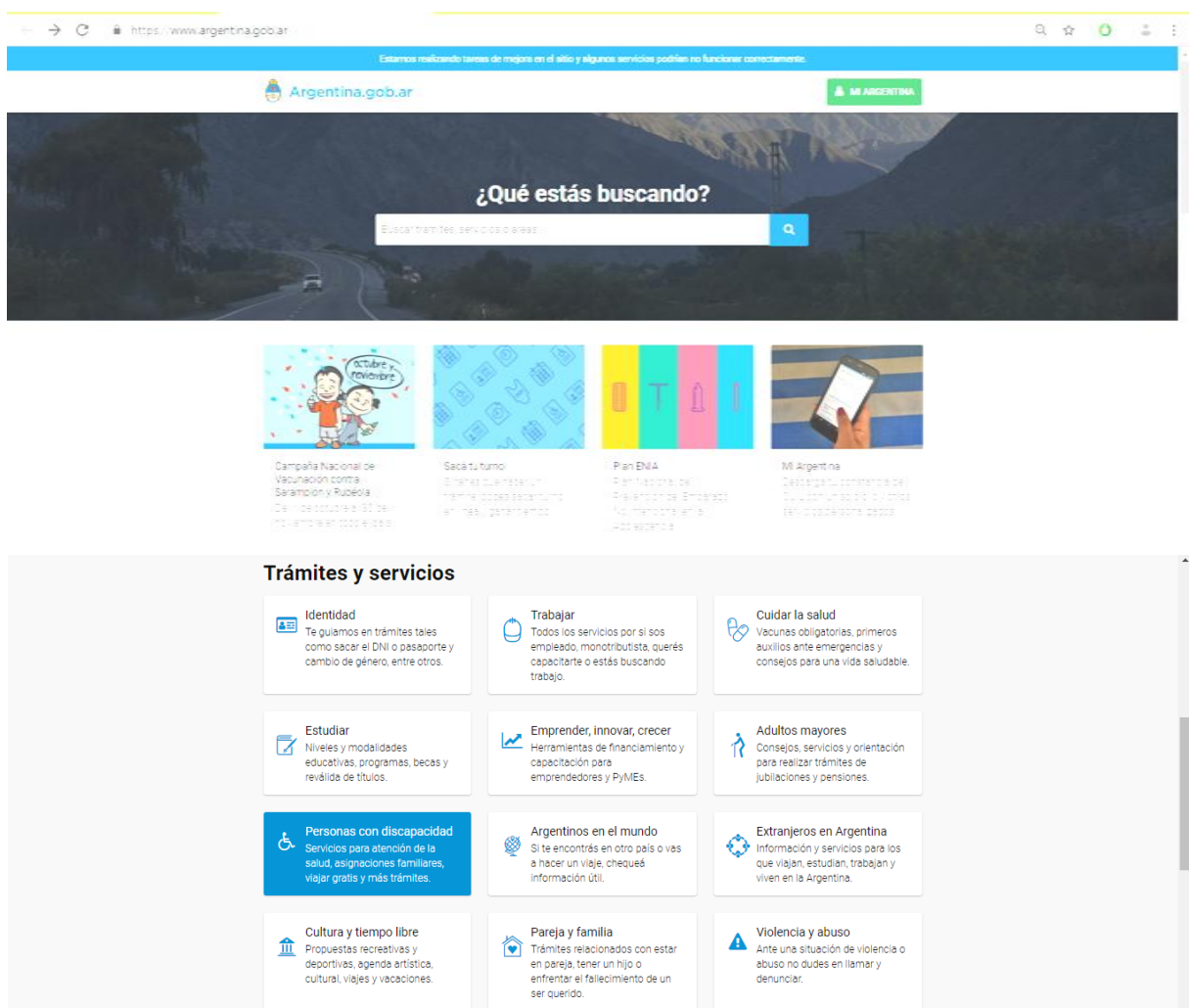




Figura 3.17: Plataforma e-Gov de Argentina.

Mediciones de e-Gov de Argentina

El **Índice de Servicios y Asistencia Gubernamental en Línea (ISAGEL)** es un indicador compuesto que mide el rendimiento de las administraciones públicas en materia de e-Gov a través de sus páginas web (Fig. 3.18). Con un rango de variación entre 0 y 1, el ISAGEL permite comparaciones de rendimientos tal que a mayor valor, mejor el **rendimiento relativo**.

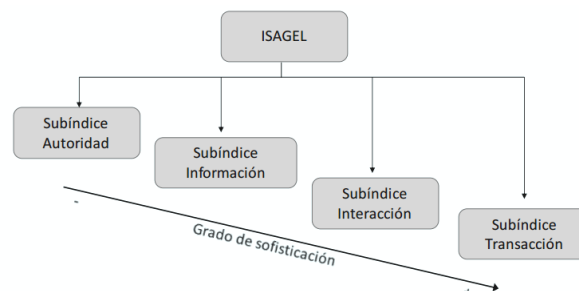


Figura 3.18: Componentes de ISAGEL. Fuente: Informe Económico N° 387 de IARAF.

En el año 2016, el **Instituto Argentino de Análisis Fiscal (IARAF)** realizó mediciones sobre el rendimiento de e-Gov en la Argentina. Sus resultados quedaron plasmados en el Informe Económico 387 [IARAF, 2016]. Se tuvo en cuenta las 24 jurisdicciones. En el eje de las abscisas se indica las jurisdicciones en estudio, el puesto que ocupa cada una en el ranking ISAGEL 2016, el puesto alcanzado en el ISAGEL 2015. Además se informa la “categoría de performance: CP” alcanzada por cada administración pública, definida según el grado de avance en materia de e-Gov alcanzado. Dichas categorías son:

- Incipiente: puntuaciones inferiores a 0,2
- Baja: puntuaciones en el intervalo 0,2 - 0,39
- Media: puntuaciones en el intervalo 0,4 - 0,59
- Alta: puntuaciones en el intervalo 0,6 - 0,79
- Superior: puntuaciones mayores o iguales a 0,8.

Los resultados obtenidos son (Fig. 3.19):

- El 71% de las provincias se concentran en la **categoría CP Baja** de e-Gov, con una media de 0,29 puntos. Las mejores de este grupo son Mendoza, Entre Ríos, Tucumán y La Pampa con puntajes entre 0,36 y 0,39. Las peores son Corrientes, Tierra de Fuego, Jujuy y San Luis, con puntuaciones alrededor de 0,21.
- Las mejores implementaciones de e-Gov corresponden a Ciudad Autónoma de Buenos Aires y de las provincias de Córdoba, Buenos Aires y Santa Fe. Sin embargo,

pertenecen a la **categoría CP Media**, lo que indica que aún queda **mucho por avanzar** en materia de e-Gov.

- La provincia de Salta, se encontraba en el 5° lugar en el año 2015, y se desplazó al 11° lugar en el año 2016. Su CP es de 0,31.

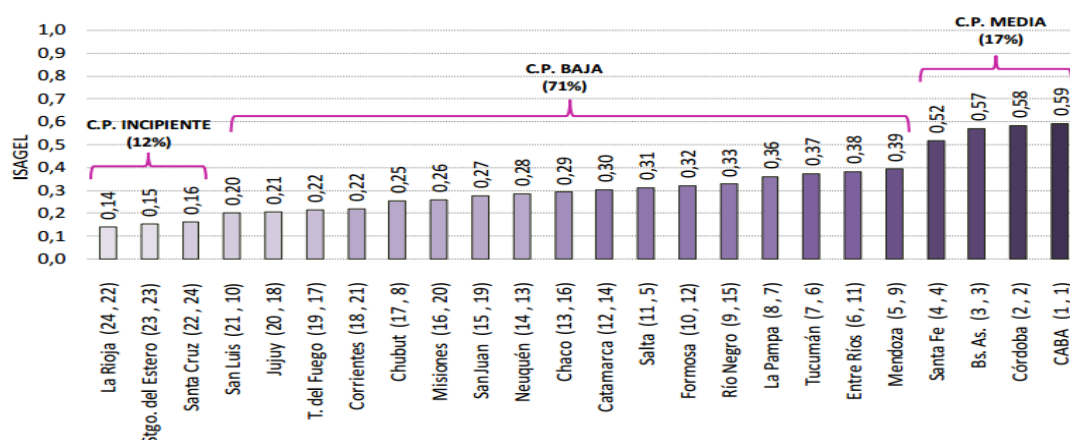


Figura 3.19: Ranking ISAGEL 2016. Estimación periodo julio-agosto 2016. Fuente: IARAF.

En la Tabla 3.3 se detallan datos del CP de algunas provincias: para la **provincia de Salta** el ítem **Autoridad** tiene una CP Alta, el ítem **Información** tiene una CP Media, el ítem **Interacción** responde a una CP Baja y el ítem **Transacción** tiene una CP Incipiente.

Jurisdicción	S.Autoridad	S.Información	S.Interacción	S.Transacción
Córdoba	Media	Alta	Alta	Media
CABA	Superior	Alta	Media	Media
Tucumán	Baja	Media	Media	Incipiente
Salta	Alta	Media	Baja	Incipiente
Jujuy	Media	Baja	Baja	Incipiente
Sgo del Estero	Incipiente	Baja	Incipiente	Incipiente

Tabla 3.3: Performance jurisdiccional según subíndice y jurisdicción. Fuente: IARAF.

En conclusión, los motivos principales de estos resultados se relacionan con una **pobre utilización de las TICs para trámites gubernamentales** y un **escaso uso de las redes sociales digitales con finalidad e-Gov**.

3.9 - Las redes sociales digitales y el e-Gov

Así mismo IARAF hace un análisis sobre las redes sociales digitales y su uso en e-Gov. Entendidas como un tipo de innovación tecnológica y de comportamiento, las **redes sociales digitales** son concebidas como **TICs tecnológicamente disruptivas**, en el sentido que cambiaron drásticamente la concepción y materialización de las relaciones entre las personas y entre éstas y las instituciones con las que conviven. *Su uso responde a una adecuación a nuevas realidades sociales*. La utilización de redes sociales para e-Gov implica:

- reconocer a los ciudadanos como potenciales agentes activos en la producción de contenidos gubernamentales en la web,
- aceptar la idea de que la innovación gubernamental también se encuentra fuera de los límites formales de las administraciones públicas,
- buscar satisfacer la creciente demanda ciudadana a la desintermediación de actividades gobierno-ciudadanía.

En conjunto, estas tres ideas caracterizan un nuevo perfil de gobierno: promotor de la colaboración, la transparencia, la participación e, incluso, la rendición de cuentas.

Uso de las Redes sociales digitales a nivel mundial y en la Argentina

A nivel mundial, **Twitter** se consolida como la red social con el mayor nivel de difusión dentro de las Administraciones públicas. Se aprovecha su potencialidad para enviar mensajes instantáneos a la ciudadanía *con anuncios de eventos y recibir una retroalimentación* casi inmediata. **Facebook** es la segunda red social más frecuentemente utilizada por los gobiernos. Se aprovecha para la **proyección de una imagen comunicativa** más que a la generación de una red de participación y colaboración. Por ej.: la turística y la cultural. Dentro de las plataformas multimedia, **Youtube** es la plataforma que probablemente ha adquirido la mayor relevancia. Se da un uso exhaustivo en los portales de prensa o noticias de las gobernaciones, así como en las áreas de gobierno vinculadas a cultura, deporte y educación.

En el caso de la Argentina, el IARAF relevó la utilización de una plataforma digital propia en 19 áreas de gobernación. Según su **orientación** (Fig. 3.20) se puede apreciar:

- el 72% de los portales de redes sociales está orientado a las **relaciones personales**, con mayor uso de Twitter (33%) y Facebook (37%)
- el resto se trata de portales orientados al **broadcasting**, como es el caso de Youtube (20%).

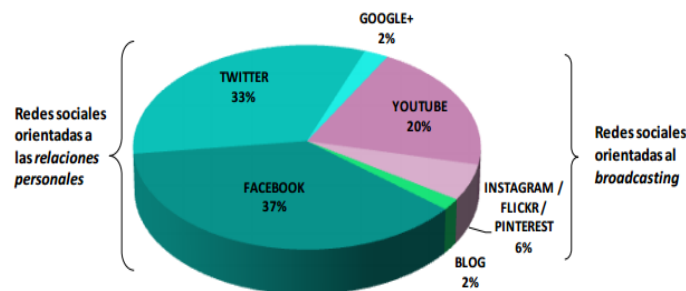


Figura 3.20: Distribución de portales web gubernamentales de redes sociales según orientación. Fuente: Informe Económico 387 de IARAF.

Con **respecto al área de gobierno** a la que estas plataformas sociales están vinculadas, sobresale su utilización por parte de las áreas de **Turismo y gastronomía (17%)**, **Educación (11%)** y **Gobernación (14%)** directamente vinculada a los portales web de Inicio de cada gobierno. Estas tres áreas utilizan los repositorios multimedia como Youtube e Instagram (Fig. 3.21).

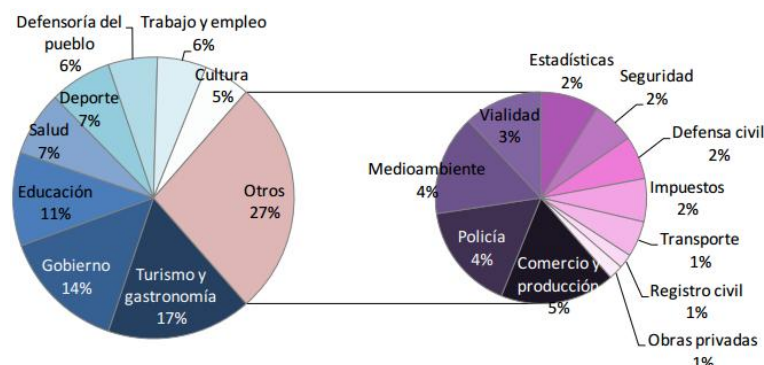


Figura 3.21: Distribución de portales web gubernamentales de redes sociales según Área de Gobernación. Fuente: Informe Económico 387 de IARAF.

Las áreas de **Salud**, **Deporte**, **Defensoría del Pueblo**, **Trabajo y empleo** y **Cultura** constituyen el 31% de los portales web de redes sociales observados, de las cuales las dos primeras se destacan con fuerte presencia en **Facebook**, **Twitter** y **Youtube**. El área **Deporte**,

además se caracteriza por la utilización del **Blog** como único canal de comunicación entre el área de gobernación (Ministerio/Secretaría) y los ciudadanos.

Las áreas menos usadas son **Obras privadas (1%)** y **Registro civil (1%)**, que levemente incluyen Facebook y Twitter.

En conclusión, hay una concentración de oferta de e-Gov solamente para los ítems de Autoridad e información, descuidando los otros aspectos necesarios para la transparencia de la gestión pública y el empoderamiento de los ciudadanos. El uso de las redes sociales digitales, concebidas como TICs multifuncionales, es precario en el ámbito de e-Gov.



“Los Servicios Web son la revolución informática de la nueva generación de aplicaciones que trabajan colaborativamente y en las cuales el software está distribuido en diferentes servidores”

CAPÍTULO 4: ARQUITECTURA DE SERVICIOS WEB

4.1 - Definiendo Servicio Web

La informática tuvo sus comienzos con aplicaciones **monousuarios**, luego surgieron las aplicaciones **multiusuarios**. Finalmente con el correr de los años surgió la **arquitectura cliente-servidor**, con lo que la **aplicación se dividía en dos**: una parte que interactuaba con el usuario, y otra parte destinada al **procesamiento de la información**. Con esta última configuración era posible que cada parte residiera en computadoras distintas. En la actualidad se habla de **aplicaciones distribuidas**, cuyos procesos se ejecutan en diferentes procesadores (CPUs). Los servicios web significan un paso más: la **computadora** ya no es considerada como un centro de procesamiento, sino **como un repositorio de servicios de un gran número de aplicaciones distribuidas** en diferentes lugares geográficos [Gil Aros, 2009], con hardware y software heterogéneo.

Entre las principales organizaciones que desarrollan estándares para servicios web se encuentran las siguientes:

- W3C - World Wide Web Consortium: es la principal organización desarrolladora de estándares para servicios web. Está conformada por aproximadamente 500 miembros.
- OASIS - Organization of the Advancement Structured Information Standards: Organización de Avances de estándares de la información estructurada. Estos estándares trabajan con XML¹⁰ y está conformada por 400 miembros que permanentemente mejoran el XML.
- IETF - Internet Engineering Task Force: Trabaja sobre las arquitecturas y tecnologías que corren y son ejecutadas por Internet.
- ISO - International Organization for Standardization: es la encargada de desarrollar los estándares para mejoramiento del comercio internacional. Está conformada por 140 países miembros.

Un **servicio web** es un conjunto de protocolos y estándares que sirven para el intercambio de datos entre aplicaciones. Distintas aplicaciones de software, desarrolladas en diferentes lenguajes de programación, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de computadoras como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos [Cruz del Valle, 2017; Gil Aros, 2009].

Un **servicio web** es un sistema de software diseñado para soportar interoperabilidad máquina a máquina a través de una red de interacción. Tiene una interfaz descrita en un formato procesable-máquina (específicamente WSDL¹¹).

¹⁰ XML: eXtensible Markup Language: Lenguaje Marcado Extensible

¹¹ WSDL: Web Services Description Language: Lenguaje de Descripción de Servicios Web

Un **Servicio Web** es un servicio de negocio programado en algún lenguaje, que envuelve cierta funcionalidad y la expone en una red de manera estandarizada, para que cualquiera conectado a esta misma red, que además soporte dichos estándares, pueda acceder a ella.

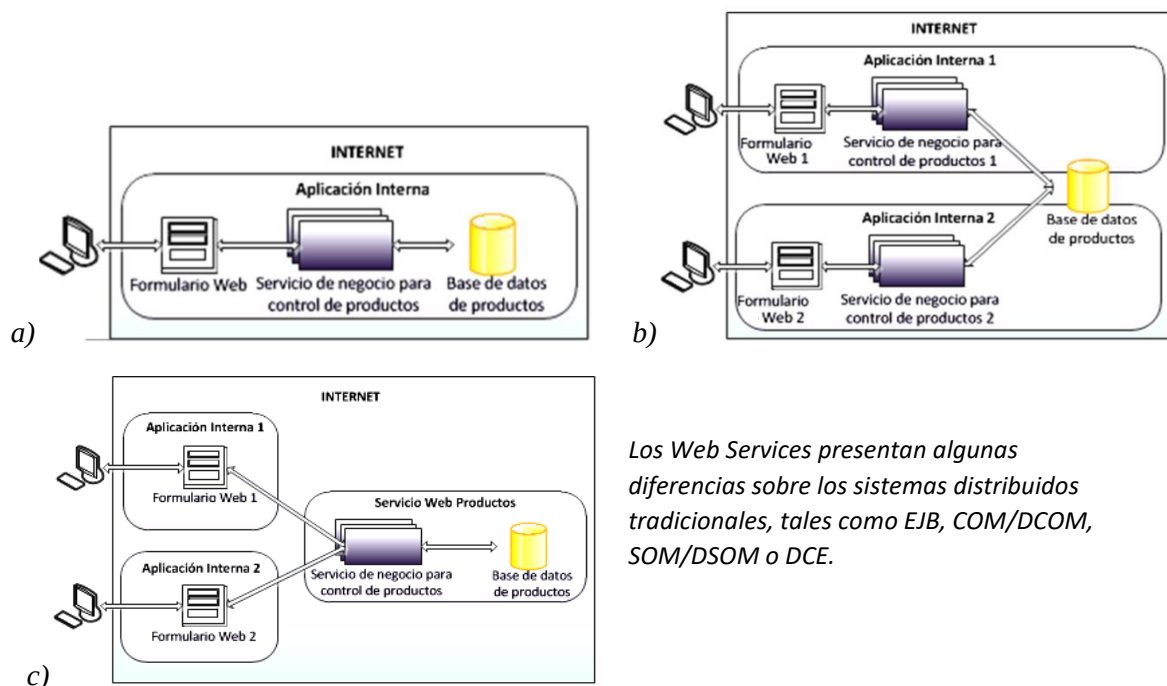
La OASIS y el W3C son los responsables de la arquitectura y reglamentación de los servicios Web. Según el W3C: *Un servicio web es un sistema software diseñado para soportar la interacción interoperable de máquina a máquina, a través de una red. Tiene una interfaz descrita en un formato procesable por una máquina (específicamente en WSDL). Otros sistemas interactúan con el servicio web de una manera establecida por su descripción utilizando mensajes SOAP¹², que generalmente se transmiten mediante HTTP¹³ con una serialización XML junto con otros estándares relacionados con la web [W3C, 2004].*

Casi todas las definiciones refieren aspectos técnicos comunes:

- Los servicios Web exponen funcionalidad útil a los usuarios Web mediante un **protocolo Web estándar**, que por lo general es el **SOAP**.
- Los servicios Web proporcionan un modo de autodescribirse detalladamente usando el WSDL. Esta descripción se proporciona generalmente en un documento XML.
- Los servicios Web se registran de modo que los potenciales usuarios puedan encontrarlos. Esto se realiza mediante **Universal Description Discovery and Integration (UDDI)**.

La mayoría de los sitios webs grandes (Facebook, MySpace, Microsoft) usan aplicaciones que utilizan servicios webs. Un servicio web es una función que diferentes servicios o equipos utilizan, solo se envían parámetros al servidor (lugar donde está alojado el servicio web) y éste responde la petición.

Un **Servicio Web** es un servicio de negocio programado en algún lenguaje, que envuelve cierta funcionalidad y la expone en una red de manera estandarizada, para que cualquiera conectado a esta misma red, que además soporte dichos estándares, pueda acceder a ella. Un ejemplo de la ventaja de usar servicios web se ve en la figura 4.1 [Desarrolloconsoa, 2014]:



Los Web Services presentan algunas diferencias sobre los sistemas distribuidos tradicionales, tales como EJB, COM/DCOM, SOM/DSOM o DCE.

Figura 4.1: Ventajas del uso de servicios web. Fuente: Desarrolloconsoa, 2014.

¹² SOAP: Simple Object Access Protocol: Protocolo Simple de Acceso a Objetos

¹³ HTTP: Hypertext Transfer Protocol: Protocolo de Transferencia de Hipertexto

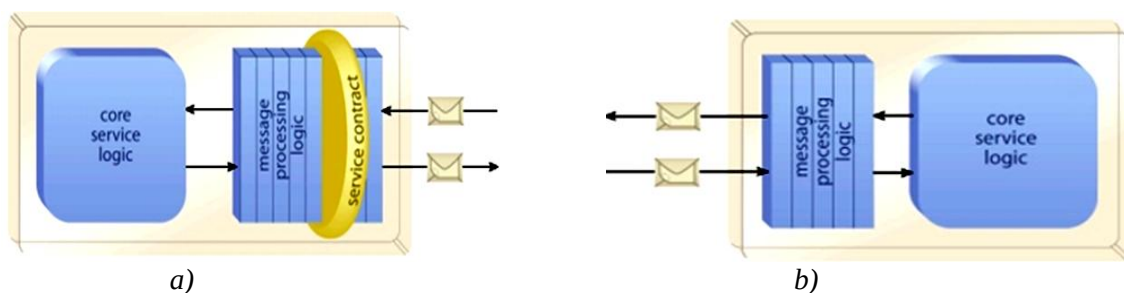
- Se tiene una aplicación interna de administración de venta de productos, y ésta involucra una aplicación web que usa servicios de negocio para realizar la consulta de productos y registrar una venta en una base de datos. Esta aplicación interna hace uso del servicio de negocio de control de productos, el cual encapsula la lógica para realizar las acciones sobre la Base de Datos. Esta solución está limitada ya que el servicio es exclusivo de esta aplicación.
- Si se quisiera tener otra aplicación diferente, donde se use el mismo servicio de productos, se tendría que duplicar, por ende el control y el mantenimiento se complican y puede volverse un caos.
- Para ello se tienen que realizar varios acuerdos entre las aplicaciones involucradas: es necesario que las 3 partes (aplicación 1, aplicación 2 y Servicio Web) establezcan comunicación por alguna vía, hablen el mismo idioma y puedan intercambiar información a través de mensajes, es decir, usen protocolos de comunicación estandarizados (Ej. SOAP).

4.2 - Componentes y roles de un servicio web

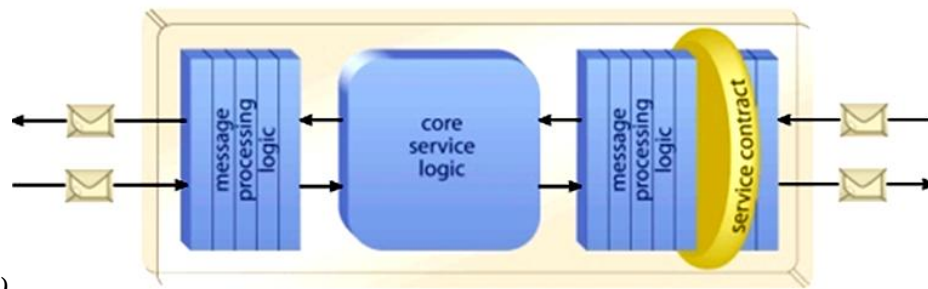
Por lo general un servicio web tiene los siguientes componentes:

- Un **contrato** de servicio técnico, físicamente desacoplado que consiste en una definición WSDL, una definición de esquema XML y, posiblemente, una definición de WS-Policy. Este contrato de servicio expone funciones públicas (llamadas operaciones) y, por lo tanto, es comparable a una API¹⁴.
- Un **cuerpo de lógica de programación**, lógica que es desarrollada específicamente para el servicio web, o bien se trata de una lógica heredada que está siendo envuelta por un servicio web para que su funcionalidad esté disponible a través de los estándares de comunicación de servicios web. En el primer caso se crea como componentes y se la denomina **lógica de servicio central** o **lógica de negocios**.
- Lógica de procesamiento de mensajes**, que existe como una combinación de analizadores, procesadores y agentes de servicio. Gran parte de esta lógica la proporciona el entorno de ejecución, pero también puede personalizarse. Los programas que llevan a cabo el procesamiento relacionado con el mensaje se basan principalmente en eventos y, por lo tanto, pueden interceptar un mensaje posterior a la transmisión o antes de su recepción. Es común que se invoquen múltiples programas de procesamiento de mensajes con cada intercambio de mensajes [Arcitura, 2018].

Un **servicio web** puede tener roles temporales, dependiendo de su utilización en tiempo de ejecución. Puede actuar como un **proveedor de servicios** cuando recibe y responde a los mensajes de solicitud, pero también puede asumir el rol de **consumidor de servicios** cuando se requiere emitir mensajes de solicitud a otros servicios web. Cuando los servicios web están posicionados dentro de las composiciones de servicio, es común que pasen por los roles de proveedor de servicio y consumidor (Fig. 4.2).



¹⁴ API: Application Programming Interface: Interfaz de programación de aplicaciones



c)

Figura 4.2: Roles de un servicio web: a) Un servicio web como proveedor de servicios; b) partes de un servicio web que actúan como consumidor de servicios; c) Transición de un servicio web a través de los roles del proveedor a consumidor de servicios. Fuente: Thomas Erl, 2005.

4.3 - Funcionamiento de un servicio web

Como está basado en el modelo cliente-servidor, se tiene el cliente consumidor (Service Requester) y el servidor (Service Provider). Un servicio web funciona de la siguiente manera (Fig. 4.3):

- El **Service Provider** define la descripción abstracta de un servicio utilizando WSDL.
- Luego se crea un **Servicio concreto “x”** a partir de la descripción abstracta del servicio.
- Dicha servicio concreto se “publica” en el UDDI (como si fuera una guía telefónica de servicios web).
- Un **Service Requester** o la aplicación del cliente requiere un servicio web “x”. Se pone en contacto con el UDDI para “localizarlo”. Identifica al proveedor.
- El cliente basándose en la descripción descrita por el archivo WSDL, envía un pedido del servicio web “x” a su proveedor (mediante protocolo SOAP).
- El proveedor valida la petición del servicio “x” y envía el dato estructurado en formato XML utilizando nuevamente el protocolo SOAP.
- El cliente valida el archivo XML utilizando un archivo XSD (XML Schema).

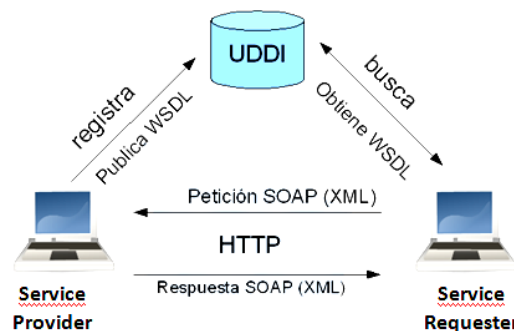


Figura 4.3: Esquema de funcionamiento de un servicio web. Fuente: Voorman H., 2006.

Un **servidor (o proveedor) de servicios web** requiere los siguientes componentes (Fig. 4.4):

- **Web Service:** Es el software o componente que realiza las operaciones. Si está escrito en Java, estas operaciones se realizarán en lenguaje Java. Los clientes invocarán estas operaciones enviando mensajes SOAP.
- **SOAP Engine:** El Web Service no sabe interpretar **SOAP requests** y crear **SOAP responses**. Para hacer esto hace falta un **SOAP Engine**, un software que se encarga del manejo de estos mensajes. *Apache Axis* es un ejemplo.
- **Application Server:** Para funcionar como un servidor que puede recibir **requests** desde diferentes clientes, el **SOAP Engine** normalmente funciona dentro de un **Application Server**. Este es otro software que proporciona un espacio libre para aplicaciones que han de ser accedidas por múltiples clientes. El **SOAP Engine**

funciona como una aplicación dentro del **Application Server**. Ejemplos son *Apache Tomcat server*, *Java Servlet* y *JSP container*.

- **HTTP Server:** Algunos **Application Servers** incluyen funcionalidades HTTP, por lo que se pueden tener **Web Services** funcionando instalando simplemente un **SOAP Engine** y un **Application Server**. Sin embargo cuando un Application Server carece de funcionalidad HTTP es necesario también un **HTTP server**, más comúnmente llamado **Web Server**. Es un software que sabe cómo manejar mensajes HTTP. Los dos más populares en la actualidad son *Apache HTTP Server* y *Nginx*.

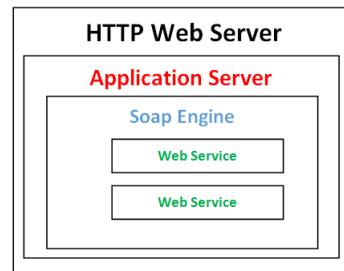


Figura 4.4: Componentes de un Servidor de servicios web.

4.4 - Ventajas de los servicios web

Entre las principales ventajas que proporcionan los servicios web se encuentran que:

- Aportan **interoperabilidad** entre aplicaciones de software independientes de sus propiedades o de las plataformas de las plataformas sobre las que se instalan.
- Los servicios web fomentan los estándares y protocolos basados en texto, que hace más fácil acceder a su contenido y entender su información.
- Al apoyarse en HTTP, los servicios web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- Permite que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- Permite la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar [Márquez, 2007].

4.5 - El Web Service

El **Web Service** es un sistema de comunicación entre diferentes servidores, a través de la red, basados en mensajes que cumplen un estándar basado en XML. El concepto de **Arquitectura de Web Service** es muy amplio que se desglosa en otras arquitecturas [W3C]:

- Arquitectura de interoperabilidad
- **Arquitectura orientada a servicios**
- Arquitecturas SOA y REST.

La **Organización para la Interoperabilidad de Servicios Web** (WSIO: Web Services Interoperability Organization) tiene como objetivo fomentar y promover la **Interoperabilidad de Servicios Web** (Web Services Interoperability: WS-I) sobre cualquier plataforma, sobre aplicaciones, y sobre lenguajes de programación. La organización busca integrar estándares para ayudar al avance de los servicios web de una manera estructurada y coherente. La WS-I ha organizado los estándares que afectan a la interoperabilidad de los servicios web en una pila basada en funcionalidades.

Plataforma de servicios web de Primera Generación

La plataforma de tecnología de servicios web original se compone de las siguientes tecnologías y especificaciones abiertas principales:

- Lenguaje de descripción de servicios web (WSDL)
- Lenguaje de definición de esquemas XML (XSD)
- SOAP (anteriormente el Protocolo de Acceso a Objetos Simples)
- UDDI (Descripción Universal, Descubrimiento e Integración)
- Perfil básico de WS-I

Estas especificaciones han existido durante algún tiempo y se han adoptado en toda la industria de las Tecnologías de la Información. Sin embargo, la plataforma que representan colectivamente carece seriamente de varias de las características de calidad de servicio requeridas para entregar una funcionalidad de producción de misión crítica a nivel empresarial.

Plataforma de servicios web de Segunda Generación (extensiones WS- *)

Algunas de las mayores brechas relacionadas con la **calidad del servicio** en la plataforma de primera generación se encuentran en las **áreas de seguridad** a nivel de mensaje, transacciones entre servicios y mensajes confiables. Estos, junto con muchas otras extensiones, están siendo proporcionados por la plataforma de servicios web de segunda generación. Este conjunto de tecnologías de servicios web, etiquetadas como "extensiones WS- *", ofrece importantes características mucho más complejas tanto en tecnología como en diseño. Algunas de ellas son:

- WS-Security (y WS-SX)
- WS-Coordination, WS-AtomicTransaction, WS-BusinessActivity (y WS-TX)
- WS-ReliableMessaging (y WS-RX)
- WS-Policy
- Direccionamiento WS.



Las telecomunicaciones se han vuelto parte fundamental de la vida actual, gracias a que las personas, organizaciones, países, empresas y gobiernos se encuentran interconectados de forma inmediata a través de éstas.

CAPÍTULO 5: ESTÁNDARES BÁSICOS

5.1 – Introducción

Los servicios web cumplen las propiedades de interoperabilidad, tienen interfaces fuertemente tipadas, usan los estándares de Internet existentes, tienen soporte multilinguaje, soportan otras infraestructuras de componentes, etc. Estas propiedades sólo son alcanzables gracias a la **base** que sustenta a esta tecnología. Se trata de **una serie de estándares que ofrecen seguridad y confiabilidad a todos los aspectos que rodean a los servicios web** (Fig. 5.1):

- UDDI: para la publicación y descubrimiento de servicios web
- WSDL: para la descripción formal de los servicios
- SOAP: para el formato de intercambios de mensajes en entornos distribuidos
- XML: para dar formato universal a la descripción de datos
- DISCO: para la búsqueda de servicios en el servidor
- HTTP: para la escritura y negociación de la representación de datos
- Otros.

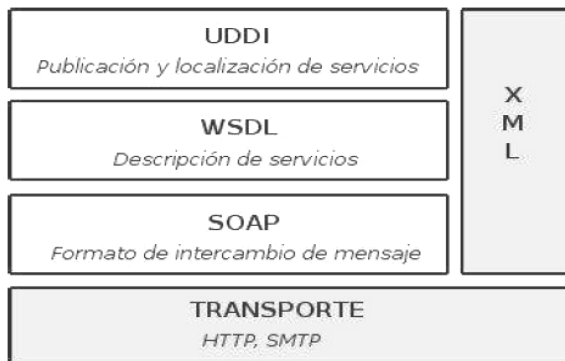


Figura 5.1: Estándares básicos.

Estos estándares sientan las bases y reglas a seguir por todos los usuarios y proveedores para una utilización universal, más allá de las plataformas y los lenguajes.

5.2 – XML: eXtensible Markup Language

Según el W3C, el **XML** (eXtensible Markup Language) es un formato de texto simple y muy flexible derivado del **Standard Generalized Markup Language** (SGML) o **Lenguaje Estándar Generalizado de Marcado** (Fig. 5.2).

XML es un **lenguaje de descripción de documentos** que no incluye ninguna información relativa al diseño de éstos. También se lo considera como un **meta-lenguaje** porque permite definir otros lenguajes de marcas con objetivos diferentes (Por eso se denomina eXtensible). Por ejemplo:

- XHTML (eXtensible, Hypertext Markup Language), nueva versión de HTML que cumple la especificación SGML y cuyo objetivo es sustituirlo como estándar de páginas web
- WML, MathML, etc.

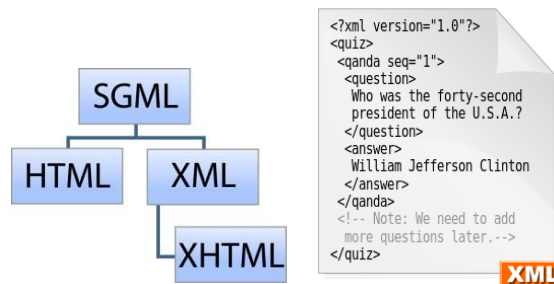


Figura 5.2: XML, su relación con SGML.

El XML desempeña un papel cada vez más importante en el intercambio de una amplia variedad de datos en la Web y en otros lugares, porque permite definir lenguajes de marcas, definir su gramática para estructurar los documentos, y puede ser utilizado para almacenar datos en forma legible [Paredes, 2018].

Por su parte, **HTML** (HyperText Markup Language) es el lenguaje de marcas (etiquetas) más conocido y utilizado para la creación de páginas web que permite la navegación tipo hipertexto. Ha sido concebido para mostrar información y dar formato a los diversos contenidos de una página.

Resumiendo, el **XML sirve para describir información** y el **HTML sirve para darle formato y presentarla a través de un navegador**. O sea que el XML no es ni será nunca un reemplazo del HTML sino un complemento que sirve para manejar la información separada del formato. XML aporta **flexibilidad** para estructurar los documentos.

Cualquiera puede desarrollar su propio lenguaje de etiquetas adecuado al tipo de aplicación más conveniente, como se ve en el ejemplo de la Fig. 5.3.

<pre><div> <p>El Quijote</p> <p><i>Cervantes</i> ~ 20 euros</p> <center></center> <p>Video promocional ~ Duración 30 segundos.</p> </div></pre>	<pre><libro> <titulo>El Quijote</titulo> <autor>Cervantes</autor> <precio moneda="euro">20</precio> <portada nombre="portada" tipo="gif"/> <video nombre="quijote" tipo="avi" duracion="30"/> </libro></pre>
--	--

Figura 5.3: HTML describe formato; XML describe estructura y semántica.

XML Schema

XML Schema fue desarrollado por la W3C, como un **lenguaje de esquema utilizado para describir la estructura y las restricciones de los contenidos de los documentos XML** de una forma muy precisa. Con este lenguaje se consigue un nivel alto de abstracción porque incorpora **mayor cantidad de tipos de datos** (como fechas, números y string).

DTD y XSD

Toda la descripción realizada con XML se guarda en un documento **DTD** (Document Type Definition). Allí se definen los tipos de elementos, atributos y entidades permitidas y las algunas limitaciones para combinarlos. También contiene los **espacios de nombres XML** que permiten separar semánticamente los elementos que forman el documento XML. En cada aplicación de XML es necesario conocer específicamente la relación que debe validarse de cada elemento presente en el documento. Esta relación se especifica en el DTD. Los documentos XML que se ajustan a su DTD son denominados **válidos**.

Los **documentos esquema**, denominados **XSD** (por XML Schema Definition) son creados usando el lenguaje **XML Schema**. Los XSD son algo similares a los DTD de XML pero ofrecen mayor capacidad en la definición de la estructura de datos y las restricciones del contenido de documento XML [Cruz del Valle, 2017].

5.3 – WSDL: Lenguaje de Descripción de servicios web

El **WSDL** (Web Service Description Language), es un **lenguaje para describir servicios web**, siendo su versión 2.0 la recomendada [W3C, 2007]. Con este lenguaje se puede describir la **interfaz pública** de un servicio web en un formato XML, como también las operaciones que tiene disponible, las estructuras de datos de los mensajes de cada operación, es decir, se describe la forma de comunicación, los requisitos del protocolo y los formatos de los mensajes.

Un ejemplo gráfico de su uso se esquematiza en la Fig. 5.4 [Desarrolloconsoa, 2014]:

- Para poder invocar un Servicio Web, es necesario crear un Cliente Consumidor que use el archivo wsdl, lo interprete correctamente y con toda la información que éste contiene, sea capaz de conectarse a él, invocarlo e intercambiar los mensajes en el formato esperado.
- Cuando se genera un Cliente Consumidor se realiza la importación del archivo wsdl, se descubren las operaciones expuestas en el Servicio Web y los mensajes que se requieren para cada uno, así el Cliente podrá ejecutarlos en cualquier momento en la manera tradicional, propia del lenguaje.
- Si otra aplicación desea usar el servicio se generará otro cliente consumidor. Por lo tanto, cada aplicación podrá invocar cuando lo requiera las operaciones expuestas en el Servicio Web, conectándose al **Endpoint**, especificado en el archivo wsdl. El **Endpoint** es la **ubicación del Servicio Web en la red**, es decir, el url donde efectivamente se está ejecutando. Así las invocaciones que realicen los Clientes son direccionadas al **Endpoint del Servicio Web** intercambiando mensajes SOAP y el archivo wsdl ya no es necesario.

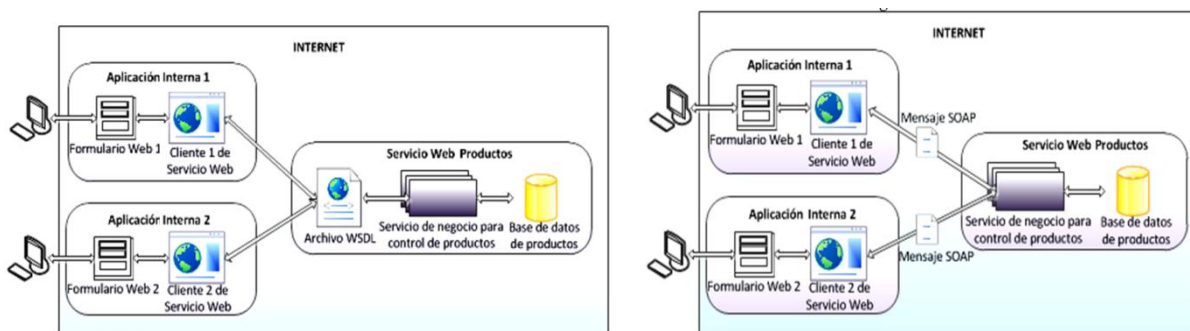


Figura 5.4: acceso a un servicio web usando archivos WSDL. Fuente: Desarrolloconsoa, 2014.

WSDL se usa a menudo en combinación con **SOAP** y **XML Schema**.

Estructura de un documento WSDL

WSDL describe un servicio utilizando varios elementos. Dichos **elementos** se pueden clasificar como **abstractos** o **concretos** [Fig. 5.5]. La **parte WSDL abstracta** describe las operaciones y mensajes con detalle, es decir, especifica “qué” hace el servicio: qué operaciones están disponibles; qué entradas, salidas, y mensajes de error tienen las operaciones; y cuáles son las definiciones de los tipos para los mensajes de entrada, salida y error. Esta parte contiene dos componentes principales: las **operaciones** que forman la

definición de la interfaz y los **tipos de datos** para los parámetros de entrada, salida y error, de las operaciones.

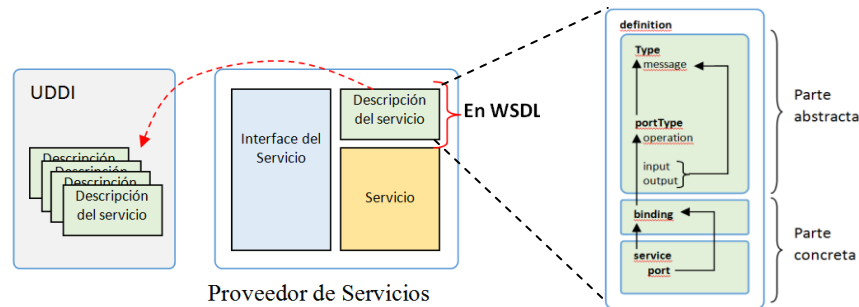


Figura 5.5: Estructura de un archivo WSDL.

La **parte WSDL concreta** describe el “cómo” y “dónde” del servicio, es decir, cómo tiene que llamar un cliente, qué protocolo debería usar y dónde está disponible el servicio. Esta parte contiene dos componente principales: la información de enlazado (binding) sobre el **protocolo** a utilizar y la **dirección en donde localizar el servicio** [DCCIA, 2014].

En conclusión, la WSDL 2.0 proporciona un modelo y un formato XML para describir **servicios web**, separando, por un lado, la descripción de la **funcionalidad abstracta** ofrecida por un servicio, y por el otro lado, los **detalles concretos** de una descripción del servicio.

5.4 – Estándar UDDI

OASIS crea el estándar **UDDI** (Universal Description Discovery & Integration) y la define como **un conjunto de servicios que apoyan la descripción y el descubrimiento de empresas, organizaciones y otros proveedores de servicios web, los servicios web que ofrecen y las interfaces técnicas que pueden utilizarse para acceder a esos servicios**.

Basado en un conjunto común de estándares de la industria, incluidos HTTP, XML, XML Schema y SOAP, UDDI proporciona una infraestructura interoperable y fundamental para un entorno de software basado en servicios web, tanto para servicios disponibles públicamente como para servicios que solo se exponen internamente dentro de una organización. La versión 3.0.2 describe los servicios web, las estructuras de datos y los comportamientos de todas las instancias de un registro UDDI [OASIS, 2004].

Registro UDDI

A partir de la especificación, se crea un directorio distribuido de servicios web, donde los datos se almacenan en XML. En este registro, se guarda la información sobre servicios: información sobre las organizaciones que los proporciona, la categoría en la que se encuentran, y sus instrucciones de uso, normalmente usando documentos WSDL. Para acceder al registro se utilizarán mensajes SOAP, que son transportados mediante protocolo HTTP (Fig. 5.6). Por ello se tiene 3 tipos de información relacionados entre sí [DCCIA, 2014]:

- Páginas blancas: Datos de las organizaciones como dirección, información de contacto y otros identificadores conocidos.
- Páginas amarillas: Clasificación de las organizaciones basada en taxonomías (según tipo de industria, zona geográfica, etc.)
- Páginas verdes: Información técnica sobre los servicios que se ofrecen. Aquí se dan las instrucciones para utilizar los servicios. Es recomendable que estas instrucciones se especifiquen de forma estándar mediante un documento WSDL.

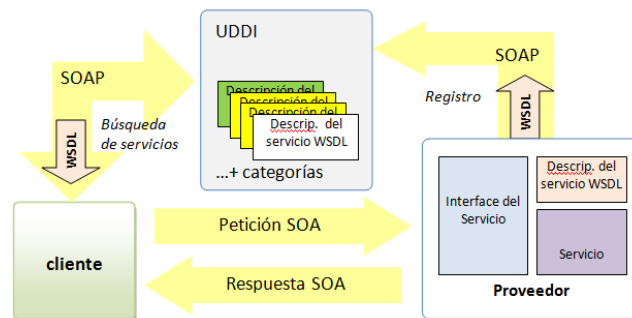


Figura 5.6: Repositorio UDDI con archivos WSDL.

5.5 – Protocolo HTTP

El **Protocolo de Transferencia de Hipertexto** (HTTP: HyperText Transfer Protocol) fue desarrollado por la W3C y por la IETF¹⁵. Se trata de un protocolo de comunicación para la transferencia de información en la World Wide Web. Permite definir la sintaxis y la semántica para la comunicación de los elementos de software de una arquitectura web, ya sean clientes, servidores, proxies, etc.

Según la W3C, el HTTP es un protocolo de **nivel de aplicación** para sistemas de información distribuidos, colaborativos e hipermedia. Es un protocolo genérico, **sin estado**, que puede usarse para muchas tareas más allá de su uso para hipertexto, como servidores de nombres y sistemas de gestión de objetos distribuidos, mediante la extensión de sus métodos de solicitud, códigos de error y encabezados. Una característica de este protocolo es la **escritura y negociación de la representación de datos**, lo que permite que los sistemas se construyan independientemente de los datos que se transfieren.

Atención de una solicitud

El HTTP es un **protocolo cliente-servidor**, es decir, orientado a transacciones que sigue un esquema de solicitud/respuesta entre un consumidor y un proveedor. El cliente (por ejemplo, un navegador web), es un agente usuario que realiza la solicitud enviando un mensaje con un determinado formato hacia el servidor. La información que se envía se conoce como **recurso**, puede ser un archivo, el resultado de la ejecución de un programa, una consulta a una base de datos, etc. A este recurso se lo identifica mediante un **localizador uniforme de recurso** (URL: Uniform Resource Locator). El servidor es el agente que envía un mensaje de respuesta a la solicitud, suelen denominarse **servidor web** (Fig. 5.7).

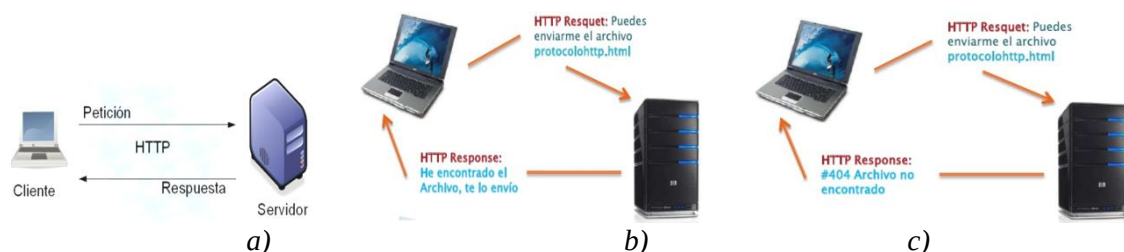


Figura 5.7: a) HTTP en el esquema cliente-servidor; b) respuesta exitosa; c) respuesta no exitosa.

Que el HTTP sea un protocolo **sin estado** quiere decir que, en general, no guarda información sobre las solicitudes anteriores o conexiones anteriores, por lo tanto cada solicitud es independiente. Sin embargo, las aplicaciones web requieren frecuentemente de la mantención de un estado. Para esto, se utilizan las **cookies**. Una **cookie** es una pequeña

¹⁵ IETF: Internet Engineering Task Force: Grupo de Trabajo de Ingeniería de Internet.

información enviada por un sitio web que es almacenada en el navegador del usuario, de manera que un sitio web pueda consultar la actividad previa del navegador [Rodríguez, 2017].

Mensajes HTTP

El HTTP usa mensajes en texto plano, que los hacen más legibles y fácil de depurar, aunque pueden llegar a ser largos [IETF, 2014]. Un mensaje HTTP tiene la siguiente estructura (Fig. 5.8):

- una línea inicial: en caso de ser una **petición**, contiene el **método de petición**, una URL de recurso y la versión de HTTP; en el caso de una respuesta se indica la versión del HTTP, **código de respuesta** (un número que indica el estado de la solicitud junto al URL)
- la cabecera: que consta de metadatos que aporta flexibilidad al protocolo
- el cuerpo: que consta de metadatos que aporta flexibilidad al protocolo. El cuerpo del mensaje, que es opcional, y contiene los datos intercambiados por el cliente y el servidor. En una **respuesta** incluirá los datos solicitados por el cliente.

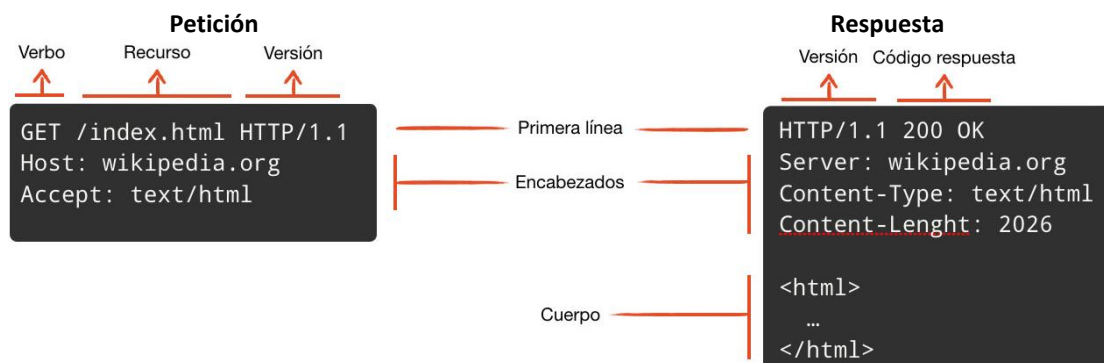


Figura 5.8: Estructura de un mensaje HTTP.

El **método de petición**, previamente definido por HTTP, indica la acción que desea que se efectuar sobre el recurso identificado. El protocolo tiene la flexibilidad para ir añadiendo nuevos métodos, los que significa nuevas funcionalidades. Algunas veces se refiere a los métodos de petición como verbos: GET, HEAD, POST, PUT, otros.

El **código de respuesta** es un número de tres dígitos que tiene un significado concreto. Hay 5 categorías: 1xx (información), 2xx (éxito), 3xx (redirección), 4xx (error en el cliente), 5xx (error en el servidor).

Seguridad con HTTPS

HTTPS es un protocolo de aplicación basado en el protocolo HTTP destinado a la **transferencia segura** de datos de Hipertexto, es decir, es la **versión segura de HTTP**. La comunicación es encriptada, mediando un **certificado de seguridad** entre las partes.

Inicialmente, las conexiones HTTPS utilizaban el protocolo de seguridad SSL (Secure Socket Layer) para proteger las conexiones de los usuarios al utilizar plataformas muy concretas (ej.: webs bancarias). Hoy en día, este protocolo **ha cambiado SSL por TLS**, (Transport Layer Security), protocolo que nos brinda mayor seguridad y mayor privacidad en las conexiones.

Mientras cualquier usuario puede **interceptar y/o modificar el contenido de los paquetes HTTP**, las conexiones HTTPS no pueden ser espiadas ni modificadas (Fig. 5.9).

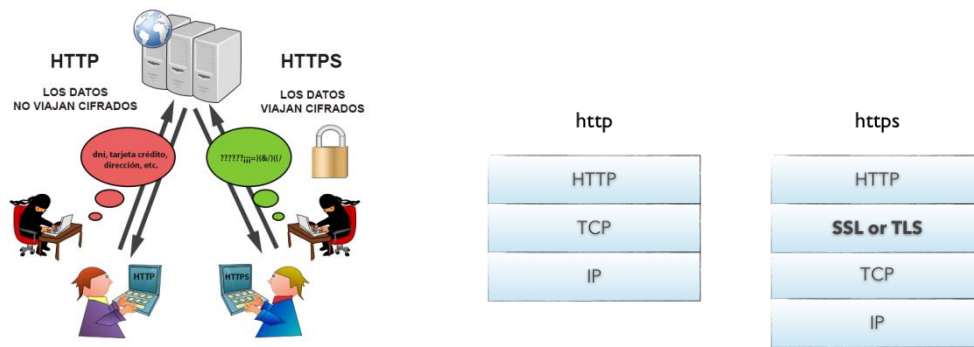


Figura 5.9: Comunicación segura con HTTPS, usando encriptación. Fuente: OSI-españa, 2014.

Versión mejorada con HTTP/2

HTTP/2 es el Protocolo de Transferencia de Hipertexto 2.0 y es la versión actualizada del protocolo HTTP. Las grandes mejoras tiene que ver con la velocidad de transferencia (entre 10 y 30%), y por consiguiente, con la carga de las páginas web y otros servicios.

5.6 – Protocolo SOAP

SOAP (Simple Object Access Protocol: Protocolo de Acceso Simple a Objetos) es un **protocolo liviano para el intercambio de información en un entorno descentralizado y distribuido**. Inicialmente fue concebido como la solución a la brecha entre plataformas basadas en RPC¹⁶, pero en la actualidad es el más usado para el intercambio de mensajes entre servicios web.

Antes de su surgimiento, las empresas utilizaban CORBA, COM y EJB para el desarrollo aplicaciones distribuidas en ambientes heterogéneos. Cada una proporciona un marco de trabajo basado en la activación de objetos remotos mediante la solicitud de ejecución de servicios de aplicación a un servidor de aplicaciones. Si bien son altamente efectivas, sin embargo presentan **incompatibilidad** y **falta de interoperabilidad** entre ellas. En este contexto, SOAP es un nuevo modelo de computación distribuida de objetos, **independiente de plataformas, de modelos de desarrollo y de lenguajes de programación** [Cubo Velázquez, 2018].

SOAP no es más que un mecanismo sencillo de expresar la información mediante un modelo de empaquetado de datos modular y una serie de mecanismos de codificación de datos. Esto permite que sea utilizado en un **amplio rango de servidores de aplicaciones** que trabajen mediante el modelo de comunicación RPC. SOAP consta de tres partes:

- El **SOAP envelope**, que define el marco de trabajo para describir el qué se puede introducir en un mensaje, quién debería hacerlo y si esa operación es opcional u obligatoria
- Las **reglas de codificación SOAP**, que definen el mecanismo de serialización (empaquetado) para encapsular en los mensajes los distintos tipos de datos
- La **representación SOAP RPC**, que define un modo de funcionamiento para una llamada RPC y obtener respuestas.

SOAP y su interacción con otras plataformas

SOAP se caracteriza por **interoperabilidad** y **funcionalidad**: mediante un mecanismo simple y ligero, usando XML, sobre HHTP, se intercambia información entre dos puntos (Fig. 5.10).

¹⁶ RPC: Remote Procedure Call: llamada a procedimiento remoto, muy utilizado en ambientes cliente-servidor.

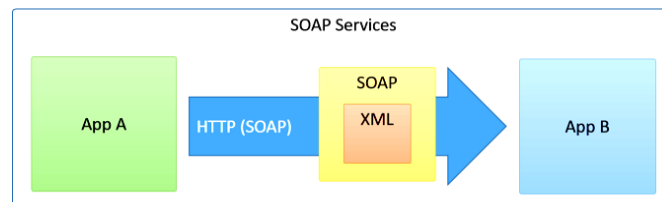


Figura 5.10: SOAP usando XML, sobre HTTP.

Los servidores Web pueden procesar las peticiones de usuario empleando tecnologías tales como Servlets, páginas ASP (Active Server Pages), páginas JSP (Java Server Pages) o sencillamente un servidor de aplicaciones con invocación de objetos mediante CORBA, COM o EJB (Fig. 5.11).

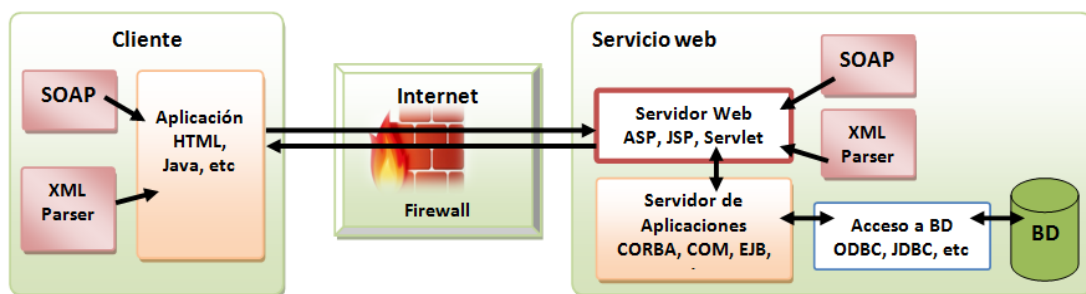


Figura 5.11: SOAP y su interacción con otras plataformas. Fuente: Cubo Velázquez, 2018.

La especificación SOAP indica que las aplicaciones deben ser independientes del lenguaje de desarrollo, por lo que las aplicaciones cliente y servidor pueden estar escritas con HTML, DHTML, Java, Visual Basic o cualquier otra herramienta o lenguaje disponibles.

Formato del mensaje SOAP

Las especificaciones de SOAP establecen un formato de mensaje que consta de un **documento XML** (capaz de contener un RCP) y un **documento centrado en datos**. Esto facilita tanto la sincronía (petición y respuesta) como la asincronía (Procesos impulsados por un evento) de intercambio de datos entre modelos [Cubo Velázquez, 2018].

Un documento XML correctamente definido e incluido en un elemento SOAP, se denomina **mensaje SOAP** (Fig. 5.12). Este consta de tres partes [W3C, 2000]:

- El **envoltorio** (envelope): define que contiene el mensaje y como procesarlo). Es obligatorio.
- El **encabezado** (header): permite enviar información relativa a como debe ser procesado el mensaje. Es opcional.
- El **cuerpo** (body): contiene información obligatoria destinada al destinatario final del mensaje. SOAP define un elemento fault para el cuerpo, utilizado para informar errores.

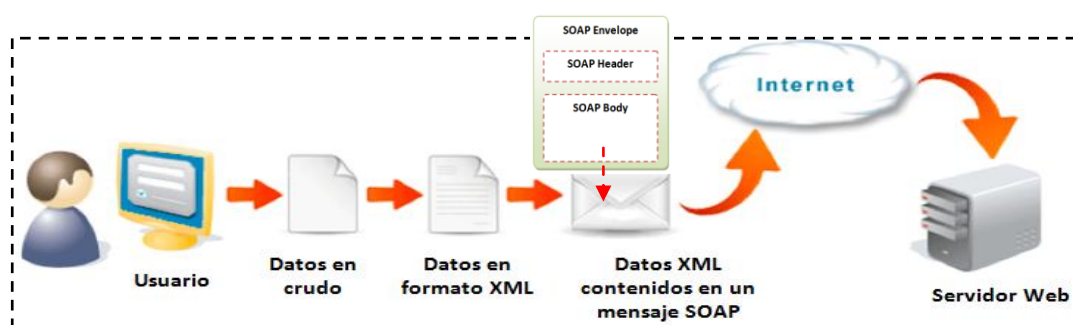


Figura 5.12: Mensaje SOAP en la comunicación cliente-servidor.

Comunicación cliente-servidor mediante SOAP

De forma global en el siguiente diagrama se puede ver la interacción entre las diferentes componentes de la comunicación por servicios web. El cliente y el servidor se comunican mediante mensajes SOAP, a través del protocolo de comunicación HTTP que permite la transferencia en Internet (WWW). La forma de comunicarse se establece en el documento WSDL (Fig. 5.13). También se puede ver como el cliente encuentra el servicio web que ofrece el servidor en el catálogo.

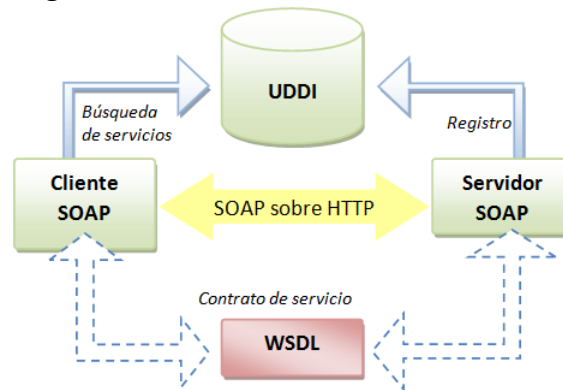


Figura 5.12: Esquema de comunicación cliente-servidor mediante SOAP.

La ubicuidad de HTTP y la simplicidad de SOAP los convierten en una base ideal para implementar servicios Web XML que pueden consumirse desde prácticamente cualquier entorno.

Se caracteriza por su **simplicidad y extensibilidad**.

5.7 – La tecnología REST

REST (Representational State Transfer: **Transferencia de estado representacional**) surge de una tesis doctoral escrita por Roy Thomas Fielding en el año 2000, quien es uno de los autores principales de la especificación de HTTP. Fielding define a REST como: *un estilo híbrido derivado de varios estilos arquitectónicos basados en red y combinados con las restricciones adicionales que definen una interfaz de conector uniforme*.

El desarrollo de REST parte un estilo nulo que es simplemente un conjunto vacío de restricciones. Un diseñador comienza con las necesidades el sistema como un todo, sin restricciones, y luego identifica y aplica incrementalmente las restricciones a los elementos del sistema para diferenciar el espacio de diseño y permitir que las fuerzas que influyen en el comportamiento del sistema fluyan naturalmente, en armonía con el sistema [Fielding, 2000].

Descripción de REST

REST hace **referencia estrictamente a una colección de principios para el diseño de arquitecturas en red**. Estos principios determinan cómo deben definirse y analizar los recursos. El término frecuentemente es utilizado en el sentido de **describir cualquier interfaz entre sistemas que utilice directamente HTTP para obtener datos o indicar la ejecución de operaciones sobre los datos, en cualquier formato (generalmente XML) sin las abstracciones adicionales de los protocolos basados en patrones de intercambio de mensajes**, como es, por ejemplo, SOAP [Espinosa, 2012].

Los **servicios web basados en REST intentan emular al HTTP** o protocolos similares mediante la restricción de establecer la interfaz a un conjunto conocido de operaciones

estándar (por ejemplo GET, PUT, etc.). Por lo tanto, este **estilo se centra más en interactuar con recursos**, que con mensajes y operaciones.

Es posible diseñar sistemas de servicios web de acuerdo con el estilo REST y también es posible diseñar interfaces XML/HTTP con RPC, pero sin usar SOAP. Estos dos usos diferentes de REST causan cierta confusión en las discusiones técnicas, aunque RPC no es un ejemplo de REST [Navarro, 2006]. Cabe destacar que aunque **REST no es un estándar**, está basado en estándares como HTTP y URL.

REST aprovecha la escalabilidad que obtiene la web a partir de:

- Un protocolo cliente/servidor sin estado mediante el uso del HTTP y el URL.
- Un conjunto de operaciones bien definidas que se aplican a todos los recursos de información, como POST, GET, PUT y DELETE definidos por HTTP.
- Una sintaxis universal para identificar los recursos. En un sistema REST, cada recurso es direccionable únicamente a través de su URI.
- El uso de hipermedias, tanto para la información de la aplicación como para las transiciones de estado de la aplicación: la representación de este estado en un sistema REST son típicamente HTML o XML.

Esquema de REST

El **esquema** utilizado por REST es el esquema cliente/servidor. En este esquema los componentes de la red (clientes y servidores) se comunican a través de una interfaz estándar, como HTTP, e intercambian representaciones de recursos [Espinosa, 2012].

En la REST, un **recurso** es un elemento informático que puede ser accedido utilizando un **identificador global URI**¹⁷. Un ejemplo de recurso son los archivos que se envían y se descargan. Para interactuar con un recurso utilizando una **solicitud** es necesario solamente conocer el URI y la acción requerida. Las solicitudes se envían a través de conectores (ya sean clientes, servidores, cachés, etc.) que lo hacen sin ver más allá de la propia solicitud, lo que se conoce como **stateless** (sin estado). Es decir, no es necesario conocer si existen cachés, proxies, cortafuegos o cualquier otro conector entre el cliente y el servidor. Esta es una restricción de REST, común en otras partes de la arquitectura de redes y de la informática. La aplicación, sin embargo, debe comprender el formato de la información devuelta (la representación), que es por lo general un documento HTML o XML, aunque también puede ser una imagen o cualquier otro contenido.

Comparación entre SOAP y REST

Tanto SOAP como REST siguen la arquitectura SOA (Fig. 5.14), por lo que las dos deberían de apegarse a los mismos principios. La diferencia está en que SOAP es un protocolo, mientras que REST es un estilo arquitectónico.

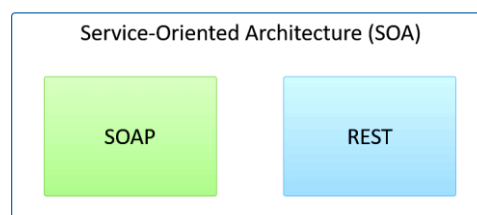


Figura 5.14: REST en SOA.

¹⁷ URI: Uniform Resource Identifier.

REST es una **tecnología mucho más flexible que transporta datos por medio del protocolo HTTP**, pero este permite utilizar los diversos métodos que proporciona HTTP para comunicarse, como lo son GET, POST, PUT, DELETE, PATCH y a la vez, utiliza los códigos de respuesta nativos de HTTP (404,200,204,409). Su flexibilidad se debe a que permite transmitir prácticamente cualquier tipo de datos, ya que el tipo de datos está definido por el Header Content-Type, lo que permite mandar, XML, JSON¹⁸, Binarios (imágenes, documentos), Text, etc. que contrasta con SOAP que solo permite la transmisión de datos en formato XML.

A pesar de la gran variedad de tipos de datos que podemos mandar con REST, la gran mayoría transmite en JSON por un motivo muy importante, JSON es interpretado de forma natural por JavaScript, lo que ha hecho que frameworks como Angular y React se aprovechen al máximo, ya que pueden enviar peticiones directas al servidor por medio de AJAX y obtener los datos de una forma nativa. Por ejemplo, los formularios de HTML pueden ser apuntados a los servicios REST sin ningún problema (Fig. 5.15).

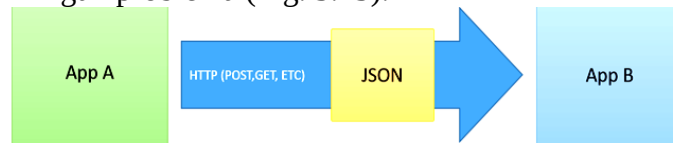


Figura 5.15: Comunicación en REST.

La Tabla 5.1 tiene muestra una comparación entre SOAP y REST.

SOAP	REST
SOAP es un protocolo	REST es un estilo arquitectónico
SOAP significa Protocolo de acceso simple a objetos	REST significa Transferencia de estado representacional
Desde SOAP no se puede utilizar REST, debido a que SOAP es protocolo	REST puede utilizar el SOAP de servicios web, ya que es un concepto y se puede utilizar cualquier protocolo como HTTP, SOAP
SOAP utiliza interfaces de servicios para exponer la lógica de negocio	REST utiliza URI para exponer la lógica de negocio.
JAX-WS es la API de Java para servicios web SOAP	JAX-RS es su propia interface
Desde SOAP se definen las normas que deben seguirse estrictamente	REST no define demasiado estándares como SOAP
SOAP requiere más ancho de banda porque hace definición de todo en su formato	REST requiere menos ancho de banda ya que solo define su arquitectura
Desde SOAP se define su propia seguridad	Servicios web RESTful hereda las medidas de seguridad del transporte subyacente
SOAP permite XML como único formato de datos	REST permite diferentes formatos de datos tales como texto son formato, HTML, XML, JSON, etc.

Tabla 5.1: Comparación entre SOAP y REST.

Principios de diseño

La arquitectura de REST tiene que cumplir con 6 principios:

- Cliente-servidor
- No maneja estado
- Capaz de almacenarse en cachés
- Sistemas por capas
- Código bajo demanda (por ejemplo Java Applets o JavaScript)
- Interface Uniforme.

¹⁸ JSON: Java Script Object Notation: Notación de Objeto de JavaScript. Se trata de un formato de texto ligero para el intercambio de datos. Es un subconjunto de la notación literal de objetos de JavaScript. En la actualidad se considera como un lenguaje independiente.



“SOA es un paradigma de diseño, va desde la forma en que desarrollamos nuestro código hasta la manera en cómo la compañía lo administra, usa y valora”.

Edgar García
Enterprise SOA Architect

CAPÍTULO 6: ARQUITECTURA ORIENTADA A SERVICIOS

6.1 – Conceptualizando SOA

Para comenzar la **Arquitectura Orientada a Servicios** (SOA: Service Oriented Architecture, en inglés) es la combinación de los dos conceptos fundamentales: sistemas distribuidos y servicios web. Pero su alcance va más allá. SOA es un paradigma, **cuyo objetivo fundamental es la utilización de “servicios” para satisfacer los requerimientos del cliente a través de un protocolo común de comunicaciones**, donde la aplicación Web Service es clave fundamental para su construcción.

Algunas definiciones de SOA

Según el **W3C**, “SOA es el conjunto de componentes que pueden ser invocados, cuyas descripciones de interfaces se pueden publicar y describir”.

Para el **Group Gartner**, “SOA es una arquitectura de software que comienza con una definición de interface y construye toda la topología de la aplicación como una topología de interfaces, implementaciones y llamados a interfaces. Sería mejor llamada “arquitectura orientada a interfaces”. SOA es una relación de servicios y consumidores de servicios, ambos suficientemente amplios para representar una función de negocios completa”.

Según **IBM**, “SOA representa una forma de construir sistemas distribuidos que permite ofrecer las funcionalidades de una aplicación como servicios tanto para aplicaciones de usuario final ó a otros servicios”.

Para **BEA**¹⁹, “Es una estrategia de TI que organiza las funciones discretas contenidas en las aplicaciones empresariales en servicios estandarizados, interoperables, de forma que puedan ser combinados y reusados fácil y rápidamente para adaptarse a los requerimientos del negocio”.

Para **OASIS**, “SOA es un paradigma para organizar y utilizar capacidades distribuidas que pueden estar bajo el control de diferentes dominios. Provee una manera uniforme de ofrecer, descubrir, interactuar con ellos y sus capacidades de uso para producir el efecto deseado consistente con precondiciones y expectativas medibles”.

Para **SUN**, “Una SOA es una estrategia donde las aplicaciones hace uso (o se basan) en servicios disponibles en una red. Siendo una manera de compartir funciones (típicamente de negocios) en una manera flexible y extendida”.

¹⁹ BEA es la Oficina de Análisis Económico de EEUU.

Para **CTP**²⁰: “Una SOA es un enfoque para diseñar y construir soluciones de negocio a partir de componentes independientes que exponen funciones como servicios accesibles por otros componentes a través de interfaces estándares” (Fig. 6.1).

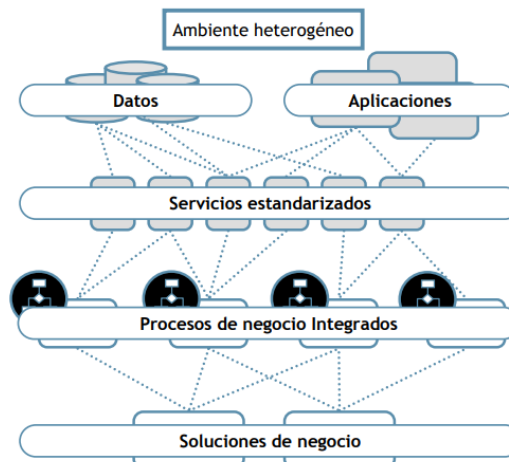


Figura 6.1: Enfoque SOA. Fuente: Consultoría CTP, 2007.

Evolución de la arquitectura

Una arquitectura empresarial está formada por diferentes soluciones que se deben integrar. La integración de sistemas significa compartir datos y procesos de negocios en forma irrestricta entre distintas aplicaciones interconectadas, de forma tal que cada proceso de negocio o porción de datos sea implementado de la mejor manera por la aplicación más adecuada, y compartido al resto [Escobar, Passerini, Arias y Blanco, 2006]. En la búsqueda de la integración fueron surgiendo nuevas arquitecturas (Fig. 6.2):

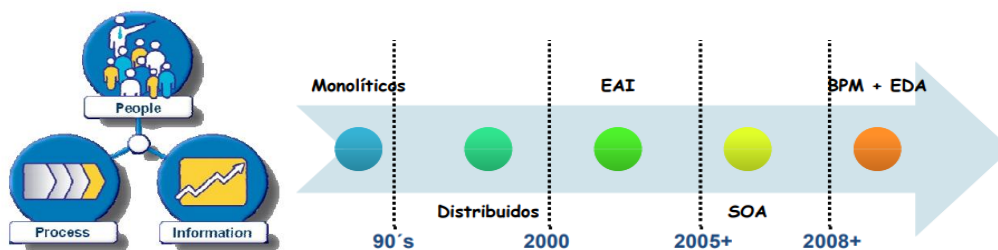


Figura 6.2: Evolución de SOA. Fuente: Escobar y otros, 2006.

Integración Punto a Punto (Fig. 6.3.a): Tiene una inversión inicial baja, pero el mantenimiento se hace costoso. El número de conexiones crece exponencialmente a medida que se incorporan aplicaciones. Requiere desarrollo de funciones complejas para comunicación, transformaciones, etc. Altamente dependiente de las aplicaciones.

Integración de aplicaciones empresariales (EAI: Enterprise Application Integration) (Fig. 6.3.b): Altamente escalable en cuanto a número de aplicaciones y volumen de datos, pero está fuertemente atada a la tecnología del proveedor del bus de integración. Provee funciones de comunicación confiable, transformaciones, enrutamiento, etc. Se justifica principalmente para proyectos de gran envergadura y grandes volúmenes de datos.

²⁰ CTP: es una consultora líder con presencia en la Región Andina (Colombia), que trabaja con 70 consultores y arquitectos con experticia en aplicaciones CRM, herramientas de integración EAI/SOA, Herramientas de Inteligencia de Negocio, y tecnologías emergentes.

Integración Orientada a Servicios (Fig. 6.3.c): Utiliza infraestructuras basadas en estándares. La escalabilidad depende en gran medida de la infraestructura utilizada. Puede desarrollarse incrementalmente. Ofrece funciones de comunicación y transformación de datos. Puede coexistir y complementar una infraestructura de EAI. SOA pertenece a esta línea.

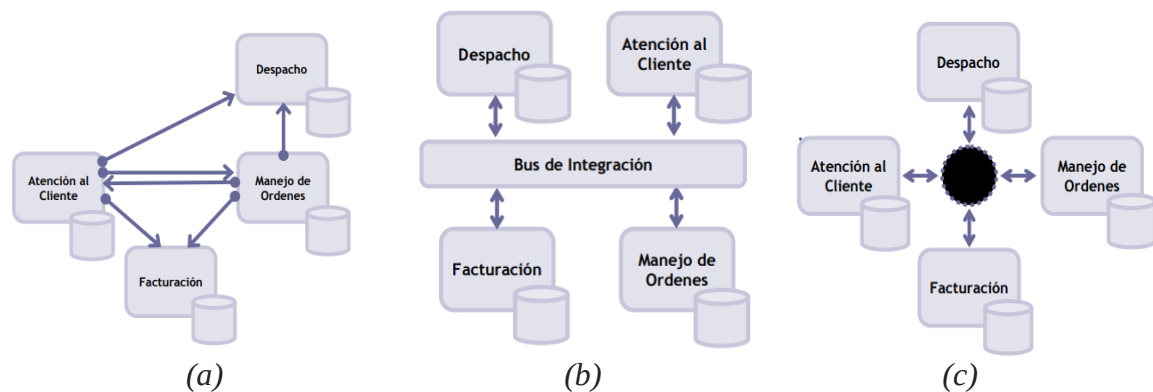


Figura 6.3: a) Integración punto a punto; b) integración de aplicaciones empresariales; c) integración orientada a servicios. Fuente: Consultoría CTP, 2007.

Características de SOA

- Se basa en el diseño de servicios que reflejan las actividades del negocio en el mundo real.
- Los servicios se representan utilizando **descripciones de negocio** para asignarles un contexto a tal fin.
- Requiere **infraestructura específica y única para este tipo de arquitectura**. En general se usa estándares abiertos para la interoperabilidad y transparencia en la ubicación de servicios.
- Está implementada de acuerdo a las **condiciones específicas de la arquitectura de TI en cada compañía**.
- Requiere un gobierno fuerte sobre la representación e implementación de servicios.
- Requerir un conjunto de pruebas que determinen que es un buen servicio [The Open Group, 2018].

6.2 - Los servicios

Se puede definir **servicio** como un **paquete funcional de software que es autónomo, sin estado, autocontenido, que acepta llamadas y devuelve respuestas** mediante una interfaz bien definida y el acceso a una infraestructura de red.

Los **servicios sin estado** no dependen de ninguna condición preexistente de algún otro servicio: reciben en la llamada **toda la información que necesitan para dar una respuesta**. Por esta característica, los servicios pueden ser secuenciados (orquestrados) en numerosas secuencias (pipelines²¹ o tuberías) para realizar la lógica del negocio.

A cada servicio se le **asigna su propio contexto funcional distinto y se compone de un conjunto de capacidades relacionadas con este contexto**. Esas capacidades adecuadas para la invocación por parte de programas de consumidores externos se expresan comúnmente a través de un **contrato de servicio** publicado como una API tradicional (Fig. 6.4).

²¹ El *pipeline* es una técnica para implementar el paralelismo a nivel de instrucciones dentro de un solo procesador.

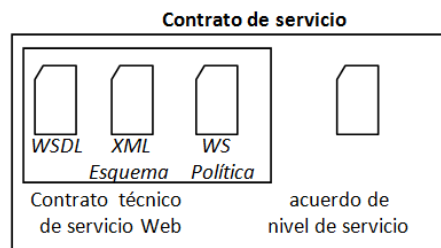


Figura 6.4: Documentos del contrato de un servicio web.

Clasificación de los servicios

El foco de SOA está en la **infraestructura funcional y sus servicios del negocio**, no en la infraestructura técnica y sus servicios técnicos. La entidad básica es la **función o proceso del negocio**. Los servicios se clasifican de acuerdo a sus características de diseño, implementación y gestión del proyecto (reusabilidad, mantenibilidad, escalabilidad, performance):

- **Básicos:** Son la base de SOA, son servidores puros y no mantienen el estado conversacional de la sesión. Se dividen en:
 - *servicios centrados en los datos*, que maneja los datos persistentes, de almacenamiento y recuperación del negocio
 - *centrados en la lógica*, que proveen encapsulamiento para cálculos complejos o reglas del Negocio.
- **Intermedios:** Son **servicios sin estado** que hacen de puente entre las inconsistencias técnicas o discrepancias conceptuales en el diseño. Son tanto clientes como servidores en SOA, mediando entre los distintos elementos que deben funcionar juntos. A su vez se dividen en:
 - *Technology gateways*: Estos servicios hacen de puente entre discrepancias tecnológicas, incorporando dos o más tecnologías para comunicación o codificación de datos. Hacen de proxies para sus servicios de Negocio y representan la funcionalidad de los servicios que están por debajo en un ambiente que es distinto tecnológicamente que el ambiente de ejecución del servicio de Negocio original.
 - *Adapters*: Es un tipo especial de servicio intermediario que mapea las firmas y formatos de mensaje de un servicio a los requerimientos de un cliente.
 - *Facades*: Su propósito es proveer una vista distinta (y posiblemente agregada) de uno o más servicios existentes, por lo que también pueden actuar como technology gateways y/o adapters. Puede utilizarse para proveer una vista específica de un conjunto de servicios básicos.
 - *Functionality - adding*: Este tipo de servicio se usa cuando se quiere agregar funcionalidad a un servicio existente sin cambiar el servicio mismo. En este caso se crea un servicio que provea la misma funcionalidad del servicio original y agregue las nuevas características requeridas.
- **Centrados en procesos:** encapsulan el conocimiento de los procesos del negocio de la Organización, controlan y mantienen el estado del proceso en ejecución. Desde el punto de vista técnico son la clase de servicios más sofisticada. Actúan tanto de clientes como de servidores en una SOA, manteniendo el estado del proceso para sus clientes. Su ámbito es dentro de la empresa.
- **Empresariales públicos:** son servicios que una empresa ofrece a socios y clientes para su consumo. Estos servicios tienen requerimientos específicos de interface, desacoplamiento, seguridad y facturación de uso, ya que las empresas deben acordar claramente cómo se realiza el uso de los mismos.

6.3 - Modelo conceptual para SOA

El modelo conceptual SOA permite visualizar claramente las capas de software en donde los diferentes servicios se alinean con los procesos de negocios:

- **Capa 1: Sistemas Operacionales.** Esta capa abarca **las aplicaciones existentes dentro de la empresa**, conocidas como *legacy systems*, que son sistemas desarrollados bajo cualquier arquitectura o tecnología, geográficamente dispersos y bajo cualquier figura de propiedad. Entre ellas se encuentran CRM's²², ERP's²³, aplicaciones de BI²⁴, Orientadas a objetos, integradas a través de SOA.
- **Capa 2: Capa de Componentes.** Esta capa contiene **los componentes que brindan la funcionalidad que exponen los servicios**. En ella se usan tecnologías para contener los componentes que existen dentro de esta, tales como servidores de aplicaciones.
- **Capa 3: Capa de Servicios.** En esta capa **residen los servicios** (generalmente como servicios web) que la organización decide exponer. Pueden ser descubiertos, referenciados directamente, o ser parte de una orquestación o de un servicio compuesto. Normalmente estos servicios exponen la funcionalidad de negocio a través de contratos que permiten invocar los componentes de negocio que se encuentran en la capa de componentes de la empresa.
- **Capa 4: Procesos de Negocio – Orquestación.** En esta capa se **exponen las orquestaciones de los servicios**. El término **orquestación** significa *secuenciar los servicios y proveer la lógica adicional para procesar datos*. Los servicios están enlazados a flujos de trabajo, y por lo tanto actúan como una sola aplicación. Aquí se utilizan herramientas visuales (ej: el diseñador de orquestación de Biztalk Server²⁵, usando el lenguaje BPEL²⁶).

Una capa opcional es la **Capa 5 de Capa de Presentación**. Los usuarios acceden a los servicios y a las orquestaciones invocando desde diversas interfaces de usuario la funcionalidad que desean consumir.

A modo transversal se presenta las capas 6 y 7 [Rojas, 2009]:

- **Capa 6: Integración (ESB-Enterprise Service Bus).** Esta capa **facilita la integración de servicios** a través de la introducción de un conjunto de capacidad tales como ruteo, mediación de protocolos, mecanismos de transformación, etc. Con uso de WSDL se especifica el binding²⁷, el cual implica la localización del servicio que se provee. ESB posibilita la **independencia de la ubicación del servicio para su integración**, ya que el mismo es el que finalmente controla el ruteo de los mensajes que llegan para ser procesados.
- **Capa 7: Administración, Monitoreo y Calidad del Servicio.** Esta capa brinda las características requeridas para monitorear, administrar y mantener la calidad del servicio en áreas tales como seguridad, desempeño, y disponibilidad. Esto se conoce como el **gobierno de SOA**.

En la figura 6.5 se puede ver como se estructuran las mismas.

²² CRM: Customer Relationship Management.

²³ ERP: Enterprise Resource Planning.

²⁴ BI: Business Intelligence.

²⁵ Biztalk Server: plataforma de integración de procesos de negocio desarrollado por Microsoft Corporation.

²⁶ BPEL: Business Process Execution Language.

²⁷ Binding: elemento de la estructura de WSDL que especifica los protocolos de comunicación usados.

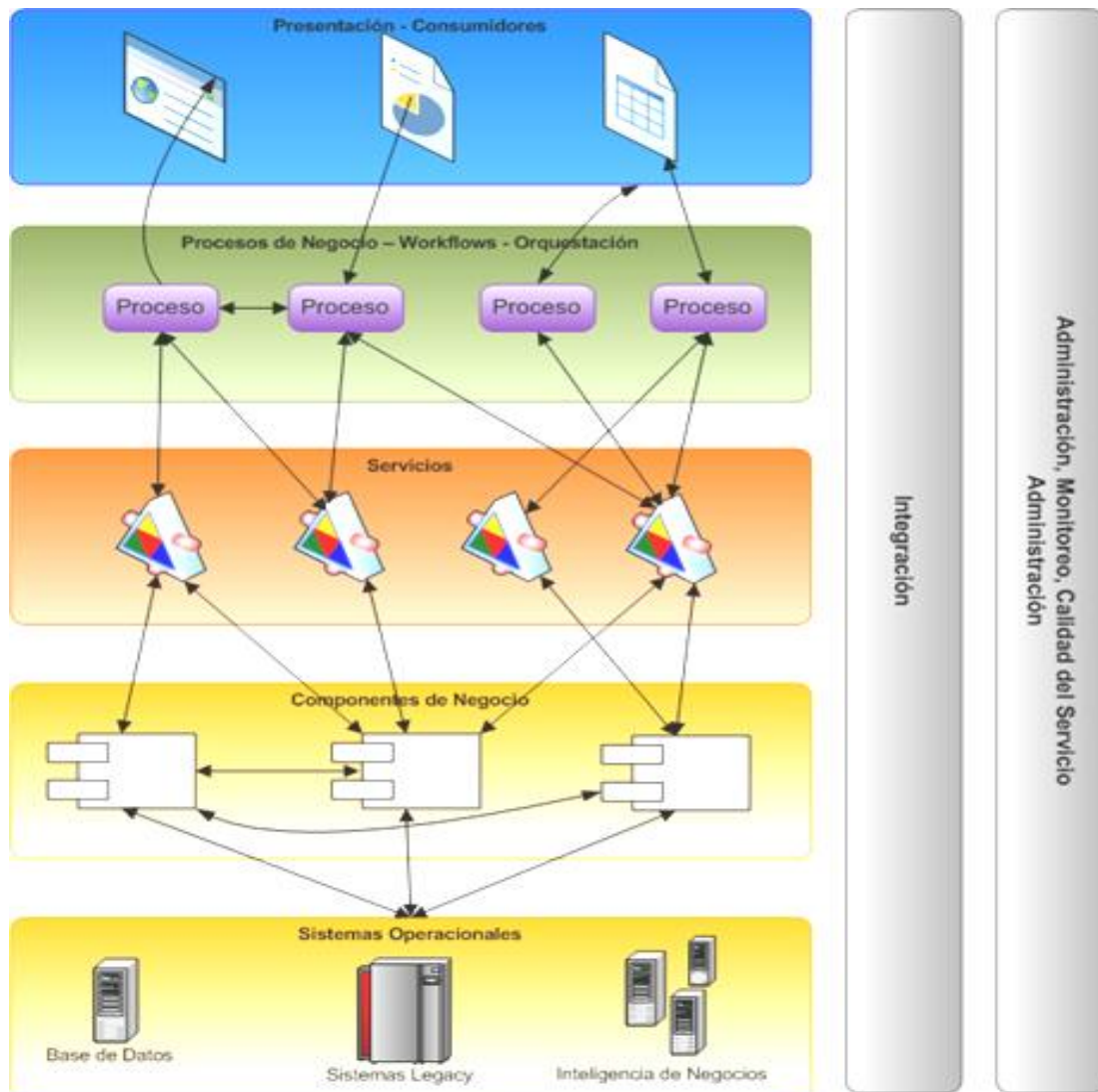


Figura 6.5: Modelo conceptual de SOA. Fuente: Rojas D., 2009.

Cada proveedor de soluciones (IBM, Oracle, BEA, etc.) tiene su **propia Arquitectura SOA de Referencia**, que incorpora sus herramientas específicas. Pero se puede hacer referencia a componentes comunes como (Fig.6.6):

- **Usuarios de Negocio**, que utilizan distintas tecnologías para acceder a la aplicación (o proceso de negocio), tales como Desktop, Notebooks, Dispositivos móviles, etc.
- **Aplicación SOA y Portal**, que se implementan mediante la utilización de componentes como son los Portlets y Servicios. Se destaca el concepto de reusabilidad. Estos componentes garantizan el desarrollo de una aplicación orientada a Servicios implementada tanto en computadores como en dispositivos móviles. Por ejemplo www.soa.com.
- **Servicios de Presentación (Portlets)**: Son los componentes realmente reutilizables. Ejemplos: un portlet de “Calendario”, un portlet para mostrar las “Publicaciones Recientes” de un blog, “Lista de pendientes” para el caso de los “Procesos de Negocio” (BPMS).
- **Procesos de Negocio**: En esta etapa se implementan las técnicas de desarrollo de aplicaciones SOA, como BPM (Businessman Process Modeling), que permite identificar las tareas interactivas (interacción participante), con actividades automatizadas (servicios), logrando un mejor proceso de negocio. Ejemplo: el proceso de “publicar un comentario en un Blog”, que dentro de sus tareas interactivas esta el “ingresar el comentario”, y “aprobar el comentario para su publicación”, y una

actividad automatizada es el servicio de “ingresar el comentario en el sistema de Blog”.

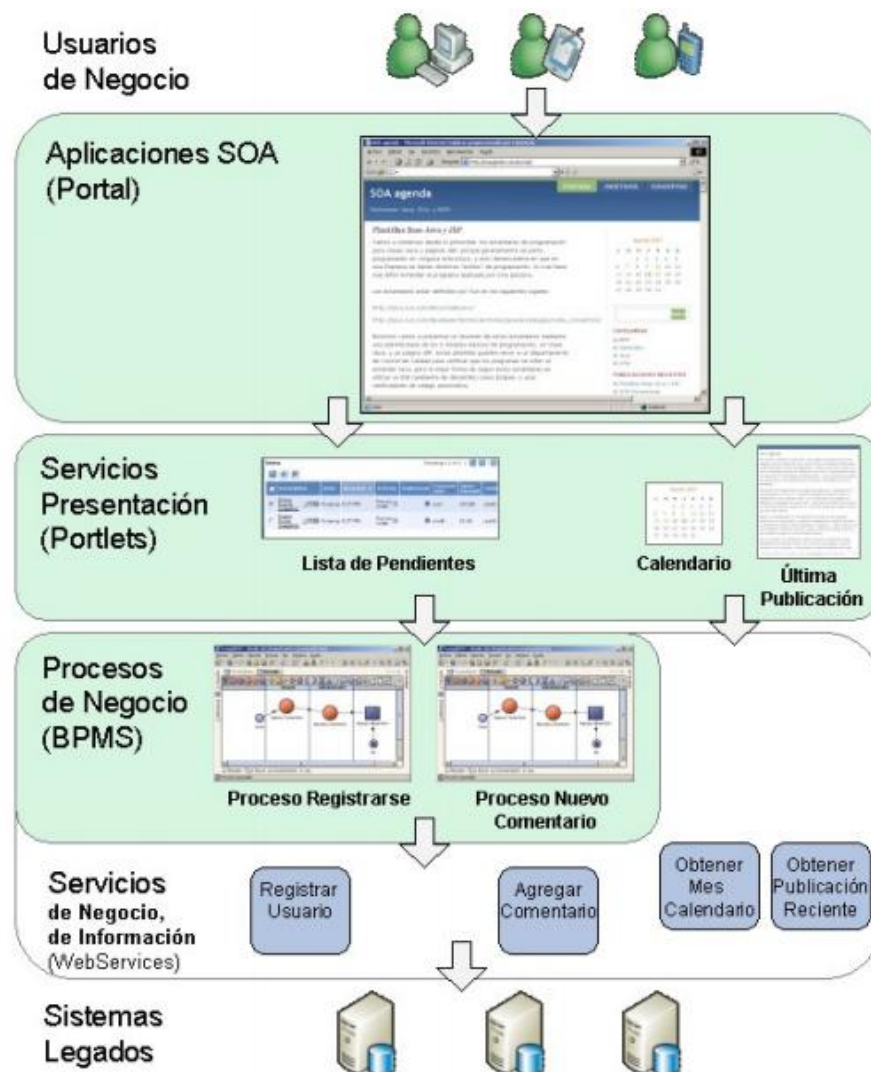


Figura 6.6: Componentes del Modelo conceptual de SOA. Fuente: www.soagenda.com.

6.4 - Diseño y desarrollo de aplicaciones SOA

Principios de diseño

Aunque no hay estándares que definen una composición exacta de SOA, sin embargo es posible distinguir 8 principios fundamentales de diseño [Erl, 2008]:

- **Contrato de servicio estandarizado:** Los servicios dentro del mismo inventario de servicios cumplen con las mismas normas de diseño del contrato.
- **Bajo acoplamiento:** Los contratos de servicio imponen requisitos bajos de acoplamiento de consumidores y se desacoplan de su entorno.
- **Abstracción:** Los contratos de servicios solo contienen información esencial y la información sobre los servicios se limita a lo que se publica en los contratos de servicios.
- **Reusabilidad:** Los servicios contienen y expresan lógica agnóstica y se pueden posicionar como recursos empresariales reutilizables.
- **Autonomía:** Los servicios ejercen un alto nivel de control sobre su entorno de ejecución de tiempo de ejecución subyacente.

- **Sin estado:** Los servicios minimizan el consumo de recursos al diferir la administración de la información del estado (sesiones) cuando es necesario.
- **Garantía de su descubrimiento:** Los servicios se complementan con metadatos comunicativos mediante los cuales se pueden descubrir e interpretar de manera efectiva.
- **Preparado para ser usado en composiciones:** Los servicios son participantes efectivos en la composición, independientemente del tamaño y la complejidad de la composición.

Como se ve, todos los Principios de la Orientación a Servicios están interrelacionados (Fig. 6.7).

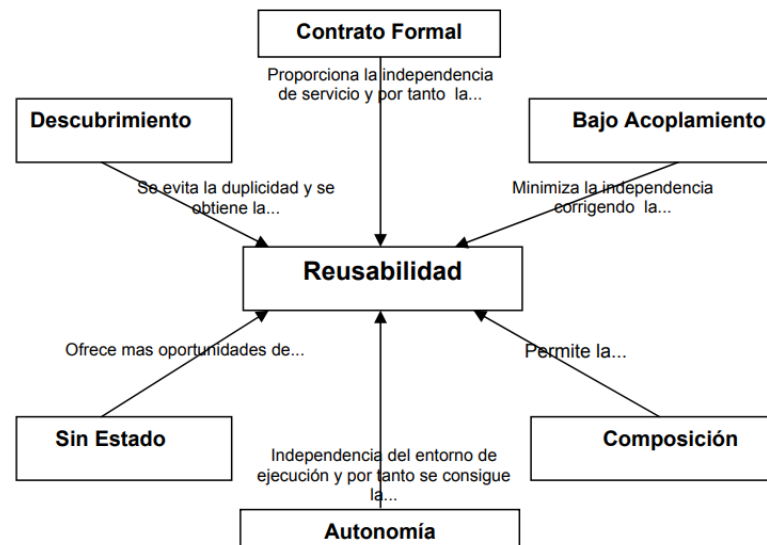


Figura 6.7: Principios de Diseño SOA. Fuente: Antonio Barco, 2006.

Metodología de análisis y diseño orientado a servicios

La **metodología de análisis y diseño orientado a servicios** para aplicaciones SOA ofrece un marco de trabajo tanto para el desarrollo de software como para la implementación. El Proyecto SOA y el ciclo de vida general de los servicios pueden relacionarse con las siguientes etapas comunes y primarias [Erl, 2005]:

1. Planificación de adopción de SOA
2. Análisis de inventario de servicio
3. **Análisis orientado a servicios (modelado de servicios)**
4. **Diseño orientado a servicios (contrato de servicio)**
5. Diseño de lógica de servicio
6. Desarrollo de servicios
7. Pruebas de servicio
8. Despliegue y mantenimiento del servicio
9. Uso y monitoreo del servicio
10. Descubrimiento de servicios
11. Versiones y retiro de servicios.

De estas etapas, la aplicación propiamente de la **Orientación a Servicios** se distingue en las etapas 3 y 4 [Arcitura, 2018]. El **análisis orientado a servicios** es un proceso que inicia con la recopilación completa de información preparatoria para ejecutar el modelado de servicios, identificando, clasificando, agrupando y componiendo servicios candidatos por medio de la descomposición de los procesos de negocio. Se realiza de forma iterativa, la

entrada es un proceso de negocio y la salida puede ser candidatos conceptuales de servicio, candidatos de capacidad de servicio y candidatos de composición de servicio.

La etapa de **diseño orientado a servicios** está dedicada a la producción de contratos de servicios como respaldo del enfoque establecido en el *contrato primero* para el desarrollo de software. La entrada es un servicio candidato (resultado de la etapa de análisis), es sometido a consideraciones adicionales que lo conforman en un contrato de servicio técnico (salida) en alineación con otros contratos de servicio que se producen para el mismo inventario de servicio. Para ello, se basa en los 8 principios fundamentales de diseño. Parte del proceso de conformación de los contratos de servicio incluye la creación del **Acuerdo de Nivel de Servicio** (SLA: Service Level Agreement) como parte del contrato. Dicho **acuerdo** define formalmente el nivel del servicio, involucrando la disponibilidad, el performance y la seguridad. Este puede ser especialmente importante para los servicios basados en la nube que se ofrecen a una base de consumidores más amplia a través del modelo de entrega en la nube de **SaaS**²⁸.

6.5 – Tecnologías y Estándares en SOA

Para la implementación de un sistema sobre la arquitectura planteada por SOA es necesario tener herramientas para la descripción y la ejecución de los servicios establecidos [Álvarez Ballestas y Rodelo Calvo, 2008]. Los estándares que se aplican a SOA relacionados con los servicios web son SOAP, WSDL, otros. El estándar relacionado con la ejecución de los procesos de negocio es BPEL (Business Process Execution Language). Los componentes tecnológicos usados son:

- ESB - Bus de Servicios, donde se despliegan y ejecutan los servicios.
- Registro de servicios, basado en el protocolo UDDI
- BPM (Business Process Management): componente para la orquestación de servicios en procesos de negocio.
- BAM (Business Activity Monitoring): componente para la visualización y el seguimiento de las actividades del negocio.

Los Protocolos utilizados son: HTTP, URL, XML, otros.

WS-CDL

WS-CDL (Web Services-Choreography Description Language) es un Protocolo que tiene como principal objetivo **la descripción del comportamiento de cada uno de los servicios establecidos para lograr una finalidad**, es un protocolo utilizado para la definición de servicios dentro de la plataforma SOA, basado en XML.

WS-CDL es importante dentro de SOA porque es una **tecnología escalable** y garantiza la **interoperabilidad efectiva y segura de servicios**. Permite tener servicios más robustos reduciendo el tiempo de implementación de los mismos. Esto es posible debido a que el lenguaje permite la descripción de las colaboraciones establecidas entre servicios de manera clara, sin ambigüedades. Es un protocolo de negociación, de tal forma que cada organización desarrolle de manera independiente su propio rol en el desarrollo de la aplicación, respetando el contrato global para que se garantice la interoperabilidad.

WS-CDL también es un **lenguaje organizado por capas**, que permiten diferentes niveles de expresión de las coreografías de un servicio. En el nivel más alto, existe un paquete que contiene todas las definiciones realizadas por WS-CDL. Estas coreografías deben incluir

²⁸ SaaS: Software as a Service. Software cuyo soporte técnico y datos se alojan en servidores de una compañía de TIC y a los cuales accede vía internet desde un cliente.

como mínimo un conjunto de roles definidos por ciertos comportamientos, una serie de relaciones entre dichos roles, canales utilizados por los roles para interactuar y un bloque de coreografías utilizado por los canales para definir la interacción. En este nivel, se describe un conjunto básico de conexiones de servicios que permiten la colaboración entre roles para lograr un objetivo; sin embargo es posible adicionar una composición estructurada, permitiendo la combinación en secuencias o actividades paralelas de las interacciones y otras coreografías [Álvarez Ballestas y Rodelo Calvo, 2008].

WS-BPEL

WS-BPEL (Web Services-Business Process Execution Language) es un lenguaje basado en la orquestación, es decir, que permite la definición de servicios, a partir de servicios ya existentes, nace de la necesidad de integrar o engranar las diversas tecnologías que funcionan bajo el ambiente de SOA. Surge como la combinación de WSFL (Web Services Flow Language), desarrollado por IBM, y de XLANG, desarrollado por Microsoft.

WS-BPEL es visto como un lenguaje basado en un flujo de trabajo extensible, el cual permite la agregación de servicios de manera recursiva y en un entorno altamente dinámico; esto favorece el cambio frecuente de servicios, desarrollando procesos desacoplados para que se adapten de manera adecuada a instancias particulares de servicios en una coreografía específica.

6.6 – SOA y los Microservicios

Los microservicios son uno de los métodos de desarrollo de software que más ha crecido en los últimos tiempos, sobre todo para la construcción de sistemas distribuidos. Esencialmente, una **arquitectura de microservicios** (MSA: Micro-Services Architecture) es un **método para desarrollar aplicaciones de software como un conjunto de servicios modulares desplegables de forma independiente**, en los que cada servicio ejecuta un proceso único y se comunica a través de la red con un mecanismo claro y bien definido para servir a un objetivo del negocio.

Una forma de entender a SOA en relación a los microservicios, es imaginando un **patrón de arquitectura de software**, cuyos componentes de aplicación proporcionan servicios a otros componentes, a través de un protocolo de comunicaciones que emplea una red. Estas comunicaciones pueden variar en complejidad e ir desde, la simple transferencia de datos hasta la coordinación de varios servicios para establecer su conexión.

Por su parte, los microservicios son también **un patrón de arquitectura de software**, aunque en este caso, está construido en base a aplicaciones complejas y los pequeños procesos independientes que las componen, comunicándose entre sí mediante APIs²⁹ agnósticas del lenguaje.

Aunque se encuentran similitudes entre MSA y SOA, existen varias diferencias entre las que cabe destacar las siguientes:

- **Compartición de componentes:** MSA minimiza el uso compartido de componentes a través de un contexto limitado. En cambio, SOA lo aprovecha aunque puede aumentar las latencias, haciendo que los sistemas basados en este tipo de arquitectura sean más lentos.

²⁹ API: Application Programming Interface

- **Granularidad de servicio:** SOA ofrece servicios más versátiles en cuanto a su funcionalidad empresarial, mientras que los microservicios se componen de servicios con un único propósito y muy especializados.
- **Coordinación:** los Microservicios generalmente carecen de coordinación entre sí, o una coordinación mínima. En SOA, es necesario coordinar con varios grupos para atender a las solicitudes de servicio.
- **Middleware:** esta opción de mensajería es la empleada por SOA, que ofrece en base a ella una serie de capacidades adicionales que no se encuentran en MSA, incluyendo mediación y enrutamiento, mejora de mensajes, mensajes y transformación de protocolos. Por su parte, los microservicios trabajan con una capa de API creada entre los servicios y los consumidores de servicios.
- **Interoperabilidad heterogénea:** SOA promueve la propagación de múltiples protocolos heterogéneos a través de su componente middleware de mensajería. Esta opción se debe considerar en los casos cuyo objetivo sea lograr la integración de varios sistemas utilizando diferentes protocolos en un entorno heterogéneo. Sin embargo, es preferible considerar los microservicios cuando todos los servicios puedan quedar expuestos y deban usar el mismo protocolo de acceso remoto, ya que MSA intenta simplificar el patrón de arquitectura al reducir el número de opciones de integración [PowerData, 2017].

6.7 - Beneficios de la implementación de SOA

La reutilización de software posibilita el incremento de la agilidad organizativa, es decir, la capacidad de las organizaciones de responder al cambio.

La independencia de las plataformas e infraestructuras tecnológicas, por lo cual se pueden integrar sistemas y aplicaciones diferentes de manera sencilla, genera flexibilidad y permite la reutilización de las tecnologías existentes.

La creación de sistemas **altamente escalables** posibilita la expansión y consolidación empresarial, ya que proporciona una metodología y un marco de trabajo para documentar las capacidades de negocio y dar soporte a las actividades de integración.

El **aseguramiento de la Calidad del Software** (SQA³⁰) es una parte esencial del desarrollo y mantenimiento del software.

³⁰ SQA: Software Quality Assurance



El número de personas que se conectan a internet cada vez es mayor.

Los negocios internacionales han adoptado las telecomunicaciones como soluciones empresariales para volverse más dinámicos e innovadores en relación a los avances tecnológicos.

CAPÍTULO 7: ESCALABILIDAD

7.1 - Conceptualizando la escalabilidad

Según el Diccionario Cambridge, la escalabilidad se define como la capacidad de un negocio o sistema para crecer más grande. El término escalable es utilizado para describir un negocio o sistema capaz de crecer o hacerse más grande, es decir, la capacidad de un sistema informático de cambiar su tamaño o configuración para adaptarse a las circunstancias cambiantes.

La escalabilidad está ligada al diseño del sistema, influye en su rendimiento de manera significativa. Si una aplicación está bien diseñada, la escalabilidad no constituye un problema. La red puede expandirse rápidamente para admitir nuevos usuarios y aplicaciones sin afectar el rendimiento del servicio enviado a los usuarios actuales.

La escalabilidad puede medir en diferentes dimensiones:

- **Escalabilidad en carga:** Un sistema distribuido facilita la ampliación y reducción de sus recursos para acomodar las cargas más pesadas o más ligeras según se requiera.
- **Escalabilidad geográfica:** Un sistema geográficamente escalable, es aquel que mantiene su utilidad y usabilidad, sin importar que tan lejos estén sus usuarios o recursos.
- **Escalabilidad administrativa:** No importa qué tantas diferentes organizaciones necesiten compartir un solo sistema distribuido, debe ser fácil de usar y manejar.

La escalabilidad cuenta como factor crítico **el crecimiento de usuarios**. Es mucho más sencillo diseñar un sistema con un número constante de usuarios que diseñar un sistema con un número creciente y variable de usuarios. Si un sistema tiene como objetivo **crecer en el número de usuarios** **mantenimiento el rendimiento actual**, tiene que analizar dos opciones:

- Con un hardware de mayor potencia
- Con una mejor combinación de hardware y software.

El tipo de escalabilidad puede ser (Fig. 7.1):

- **Vertical o escalar hacia arriba:** donde al añadir más recursos a un nodo particular del sistema, éste mejora en conjunto. Por ejemplo: agregar un disco rígido de mayor velocidad y capacidad de almacenamiento mejora el rendimiento de todo el equipo.
- **Horizontal:** donde al agregar más nodos al sistema, el rendimiento de éste mejora. Por ejemplo: incorporar una computadora nueva a un sistema y se distribuya en ella la carga de una computadora existente, puede llegar a mejorar el rendimiento de todo el sistema.

Ventajas de la escalabilidad vertical:

- No implica un gran problema para las aplicaciones, pues todo el cambio es sobre el hardware

- Es mucho más fácil de implementar que el escalamiento horizontal.
- Puede ser una solución rápida y económica (compara con modificar el software)
- En la escalabilidad vertical, el límite lo pone el hardware; en la horizontal, no existen limitaciones de crecimiento, pero sí supone una gran modificación en el diseño y luego tareas de reimplantación.

Desventajas de la escalabilidad vertical:

- El crecimiento está limitado por el hardware.
- Una falla en el servidor implica que la aplicación se detenga.
- No soporta la Alta disponibilidad.
- Hacer un upgrade del hardware al máximo pues llegar a ser muy caro, ya que las partes más nuevas suelen ser caras con respecto al rendimiento de un modelo anterior.

Ventajas de la escalabilidad horizontal:

- El crecimiento es prácticamente infinito, podríamos agregar cuantos servidores sean necesarios
- Es posible combinarse con el escalamiento vertical.
- Soporta la alta disponibilidad
- Si un nodo falla, los demás sigue trabajando.
- Soporta el balanceo de cargas.

Desventajas de la escalabilidad horizontal:

- Requiere de mucho mantenimiento
- Es difícil de configurar
- Requiere de grandes cambios en las aplicaciones (si no fueron diseñadas para trabajar en cluster)
- Requiere de una infraestructura más grande.



Figura 7.1: Escalabilidad vertical y horizontal.

Balance de carga

A la hora de diseñar un sistema que comparte recursos, es necesario considerar el **balance de la carga de trabajo**. Este concepto hace referencia a la técnica usada para dividir el trabajo a compartir entre varios procesos, computadoras, u otros recursos. Para evitar los cuellos de botella, el balance de la carga de trabajo se reparte de forma equitativa a través de un algoritmo que estudia las peticiones del sistema y las redirecciona a la mejor opción. El balance de carga puede ser por hardware y o software.

7.2 - Internet, una Red Escalable

La mayor red conocida es Internet. Desde sus inicios se ha expandido aceleradamente sin afectar seriamente el rendimiento de los usuarios individuales. Esto es el resultado del diseño de protocolos y tecnologías subyacentes sobre la cual se construye. **Internet es altamente escalable**. Los nuevos protocolos y estructuras de direccionamiento están en desarrollo para cumplir con el ritmo acelerado al cual se agregan los servicios y aplicaciones de internet.

Internet, conformada por una colección de redes públicas y privadas interconectadas, tiene una estructura jerárquica en capas para servicios de direccionamiento, designación y conectividad. En cada nivel o capa de la jerarquía, **los operadores de red individual mantienen relaciones entre pares con otros operadores en el mismo nivel**. Como resultado, el tráfico de redes destinado a servicios regionales y locales no necesita cruzar a un punto central para su distribución. Los servicios comunes se pueden duplicar en diferentes regiones, por ello mantienen fuera el tráfico de las redes backbone de alto nivel.

Aunque no hay una organización única que regule Internet, **los operadores de muchas de redes individuales que proporcionan conectividad a Internet cooperan para seguir estándares y protocolos aceptados [CCNA, 2015]. La adopción de estándares, permite a fabricante de hardware y software proveer de productos que pueden integrarse a infraestructuras existentes.**

En el nivel 1 (Fig. 7.2), se encuentran los **proveedores de Internet** (ISP: Internet Service Provider), que brindan conexiones nacionales e internacionales. Algunos ejemplos son: Verizon, Sprint, AT&T, NTT, sistemas de cable y redes inalámbricas de área amplia.



Figura 7.2: Nivel 1 – los ISP1 y el backbone de Internet. Fuente: CCNA, 2015.

En el nivel 2 (Fig. 7.3), se encuentran los ISP más pequeños (ISP2), que generalmente brindan un servicio regional. Estos proveedores les pagan a los ISP de nivel 1 por la conectividad con el resto de Internet. El punto donde se interconectan ambos tipos de proveedores se denomina “límite”.

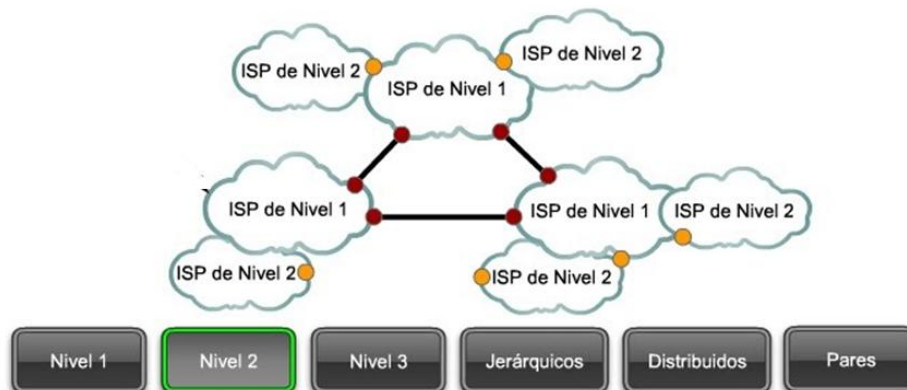


Figura 7.3: Nivel 2 – los ISP2. Fuente: CCNA, 2015.

En el nivel 3 (Fig. 7.4), se encuentran los proveedores de servicio local (IPSlocal) que directamente brindan el servicio a los usuarios finales. Estos proveedores están conectados a los ISP2 y les pagan a los ISP2 para acceder a Internet.

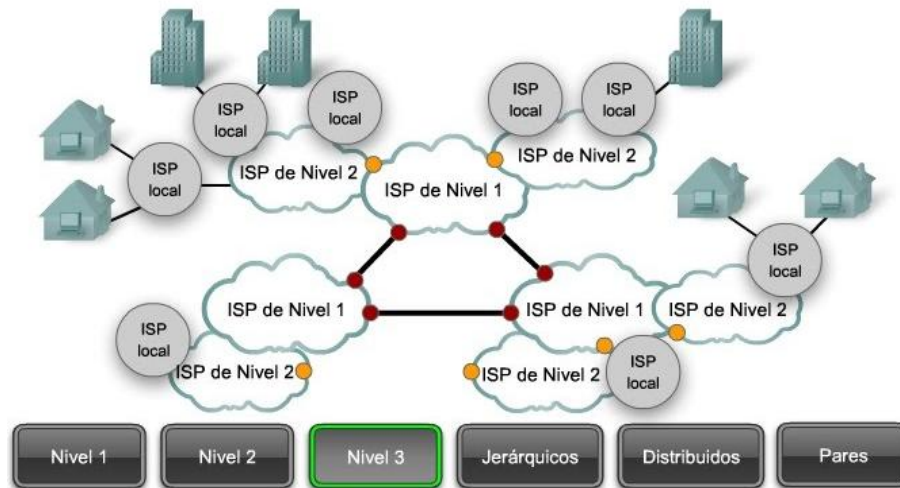


Figura 7.4: Nivel 2 – los ISP2. Fuente CCNA, 2015.

En el siguiente nivel (Fig. 7.5), se encuentra el Sistema de Nombre de Dominio (DNS: Domain Name System). Este proporciona un directorio jerárquico de direcciones, es decir, un servidor no necesita guardar la lista completa de millones de direcciones.

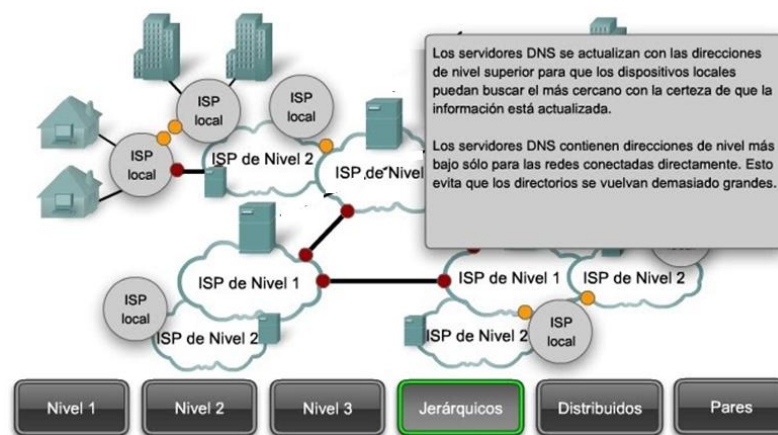


Figura 7.5: El DNS. Fuente CCNA, 2015.

La naturaleza distribuida de las conexiones y los directorios significa que la comunicación puede evitar los cuellos de botella y las interrupciones (Fig. 7.6). Originalmente diseñado para protegerlo de los ataques militares, el sistema ha demostrado ser la mejor manera de ofrecer una red civil confiable y escalable.

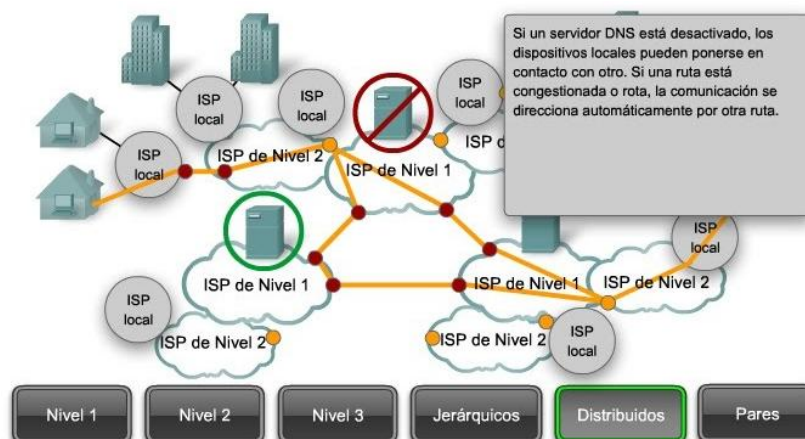


Figura 7.6: la conectividad distribuida. Fuente: CCNA, 2015.

La conexión de pares entre redes que se encuentran en el mismo nivel brinda conexiones directas y evitan así rutas más largas y la congestión en el backbone (Fig. 7.7).

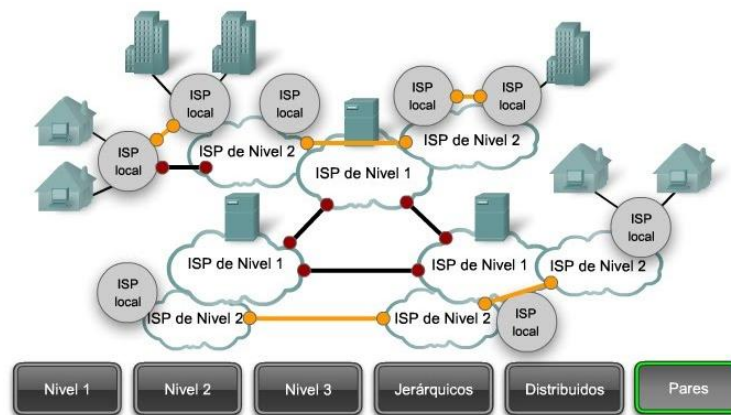


Figura 7.7: la conectividad entre pares. Fuente CCNA, 2015.

Internet tuvo un problema de escalabilidad con las direcciones IPv4 de la forma xxx.xxx.xxx.xxx (por ejemplo: 190.191.74.50). Para evitar este límite se crearon las direcciones IPv6 que posibilita 340.282.366.920.938.463.374.607.431.768.211.456 de direcciones. Actualmente en internet conviven ambas tecnologías; crear el IPv6 fue una forma de lograr la escalabilidad de la gran red de redes.



“La convergencia de la radiodifusión, la informática y las telecomunicaciones está cambiando la naturaleza de los servicios y de las redes por las que éstos se ofrecen. La convergencia se está viendo propiciada en gran medida por la transición hacia las NGN, sobre una nueva arquitectura basada en IP”

ITU

CAPÍTULO 8: REDES DE SIGUIENTE GENERACIÓN

8.1 - Red NGN

Las **Redes de Siguiete Generación** (NGN: Next Generation Networks) son **redes multiservicios capaz de apoyar la voz, datos y video**. Se caracterizan por la separación del **plano de control** (señalización, control) y el **plano de transporte** (de la conmutación). Utilizan tecnología en modo de paquete al transporte (IP³¹) para toda clase de información e interfaces abiertas entre el transporte, control y los usos de estándares.

Los objetivos de esta nueva tecnología **son: la flexibilidad para la construcción de una plataforma multiservicios integrados**, la reducción de costos, el compartimiento de la infraestructura y sistemas de comunicación unificados que hace práctico su despliegue.

Con el apoyo de **IMS** (IP Multimedia Subsystem), NGN trata de unificar todos los servicios sobre la misma red. La unificación como tal conlleva a redes convergentes de servicios y de infraestructura. Lo que es un beneficio a la dificultad de pasar de una red a otras.

El modelo vertical del pasado con redes y servicios estrechamente ligados era un modelo de red monoservicio. La nueva generación pretende un **modelo multiservicio**, es decir, **un modelo horizontal con independencia absoluta entre red y servicio y con una única solución de red común a todos ellos**.

Dos ventajas de esta propuesta son: **una gestión y tarificación común para todos los servicios, y más económico para el usuario y para el proveedor**. Una desventaja del modelo es que favorece una arquitectura monopolista.

El surgimiento de las redes NGN, con inteligencia incorporada, es parte del fenómeno de convergencia. El surgimiento de NGN está impulsando un cambio de paradigma en las telecomunicaciones. Entre los rasgos esenciales se encuentran:

- redes IP o redes basadas en paquetes (incluyendo la migración de PSTN³² basada en circuitos a NGN basada en paquetes)
- separación de los servicios de aplicación de la red de transporte
- convergencia o redes integradas de banda ancha
- e inteligencia distribuida en la red.

SDH

SDH (Synchronous Digital Hierarchy) o **Jerarquía Digital Sincrónica**, se puede considerar como la revolución de los sistemas de transmisión, como consecuencia de la

³¹ IP: Internet Protocol: Protocolo de Internet

³² PSTN: Public Switched Telephone Network: Red de Telefonía Pública Conmutada

utilización de la fibra óptica como medio de transmisión, así como de la necesidad de sistemas más flexibles y que soporten anchos de banda elevados. SDH se desarrolló en EEUU bajo el nombre de SONET y posteriormente el CCITT³³ en 1989 publicó una serie de recomendaciones donde quedaba definida con el nombre de SDH [ADSLFAQS, 2018].

La trama básica de SDH es el STM-1 (Synchronous Transport Module level 1), con una velocidad de 155 Mbps. Cada trama va encapsulada en un tipo especial de estructura denominado **contenedor**. Las tramas contienen información de cada uno de los componentes de la red, trayecto, línea y sección, además de la información de usuario. Los datos son encapsulados en contenedores específicos para cada tipo de señal tributaria. A éstos se añaden cabeceras de control, y luego de un proceso de multiplexación, se integra dentro de la estructura STM-1 en varios niveles. A estos contenedores se les agrega información adicional (Path overhead) conformando los denominados contenedores virtuales o VC (Fig. 8.1).

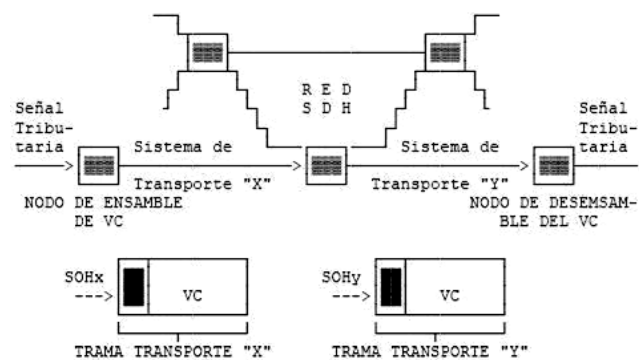


Figura 8.1: Trama con SDH. Fuente: ADSLFAQS, 2018.

SDH no sólo sirve para transportar canales de voz sino que ha ido incorporando capacidades de transporte de tráfico ATM³⁴, IP y Ethernet. La **flexibilidad de ancho de banda** y el **uso más eficiente de recursos** son algunos de los beneficios. Se caracteriza también por el **ajuste de capacidad a las necesidades de tráfico** y no exige disponibilidad de recursos contiguos por lo que es transparente a equipos intermedios antiguos y tiene mejor comportamiento frente a cortes.

Evolución hacia una red de Transporte de Próxima Generación

La tecnología de **Multiplicación por Longitud de Onda (WDM)** se ha desarrollado en los últimos años alcanzando velocidades de hasta 1.6 Tbps.

En la medida en que el tráfico se concentre más y más en servicios IP dinámicos, es indispensable contar con una infraestructura flexible de la Red Óptica. La tendencia es hacia la **introducción de soluciones de conmutación y otras funciones al nivel de longitud de onda, con un mínimo de transformaciones al nivel eléctrico**.

En este contexto surge el concepto de **Red Óptica de Conmutación Automática (ASON: Automatic Switched Optical Network)** para **transformar la red estática en una red de conmutación automática**. *Permite la asignación dinámica de los recursos a las rutas, suministrar y desarrollar servicios de forma rápida, instalación de los servicios en menos tiempo, costos bajos de operación y mantenimiento, recuperación rápida por fallas, entre*

³³ CCITT: Consultative Committee for International Telegraphy and Telephony: Comité Consultivo Internacional Telegráfico y Telefónico

³⁴ ATM: Asynchronous Transfer Mode: Modo de Transferencia Asíncrona

otras. Para lograr la operación de redes ópticas aun quedan por resolver diversos problemas de hardware, software y estandarización.

En el nuevo modelo de red, la **capa de transporte incluye las funcionalidades de transmisión, conmutación y enrutamiento de paquetes en la red**. En ella se distinguen tres subcapas: **acceso, transporte y red de cliente**. Se está dando un gran crecimiento del tráfico de datos y la demanda de ancho de banda crece, y continuará creciendo, motivada sobre todo por el aumento de los servicios IP y esto debe corresponderse con un crecimiento en la capacidad de las redes.

Actualmente se han desarrollado extensiones del protocolo SDH para solucionar algunos de sus inconvenientes para el transporte de datos como por ejemplo:

- GFP (Generic Framing Protocol), que es un protocolo que estandariza el empaquetado de datos en tramas SDH/SONET. Es superior a POS (Packet over SONET).
- LCAS (Link Capacity Adjustment Scheme), un mecanismo que permite multiplexar varios flujos de datos sobre una trama SONET/SDH de velocidad superior.
- VCAT (Virtual Concatenation), que agrupa múltiples VCs para ser enrutados independientemente por fibras y nodos distintos. La combinación LCAS y VCAT es una **herramienta para el ajuste del ancho de banda en demanda**.
- Especificación de la interfaz STM-256 (40 Gbps).

Con **GMPLS (Generalized Multi-Protocol Label Switching)**, un estándar del IETF, se han desarrollado extensiones de los protocolos tradicionales de MPLS³⁵ para posibilitar el trabajo con etiquetas que representan: paquetes IP, celdas ATM, ranuras de tiempo de multiplexado, longitud de onda y espacio.

En el modelo SDH, **WDM** es una capa puramente física para aumentar la capacidad y el enrutamiento y conmutación se realizan a nivel de longitudes de onda y utilizan componentes como los OXC fotónicos o eléctricos (Fig. 8.2).

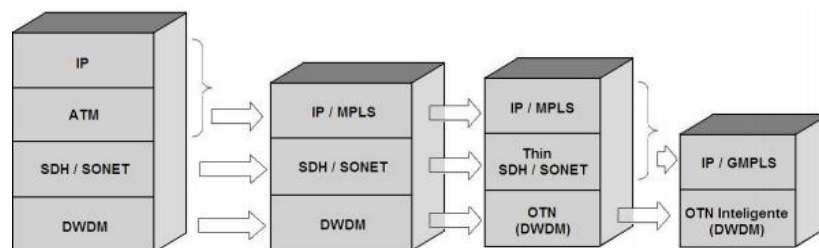


Figura 8.2: Evolución de las redes de transporte.

De este modo la arquitectura nueva de la red cumple con las características de ASON en cuanto a protecciones, descubrimiento de la red y facilidad de gestión, mejorando el uso del ancho de banda al no usar ATM y sobre todo por la nueva filosofía implícita en las soluciones de protección.

8.2 - Arquitectura NGN

La arquitectura de una red NGN debe contemplar Interfaces y protocolos abiertos basados en estándares y normas a fin de ofrecer interoperabilidad de productos de distintos proveedores y una arquitectura de control con señalización central, lo cual ayuda a reducir los costos de ejecución. Debe trabajar con servicios adaptables que puedan crearse fácil y

³⁵ MPLS: Multi-Protocol Label Switching; Conmutación de Etiquetas Multiprotocolo.

rápidamente y ser desplegados en toda la red y convivir con servicios existentes provenientes de redes anteriores.

La arquitectura NGN tiene las siguientes capas (Fig. 8.3):

- **Capa de Acceso:** incluye las tecnologías para conectar los clientes finales, se incluyen líneas de cobre y conexiones convencionales. Se presentan también tecnologías como sistemas de cable, xDSL³⁶, inalámbricos y anillos MetroEthernet.
- **Capa Core:** contiene el transporte y control de señalización basada en tecnología de paquetes IP, utilizando ATM, MPLS y Ethernet. En el borde se usan **Gateways para conexión con otras redes (TGW)** o bien **con los equipos del cliente (AGW)**.
- **Capa de Gestión:** es una capa esencial en NGN que proporciona funciones de dirección empresarial, de los servicios y de la red. Permite la provisión, supervisión, recuperación y análisis del desempeño para dirigir la red.
- **Capa de Servicio:** incluye equipamiento para proporcionar los servicios y aplicaciones a la red. Los servicios deben ser independientes de la tecnología a utilizar. Se colocan generalmente de forma centralizada a fin de lograr mayor eficiencia y además distribuirlos a la red. Los tipos de servicios deben abarcar los ya existentes además una gama de servicios de datos y servicios multimedia en cualquier combinación posible.

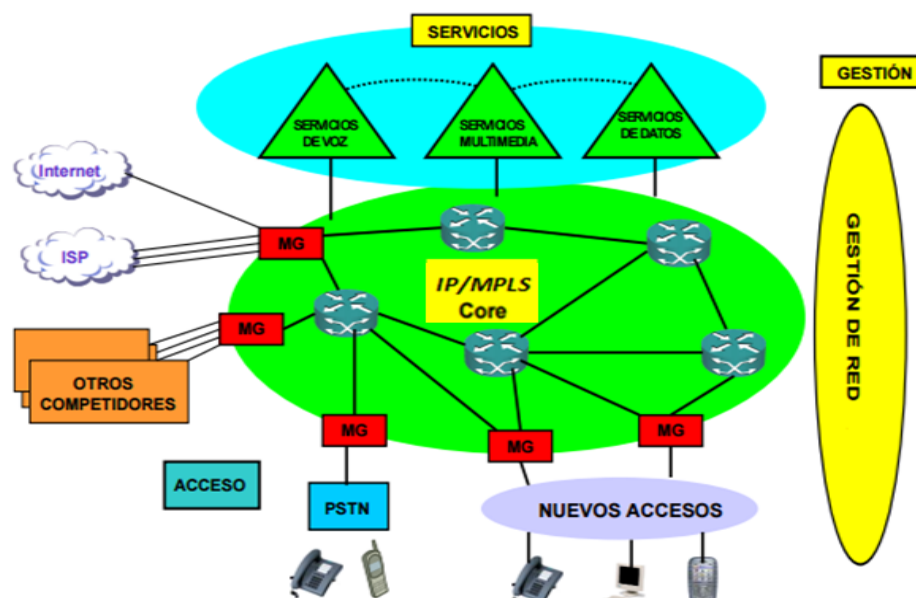


Figura 8.3: Arquitectura NGN. Fuente: Rico-Bautista, 2017.

8.3 - Aplicación de ASON

ASON como un tipo de arquitectura de red definida por la ITU-T³⁷ en la recomendación G.8080 define diferentes interfaces: I-NNI (internal network interface), E-NNI (connections between networks), UNI (User to Network interface), NMI (Management – Network interface) y CCI (conexiones entre el plano de Control y los Elementos Ópticos) (Fig. 8.4).

³⁶ xDLS: hace referencia al conjunto de tecnologías que proveen un gran ancho de banda permitiendo el flujo de información tanto simétrico como asimétrico de alta velocidad. Algunas pueden ser: IDSL, ADSL, HDSL, SDSL, etc.

³⁷ ITU-T: International Telecommunications Union. Unión Internacional de Telecomunicaciones.

ASON permite definir caminos por demanda usando reglas de implementación generales, determinadas por el operador en base: al tipo de servicio, QoS³⁸ que se pretende brindar, costos de los enlaces, cantidad de saltos, retardos, etc.

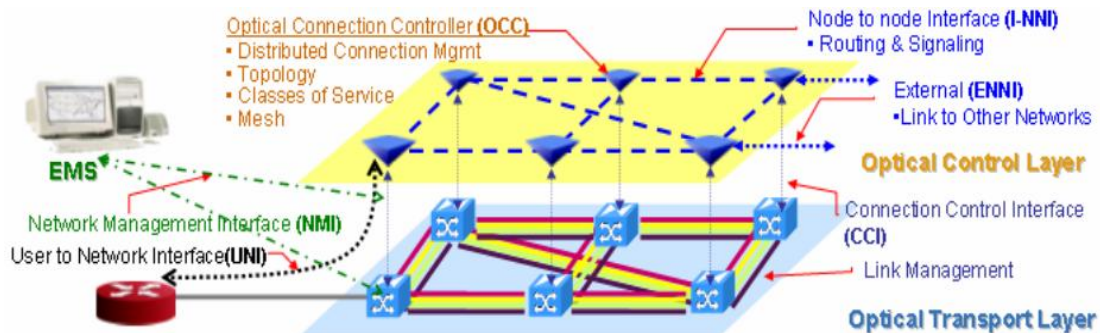


Figura 8.4: Esquema de aplicación de ASON.

Con la consolidación de servicios sobre el backbone óptico de transporte es posible equilibrar las demandas de servicios, sin necesidad de tener que sobre invertir para hacer frente al crecimiento de ancho de banda. Una vez desplegado el plano de control es posible reorganizar las asignaciones de tráfico en forma flexible.

Algunas de las ventajas principales que aporta ASON son:

- Descubrimiento automático de los recursos: cada nodo puede encontrar a su vecino en la red de forma automática y construir la topología de la red por sí mismo.
- Configuración de servicio extremo a extremo como ancho de banda, calidad de servicio y establecimiento del servicio. También realización y selección rápida y automática de rutas en cada nodo, operación a través de las rutas y señalización entre la red y adición de restricciones con el propósito de establecer el trayecto óptimo.
- Protección de red mallada que hace la red más fiable.
- Reducción y ahorro de hasta la mitad del ancho de banda antes utilizado debido a la reconfiguración de caminos y el balance de carga posibles.

8.4 - IP/MPLS

MPLS (Multi-Protocol Label Switching) o **Conmutación de Etiquetas Multiprotocolo**, es un esquema diseñado para acelerar el reenvío de Paquetes IP empleando etiquetas de tamaño fijo en la cabecera del paquete para tomar decisiones de reenvío del mismo (los paquetes son conmutados no ruteados) y está preparado para soportar cualquier protocolo de capa de red y capa de enlace.

En MPLS se define **FEC** (Forwarding Equivalence Class) para denominar a *un conjunto de paquetes que entran en la red MPLS por la misma interfaz, que reciben la misma etiqueta y por lo tanto circulan por un mismo trayecto*. Normalmente se trata de datagramas que pertenecen a un mismo flujo. Una FEC puede agrupar varios flujos, pero un mismo flujo no puede pertenecer a más de una FEC al mismo tiempo. Un paquete FEC puede ser determinados por: dirección IP de origen y/o destino, número de puerto de origen y/o destino, protocolo ID e interface de entrada.

Un **router interior MPLS** es llamado **LSR** (Label Switching Router) y encamina paquetes en función del valor de la etiqueta MPLS dentro de la red MPLS cambiando las etiquetas para cada FEC según indica su **LIB**. **LIB** (Label Information Base) es una tabla de etiquetas que

³⁸ QoS: Quality of Service

manejan los **LSR** para relacionar la interfaz de entrada y etiqueta de entrada con interfaz de salida y etiqueta de salida. Y **LDP** (Label Distribution Protocol) es el protocolo que utilizan los LSR para asignar las etiquetas MPLS (Fig. 8.5).

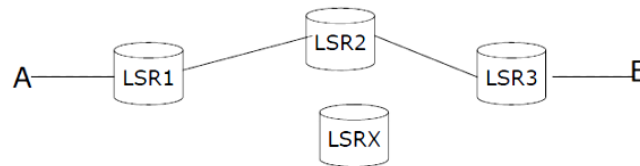


Figura 8.5: Esquema de comunicación MPLS.

Los **LER (Edge LSR)** son los **routers de borde o frontera**. Pueden ser de ingreso o egreso. Los primeros se encuentran en la entrada del flujo a la red MPLS (al principio del LSP), y se encarga de clasificar los paquetes en FECs y poner las etiquetas correspondientes. Y los segundos se encuentran a la salida del flujo de la red MPLS (al final del LSP) y se encargan de eliminar del paquete la etiqueta MPLS dejándolo tal como estaba en el principio.

Un **LSP (Label Switched Path)** es un camino que siguen por la red MPLS los paquetes que pertenecen a la misma FEC. Es equivalente a un circuito virtual en ATM o Frame Relay.

Una de las principales metas del IETF cuando definieron MPLS fue que al agregar esta capa no fuera necesario modificar las capas existentes en el modelo OSI y que sean compatibles con los nuevos protocolos.

El funcionamiento de MPLS es el siguiente:

- En el LER (de ingreso), al llegar un paquete a la red, se coloca una cabecera MPLS antes que el paquete sea reenviado. Esta etiqueta codifica el paquete como FEC. Por cada FEC, se especifica un camino LSP que es unidireccional.
- En los subsiguientes LSRs, la etiqueta sirve como un índice dentro de la tabla de reenvío que especifica el siguiente salto y la nueva etiqueta. La etiqueta vieja con la que llega al LSR es reemplazada por la nueva, y el paquete es reenviado al siguiente salto.
- En el LER (de egreso), se retira la etiqueta MPLS y se reenvía el paquete a su destino final utilizando la cabecera IP.

La Fig. 8.6 muestra el esquema de funcionamiento de MPLS.

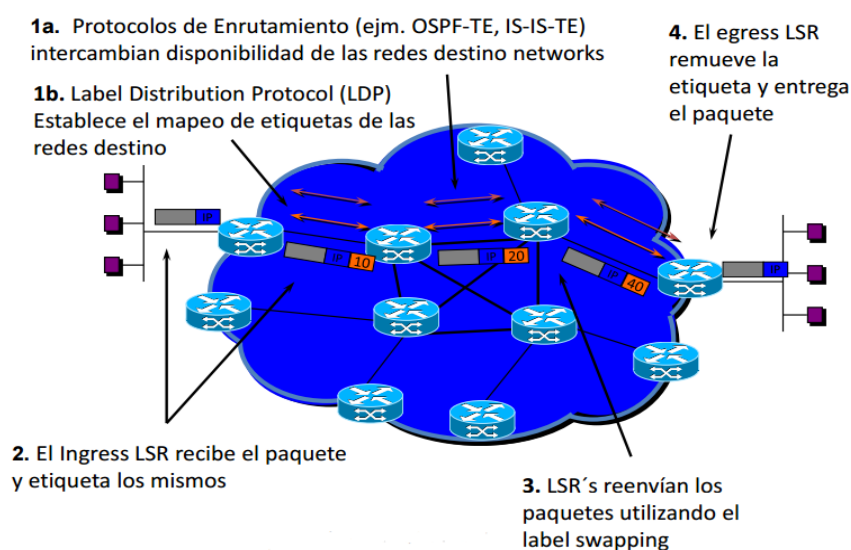


Figura 8.6: Esquema de transporte con MPLS.

Para configurar un LSP, cada LSR debe asignar una etiqueta de entrada al LSP por el FEC correspondiente, la cual solo tiene sentido local, informar al nodo superior de la etiqueta asignada y aprender la etiqueta que el nodo inferior le ha asignado al LSP. Como se necesita un protocolo de distribución de etiquetas para que los LSR puedan informar a otros LSR de las etiquetas/FEC resultantes, se crea una tabla de reenvío junto al LDP.

MPLS Generalizado

GMPLS es un conjunto de protocolos que proveen un control común tanto a paquetes, TDM³⁹ o servicios de longitudes de onda desarrollado desde MPLS pero con variaciones descritas en el RFC 3945.

Es la propuesta para el protocolo de señalización para redes ópticas debido a los problemas de mover un gran volumen de tráfico de manera efectiva en relación al costo, de complejidad en el manejo de capas, de ineficiencia respecto al ancho de banda y de escalabilidad.

Para una arquitectura de capas IP, ATM, SONET/SDH, DWDM⁴⁰, la solución fue eliminar las capas intermedias, quedando IP/WDM, y un protocolo que realice las operaciones de capas intermedias.

Para esto era necesario un plano de control común que: soporte múltiples tipos de tráfico (ATM, IP, SONET), soporte ambas arquitecturas ópticas (modelos peer y overlay), soporte multi-vendors y ejecute un aprovisionamiento rápido.

GMPLS aplica técnicas del plano de control de MPLS a conmutadores ópticos y enrutadores IP con algoritmos que manejan caminos ópticos en una red óptica pero realiza algunas modificaciones a MPLS en cuanto a la separación de los canales de datos y señalización y soporta más tipos de control de interface. Diversas interfaces de Control se pueden jerarquizar dentro de otras (Fig. 8.7):

- Packet Switch Capable (**PSC**): Router/ATM Switch/Frame Relay Switch.
- Time Division Multiplexing Capable (**TDMC**): SONET/SDH ADM/Digital Cross Connects.
- Lambda Switch Capable (**LSC**): All Optical ADM or Optical Cross Connects (OXC).
- Fiber-Switch Capable (**FSC**).

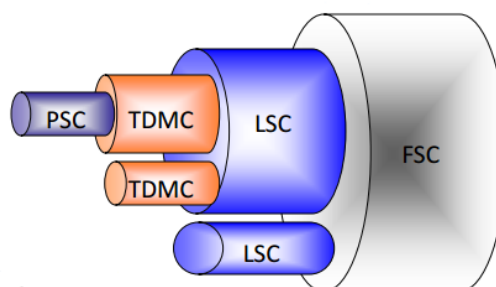


Figura 8.7: Jerarquías de las interfaces de control.

Con la aplicación de IP/GMPLS a las redes ópticas surge una evolución en el esquema general de red para llegar a NGN como lo ilustra la Fig. 8.8.

³⁹ TDM: Time Division Multiplexing; Multiplexación por División de Tiempo.

⁴⁰ DWDM: Dense Wavelength Division Multiplexing; Multiplexación por División en longitud de onda densa.

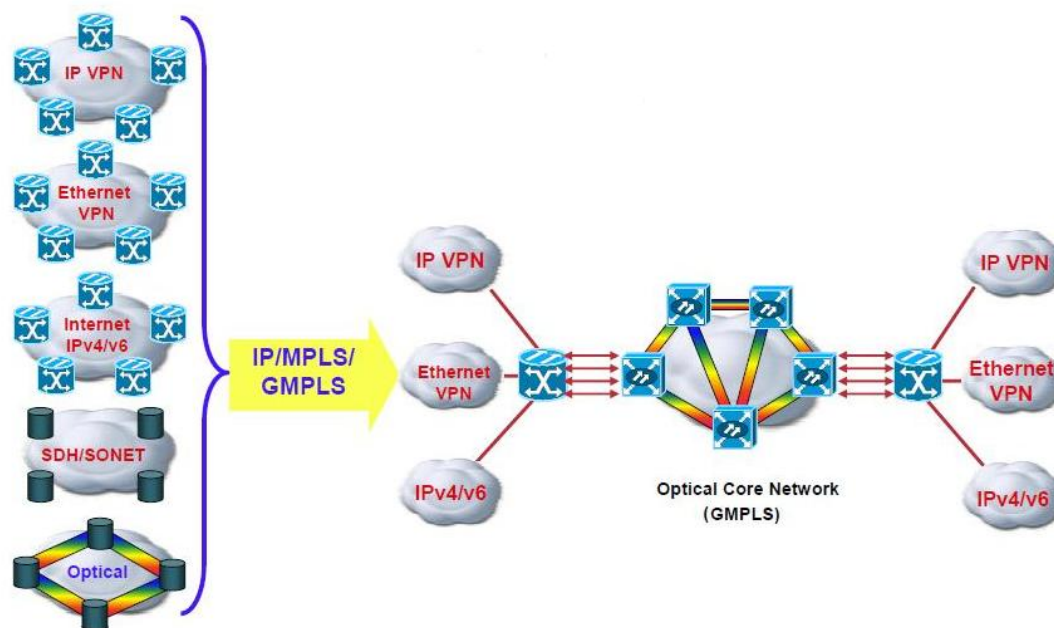


Figura 8.8: Esquema de integración IP/MPLS.

8.5 - Protocolo IMS

El protocolo **IMS (IP Multimedia Subsystem)** es una **arquitectura concebida para ofrecer servicios multimedia**, que además permite a operadores brindar servicios de valor agregado sobre una infraestructura IP existente, sin importar el tipo de red de acceso a estos servicios (Celular, Wireless, redes cableadas, y redes de banda ancha). También, habilita nuevos servicios convergentes multimedia permitiendo la interoperabilidad de estos en distintos tipos de redes. Como IMS trabaja con protocolos abiertos estándares de IP definidos por la IETF, se puede tener gran cantidad de servicios basados en IP [Rivera Rubio, J. M., 2008].

IMS se caracteriza por la independencia de Acceso, ya que trabaja con cualquier tipo de red con funciones packet switching, fija o móvil (CDMA⁴¹, GPRS⁴², UMTS⁴³, WLAN) y diferentes arquitecturas de red. En redes móviles, la red brinda movilidad de terminal mientras que la movilidad del usuario es proporcionada por IMS y SIP. **SIP** es usado por IMS como **protocolo de señalización para el control de sesiones y el control de servicio**.

Arquitectura IMS

La arquitectura IMS sigue la arquitectura modelo de una red NGN definiendo cuatro capas (Fig. 8.9):

- **Capa de Acceso:** puede representar todo acceso de alta velocidad tal como UMTS, UTRAN, CDMA2000, xDSL, Wireless IP, WiFi, WiMax, entre otras.
- **Capa de Transporte:** representa una red IP que tiene la capacidad de integrar mecanismos de calidad de servicios con MPLS, Diffserv, RSVP⁴⁴, entre otros, y tecnologías de transmisión como IP/ATM/SDH, IP/Ethernet, etc.
- **Capa de Control:** IMS introduce un ámbito de control de sesiones sobre el campo de paquetes con **CSCF (Call State Control Function)**. consiste en controladores de sesión

⁴¹ CMDA: Code Division Multiple Access: Acceso Múltiple por División de Código.

⁴² GPRS: General Packet Radio Service: Servicio General de Paquetes vía Radio

⁴³ UMTS: Universal Mobile Telecommunications System: Sistema Universal de Telecomunicaciones Móviles.

⁴⁴ RSVP: Resource Reservation Protocol: Protocolo de Reserva de Recursos

responsables del encaminamiento de la señalización entre usuarios y de la invocación de los servicios.

- **Capa de Aplicación:** consiste en servidores de aplicación (SA: Server Application) y MRF (Multimedia Resource Function) que los proveedores llaman **Servidores de Media IP** (IP MS: IP Media Server).

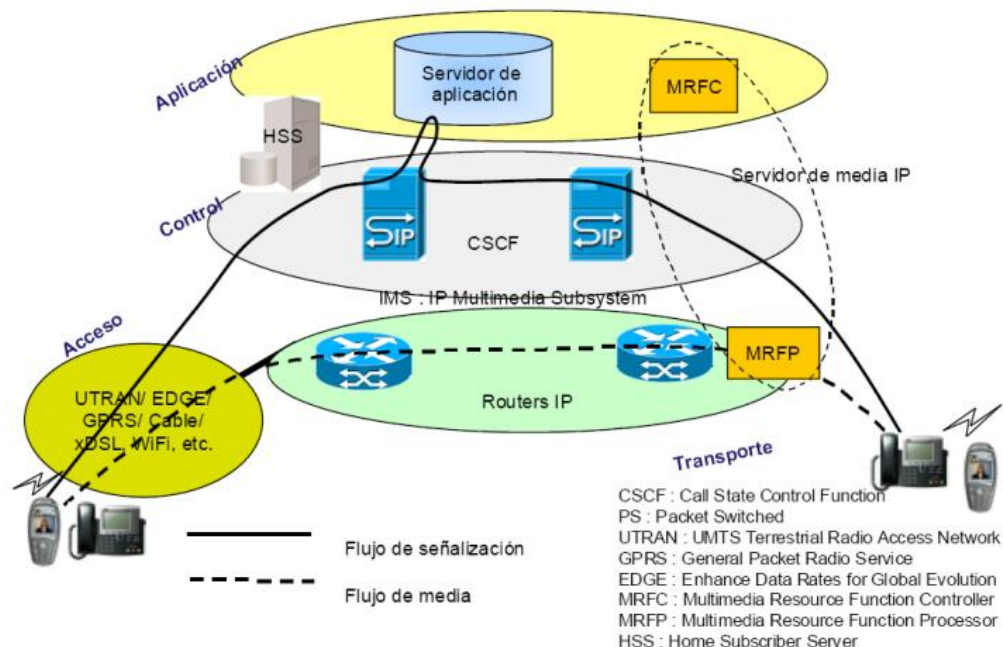


Figura 8.9: Esquema de la arquitectura de IMS.

8.6 - Softswitch

Como la tecnología avanza hacia redes basadas en paquetes, la idea es proporcionar una diversidad de servicios de comunicaciones basados en IP equivalentes a los servicios de redes tradicionales por su calidad y facilidad de uso.

El **Softswitch** es un dispositivo que provee Control de llamada y servicios inteligentes para redes de conmutación de paquete y **sirve como plataforma de integración para aplicaciones e intercambio de servicios**. Son Dispositivos que utilizan estándares abiertos para crear redes integradas de última generación capaces de transportar Voz, Vídeo y datos con gran eficiencia y en las que la inteligencia asociada a los servicios esta desligada de la infraestructura de red.

Es un conjunto de productos, protocolos y aplicaciones capaz de permitir que cualquier dispositivo acceda a los servicios de Internet y servicios de telecomunicaciones sobre las redes IP. Una característica clave del Softswitch, es su **capacidad de proveer a través de la red IP un sistema telefónico tradicional, confiable, de alta calidad y alta capacidad en todo momento**.

Los beneficios que ofrece son:

- Bajo Costo de desarrollo
- Fácil integración de redes diversas
- Mejora los servicios para el cliente lo cual reduce el tiempo y costo
- Mensajes unificados
- Flexibilidad al soportar el desarrollo de equipos de telefonía de gran nivel
- Mejores ingresos para los proveedores de servicios y operadores.

Arquitectura de Servicios del softswitch

Un softswitch puede consistir en uno o más componentes, sus funciones pueden residir en un sistema o expandirse a través de varios sistemas. Sus componentes más comunes son:

- **Gateway Controller (Call Agent o Media Gateway Controller):** Es la unidad funcional del softswitch, que mantiene las normas para el procesamiento de llamadas, por medio del **Media Gateway** y el **Signalling Gateway** los cuales ayudan a mejorar su operatividad. Lo componen:
 - **Signalling Gateway:** Sirve como puente entre la red de señalización SS7 y los nodos manejados por el Softswitch en la red IP; es responsable para ejecutar el establecimiento y desconexión de la llamada.
 - **Media Gateway:** soporta TDM para transporte de paquetes de voz al switch TELCO. Las aplicaciones de Codificación de voz, Decodificación y compresión son soportadas, así como las interfaces PSTN y los protocolos CAS e ISDN⁴⁵.
 - **Media Server:** Mejora las características funcionales del Softswitch si es requerido soporta DSP⁴⁶ y la funcionalidad de IVR.
- **Feature Server:** Controla los datos para la generación de la facturación, usa los recursos y los servicios localizados en los componentes del softswitch.

NGN soporta el protocolo SS7 como sistema de señalización e implementa interworking con redes de conmutación de circuitos. Este protocolo es una estándar que representa señalización por canal común, lo que implica superponer una red de señalización a la red de canales de voz. La señalización es utilizada por los nodos de conmutación para: control de conexión, supervisión y desconexión de los terminales involucrados en una llamada o sesión.

SIP es un protocolo de señalización simple utilizado para telefonía y videoconferencia por Internet, está basado en una arquitectura cliente servidor en la cual los clientes inician las llamadas y los servidores responden las llamadas. SIP es ampliamente soportado en NGN y no es dependiente de un solo fabricante de equipos.

⁴⁵ ISDN: Integrated Services Digital Network. Red Digital de Servicios Integrados.

⁴⁶ DSP: Digital Signal Processor .



La computación en la nube representa un cambio de paradigma: el acceso a la información y a la infraestructura tecnológica es prácticamente omnipresente.

CAPÍTULO 9: INFRAESTRUCTURAS HIPERCONVERGENTES

9.1 - Computación en la nube

La **computación en la nube** o **cloud computing** es una tecnología mediante la cual se prescinde de los recursos de la computadora local y se saca provecho de la capacidad de procesamiento y almacenamiento computacional basado en internet: **la nube**. Por lo tanto, solo se requiere una conexión a internet para acceder a recursos con los que el usuario local no cuenta [xertica, 2018]. Físicamente, los servicios cloud están ubicados en las oficinas de proveedores que el usuario haya escogido (Ej.: el Drive en Google).

Existen 3 tipos principales de nube:

- Privada: es la que ofrece servicios informáticos a través de una red interna privada, exclusiva a algunos usuarios y no disponibles al público general.
- Pública: es la que ofrece servicios informáticos por proveedores externos a través de internet.
- Híbrida: combina ambas características, lo cual permite una dinámica entre nubes, dependiendo de las necesidades y los costos con los que se cuente.

Dentro de la nube existen categorías de servicios: SaaS, PaaS, IaaS. Estos son conocidos como tecnologías hiperconvergentes.

9.2 - Tecnologías hiperconvergentes

Si bien, en los laboratorios ya se venían trabajando, los sistemas hiperconvergentes reciben su denominación hacia el año 2012, con el fin de diferenciarlos de las tecnologías convergentes. En ésta última tecnología, la computación, el almacenamiento y la red pueden separarse, mientras que en la hiperconvergente no es posible. Los tres componentes se combinan en un único sistema. Esto es porque están basados en software y no en hardware, está basado en la virtualización.

Con la implementación de esta tecnología, es más sencillo gestionar el data center y es más fácil adaptar el mismo al crecimiento de la organización. La **infraestructura hiperconvergente es escalable, se puede expandir añadiendo nodos a la unidad base**. La mayoría de estos sistemas requieren un mínimo de tres nodos de hardware para una alta disponibilidad. Varios nodos se agrupan en cluster y crean pools de computación compartida y recursos de almacenamiento, diseñados para el consumo conveniente. Solo se usa lo que se necesita.

Estas tecnologías incluyen un hipervisor para computación virtualizada, un almacenamiento definido por software y una red virtualizada (Fig. 9.1). Por lo general, se ejecutan en servidores estándar y listos para utilizar. Normalmente, se venden en forma de

caja con todo el hardware necesario, aunque puede ser también un software preparado para instalarse en un hardware ya existente.

La infraestructura hiperconvergente, para organizaciones en crecimiento

Los **sistemas hiperconvergentes** están basados en un almacenamiento definido por software y esto representa una gran oportunidad para las organizaciones pequeñas con pocos recursos y con un mínimo espacio en sus instalaciones. Pueden disfrutar de una infraestructura de punta sin tener que depender de almacenamiento en local.

Con esta tecnología las pequeñas y medianas empresas pueden tener acceso al almacenamiento, la computación y la red que necesitan sin tener que invertir en un inicio una gran cantidad de dinero. Pueden ajustar la tecnología utilizada para gestionar los datos del negocio a los presupuestos limitados de las mismas.

IaaS

IaaS (Infrastructure as a Service: Infraestructura como servicio) permite a un proveedor alquilar en la nube una infraestructura de TI altamente automatizada y **escalable**, como ser almacenamiento, alojamiento, computación, redes (Fig. 9.1). Los costos en que incurre la organización solicitante abarcan solo lo que usa. Por lo tanto, en lugar de poseer activos como licencias de software o servidores, las empresas pueden alquilar recursos de forma flexible de acuerdo con sus necesidades [Bitec, 2018].

Grandes Proveedores actuales: AWS, Microsoft, Google e IBM.

PaaS

PaaS (Platform as a Service: Plataforma como servicio) es una categoría de servicios en **la nube** que proporciona una **plataforma y un entorno que permiten a los desarrolladores crear aplicaciones y servicios que funcionen a través de internet** (Fig. 9.1). Los servicios PaaS se **alojan en la nube**, y los usuarios pueden acceder a ellos simplemente a través de su navegador web [Bitec, 2018].

El modelo PaaS permite a los **usuarios crear aplicaciones utilizando herramientas suministradas por el proveedor**. Los servicios PaaS pueden consistir en funcionalidades preconfiguradas a las que los clientes puedan suscribirse, eligiendo las funciones que deseen incluir para resolver sus necesidades y descartando aquellas que no necesiten. **Así, los paquetes pueden variar desde un sencillo entorno que se maneje con el ratón y no requiera ningún tipo de conocimiento o instalación especial por el lado del usuario, hasta el suministro de opciones de infraestructura para desarrollo avanzado.**

La infraestructura y las aplicaciones se gestionan en nombre del cliente, y se ofrece también soporte técnico. Los servicios se actualizan constantemente, mejorando las funcionalidades existentes y añadiendo otras nuevas. **Los proveedores de PaaS pueden colaborar con los desarrolladores desde la concepción de sus ideas originales hasta la creación de las aplicaciones, llegando incluso hasta las fases de pruebas e implantación.** Y todo eso se consigue utilizando un solo mecanismo gestionado [Interoute, 2018].

Los servicios PaaS se **facturan como una suscripción en la que el cliente acaba pagando al final sólo por lo que realmente utiliza.**

Estas son algunas de los servicios dentro de una propuesta de PaaS: **Sistema operativo, Entorno de scripting de servidor, Sistema de gestión de base de datos, Software de servidor, Soporte técnico, Almacenamiento, Acceso a la red, Herramientas de diseño y desarrollo, Hosting.**

El modelo PaaS aporta ventajas tanto a los desarrolladores de software como a los programadores de webs y a las empresas. **Una propuesta PaaS proporciona un entorno de trabajo para el desarrollo de aplicaciones [Interoute, 2018].** Un proveedor de PaaS es SmartFile.

SaaS

SaaS (Software as a Service: Software como servicio) se describe cualquier servicio en la nube en el que los **consumidores puedan acceder a aplicaciones de software a través de internet** (Fig. 9.1). Esas aplicaciones están alojadas "en la nube" y pueden utilizarse para una amplia variedad de tareas, tanto para particulares como para organizaciones [Bitec, 2018].

Ejemplos de proveedores de SaaS: Google, Twitter, Facebook y Flickr. Los usuarios pueden acceder a los servicios a través de cualquier dispositivo que pueda conectarse a internet. **Los usuarios empresariales pueden utilizar aplicaciones para resolver necesidades muy diversas, desde la contabilidad y la facturación hasta el seguimiento de ventas, planificación, control de rendimiento y comunicaciones** (por ejemplo, el correo web y la mensajería instantánea).

El modelo SaaS se conoce también a veces como "software a demanda", y la forma de utilizarlo se parece más a **"alquilar" el software que a comprarlo**. Con una aplicación tradicional, el software se compra y luego se lo instala. Su licencia puede indicar limitaciones. Por el contrario, en SaaS, los usuarios se "suscriben o alquilan periódicamente" aplicaciones, y los archivos se guardan en la nube, no en la computadora.

El uso de servicios SaaS es escalable, con actualizaciones disponibles a demanda; el acceso al software como servicio es compatible con todo tipo de dispositivos capaces de conectarse a internet [Interoute, 2018].

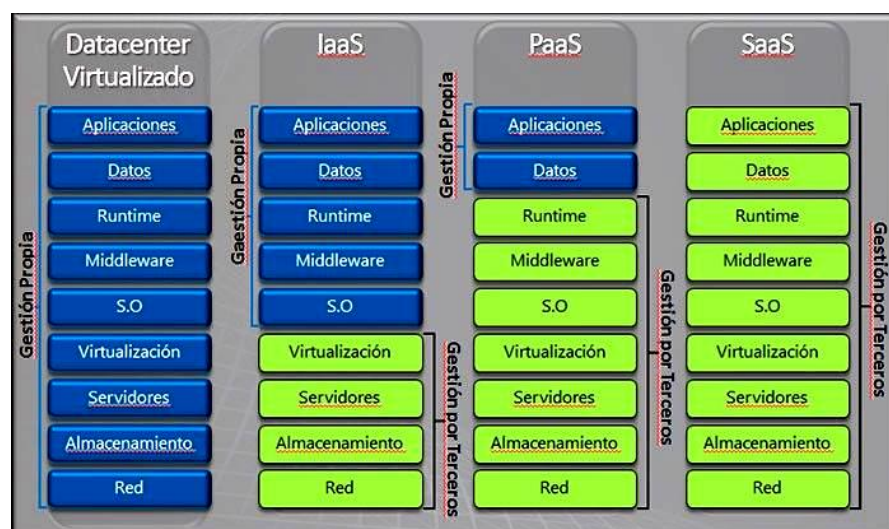


Figura 9.1: IaaS, PaaS, SaaS. Fuente: Bitec, 2018.



CAPÍTULO 10: DISEÑO DE UNA RED ESCALABLE PARA E-GOV

10.1 – Etapas del diseño

En este capítulo se presenta un **diseño genérico de red escalable para e-Gov**, que será denominado **“Red E-GovG”**. **Toda fundamentación que sustenta esta propuesta esta descrita en los capítulos anteriores: e-Gov y estudios de caso, servicios web, estándares básicos, SOA, escalabilidad, redes NGN y tecnologías hiperconvergentes.**

Para la obtención del Modelo de la Red E-GovG se propone una serie de etapas, detallando infraestructura, plataformas, aplicaciones, protocolos y demás elementos necesarios:

- Etapa 1: Definición del ámbito de E-Gov para esta propuesta
- Etapa 2: Selección de los modelos de comunicación del E-GovG a adoptar.
- Etapa 3: Definición del tipo de Escalabilidad a lograr en E-GovG
- Etapa 4: Provisión de la infraestructura de la red
- Etapa 5: Diseño de la Arquitectura General de la red E-GovG
- Etapa 6: Diseño de la accesibilidad para la plataforma de E-GovG
- Etapa 7: Implantación del modelo SOA con servicios web.

10.2 – Diseño

Etapa 1: Definición del ámbito de e-Gov para esta propuesta

En la búsqueda de antecedentes sobre e-Gov y el marco teórico de este trabajo (**Capítulos 2 y 3**) se pudo observar que existen plataformas de e-Gov tanto para municipios, provincias o países. Desde cada portal se observan diferentes niveles de interacción con la ciudadanía, las empresas y las entidades gubernamentales, dentro del ámbito correspondiente.

Esta propuesta de diseño será válida para un “e-Gov en cualquier ámbito”, de ahora en más denominado “**E-GovG**” (Fig. 10.1).



Figura 10.1: E-GovG, un e-Gov para cualquier ámbito.

Etapa 2: Selección de los modelos de comunicación del E-GovG a adoptar

Los **actores** de un e-Gov son los **ciudadanos**, las **empresas** y las **entidades gubernamentales**. Como se vio en el **capítulo 1**, en un esquema de e-Gov se pueden distinguir diferentes niveles de interacción: Gobierno–Ciudadano, Gobierno–Empresa y Gobierno–Gobierno.

En esta propuesta, se contemplarán los tres tipos de interacción. El diseño responde a estándares normalizados y plataformas de uso general (Fig. 10.2).

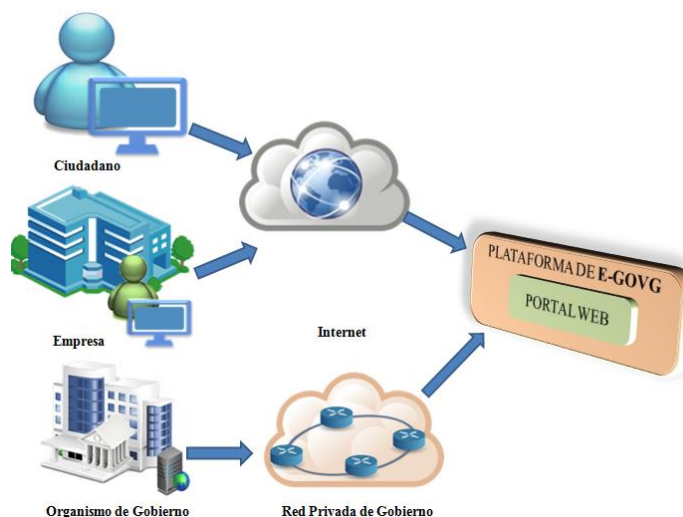


Figura 10.2: Esquema de interacción con E-GovG.

Etapa 3: Definición del tipo de Escalabilidad a lograr en E-GovG

La escalabilidad es necesaria para la expansión y el crecimiento de las empresas, para su adaptación dinámica en contextos tecnológicos actuales. Como se vio en el **capítulo 7** la escalabilidad puede ser horizontal o vertical. Cada tipo tiene una finalidad e influye en el rendimiento general del sistema. Así también, las tecnologías hiperconvergentes, analizadas en el **capítulo 9**, permiten la escalabilidad mediante servicios IaaS, PaaS y SaaS.

En esta propuesta el modelo de E-GovG permitirá la escalabilidad horizontal y escalabilidad en la nube. La finalidad es posibilitar una mejor oferta de servicios para una gobernabilidad más eficiente, más ágil y participativa. Por consiguiente se propone escalabilidad horizontal y escalabilidad en la nube.

La **Escalabilidad horizontal**, posibilitará la incorporación de diferentes nodos (Fig. 10.3), que se denominarán:

- “**nodo T**”, que incluye infraestructura de hardware y software (plataforma, aplicaciones) propios por el gobierno. Con este tipo de nodos se podrá expandir la plataforma, es decir, agregar nuevos servicios para los actores del E-GovG.
- “**nodo F**”, que incluye el aumento de funcionalidad (nuevos servicios para los actores) sin cambiar el hardware subyacente propio del gobierno. Muy probablemente esto generará un desbalance de carga, pero con aplicaciones adecuadas se podrá redistribuir la misma.

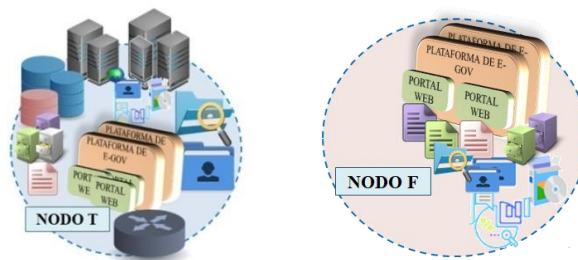


Figura 10.3: Nodos T y F para la escalabilidad horizontal.

Por otro lado, la **Escalabilidad en la nube** se implementará según el tipo de servicio requerido. Se denominarán “nodo N” a los nodos que permitan la escalabilidad en este paradigma (Fig. 10.4). En esta área, la escalabilidad puede ser horizontal o vertical, va a depender de una decisión estratégica-gubernamental. A su vez, los nodos N podrán ser:

- “**nodo NI**”, que incluye el alquiler solamente de servicios de Infraestructura. Esto permitirá **la escalabilidad vertical**.
- “**nodo NP**”, incluye el alquiler de servicios de Infraestructura y plataformas. Esto permitirá **la escalabilidad vertical**.
- “**nodo NS**”, incluye el alquiler total (Infraestructura, plataforma, aplicaciones). Esto permitirá **la escalabilidad horizontal**.

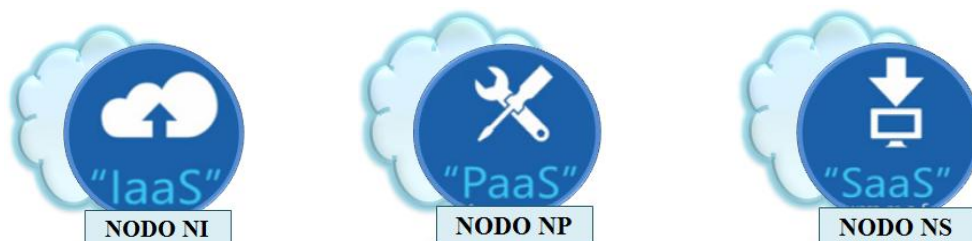


Figura 10.4: Nodos para la escalabilidad en la nube.

La **escalabilidad horizontal** permitirá expandir la funcionalidad de E-GovG: brindar mayores servicios a los ciudadanos, empresas y entidades gubernamentales (Fig. 10.5). Cada

vez que se requiera implementar un nuevo servicio se deberá anexar un nodo T, F o NS, dependiendo de las disponibilidades de los recursos y decisiones gubernamentales.

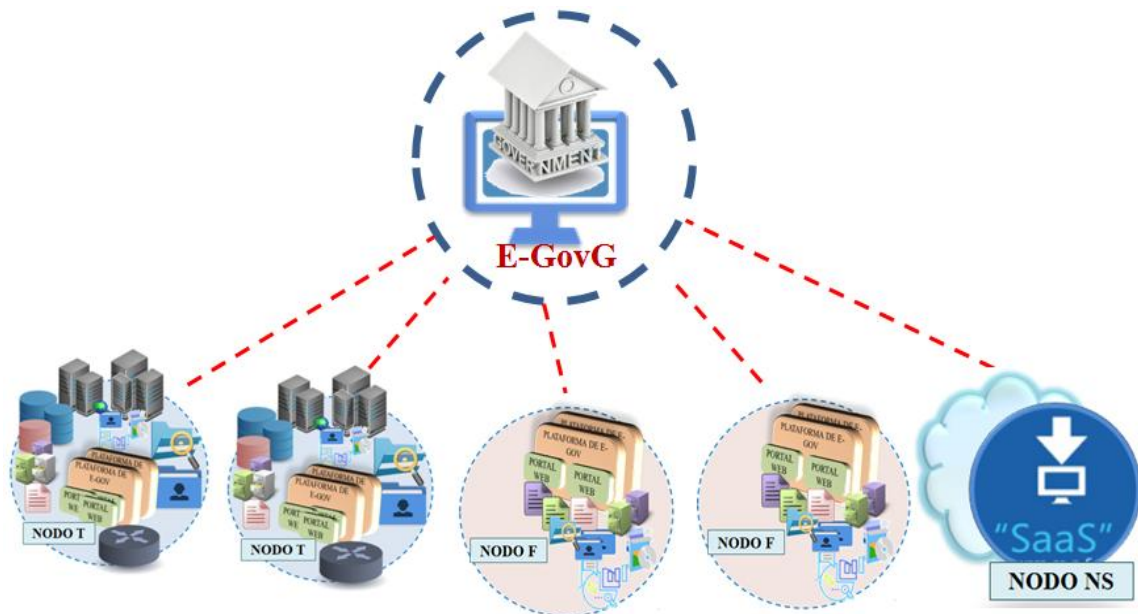


Figura 10.5: Escalabilidad horizontal para expandir la funcionalidad de E-GovG.

Por su parte la **escalabilidad vertical** se podrá lograr adquiriendo mayor infraestructura para mejorar la capacidad computacional de la red, o bien, en caso de no disponer de fondos propios, se podría recurrir a servicios de la nube, alquilando infraestructura, o bien plataformas de desarrollo (Fig. 10.6).



Figura 10.6: Escalabilidad Vertical para una mejorar la infraestructura y capacidad computacional.

Estos modelos de escalabilidad propuestos permitirán evolucionar al E-GovG en su desarrollo. Según el **capítulo 1**, existen diferentes modelos de desarrollo. El Modelo de Gartner es el más representativo con 4 fases: presencia, interacción, transacción y Transformación. **Esta propuesta de escalabilidad permitirá implementar ese modelo, e inclusive agregar una quinta fase: la democracia electrónica.**

Etapa 4: Provisión de la infraestructura de la red

Según los casos de estudio del **capítulo 2**, un gobierno puede implantar su plataforma de E-Gov por sus propios medios (Caso de Uruguay y Estonia), o bien pueden alquilar espacios en la nube, (países de Sudáfrica).

En esta propuesta se puede considerar ambos modos de provisión (Fig. 10.7):

- Caso 1: **E-GovG con recursos propios** (Infraestructura, plataformas, datacenter, servidores, aplicaciones, plataformas de desarrollo, redes, etc.).

- Caso 2: **E-GovG con recursos alquilados** a un proveedor, en forma total o parcial. O bien, configurar un tercer Caso: **E-GovG con recursos propios** y otros alquilados, es decir, **híbridos**. Lo importante es que la infraestructura debe responder a estándares normalizados e interoperables.



Figura 10.7: E-GovG con recursos propios y E-Gov con recursos alquilados.

Si el gobierno tiene su propia infraestructura, en el core de la red se interconectan los nodos T y F para la escalabilidad horizontal (Fig. 10.7).

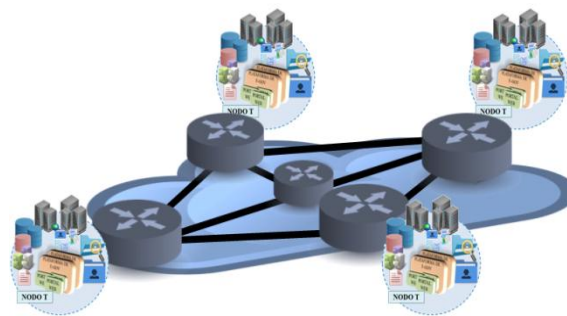


Figura 10.8: Core de E-GovG.

Para la conexión de los nodos T se realizará un tendido de fibra óptica que soporte la tecnología DWDM en los equipos de la red, pudiendo alcanzar velocidades del orden los terabytes. La señalización de los paquetes y la conmutación de los mismo se realizará utilizando los protocolos IP/GMPLS. A modo de ejemplo se puede ver en la Figura 10.9 como sería el despliegue del core para un E-GovG a nivel nacional.



Figura 10.9: Mapa de red del gobierno de la República Argentina. Fuente: Castillo, Echenique y García, 2017.

Para la comunicación con los a nivel gubernamental el core se conectará a la red privada de gobierno. Para ello se requiere el despliegue de la red, que siguiendo el ejemplo, se realizaría por todo el país. La Figura 10.10 muestra un ejemplo de parte del despliegue de la red nacional hacia el noroeste del país en estado de desarrollo.

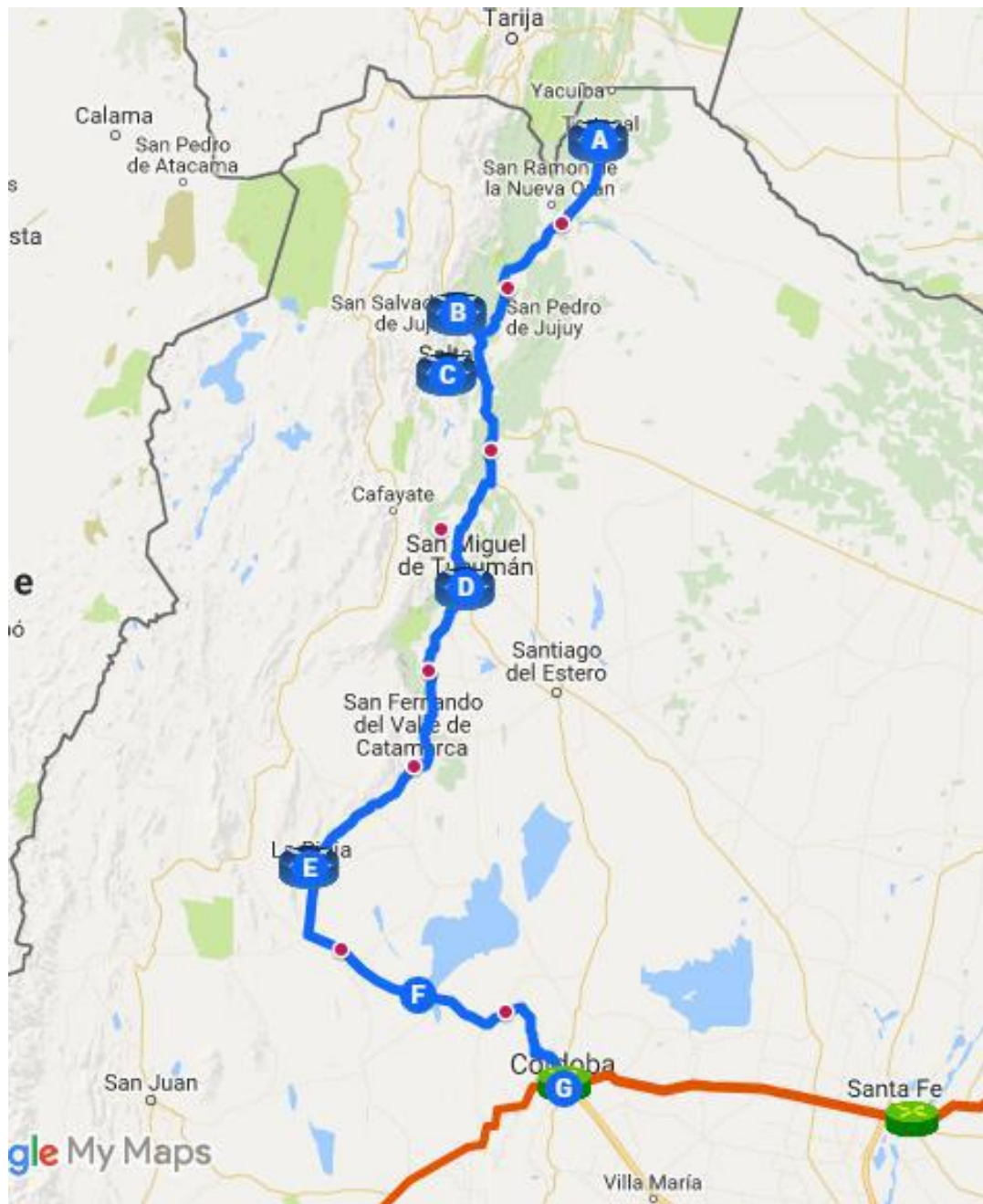


Figura 10.10: Mapa de red del gobierno de la República Argentina – Zona Noroeste. Fuente: Castillo, Echenique y García, 2017.

Etapa 5: Diseño de la Arquitectura General de la red E-GovG

Según lo estudiado en el **capítulo 8**, se propone usar la arquitectura NGN. El diseño general de la red de E-GovG quedará conformado en cuatro capas (Fig. 10.11):

- Acceso
- Transporte
- Control
- Gestión, servicios y aplicación.

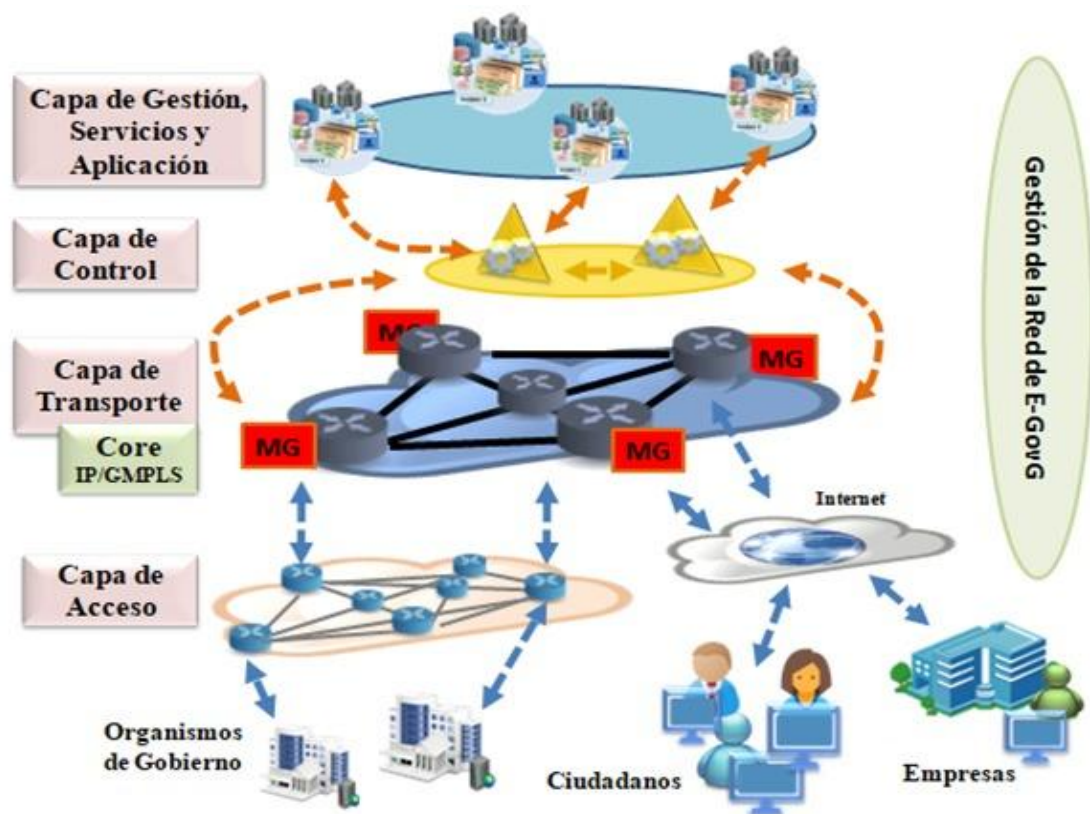


Figura 10.11: Arquitectura general de la red E-GovG.

En la **capa de acceso** se puede distinguir los distintos accesos de acuerdo al tipo de actor:

- Los ciudadanos y las empresas se conectan a través de un proveedor de servicio de internet (ISP). La nube del ISP puede conectarse al LER del core de E-GovG utilizando cualquier tecnología, ya que el protocolo GMPLS implementado realiza la encapsulación de los paquetes que recibe y los conmuta internamente.
- Los organismos de gobierno utilizan la red de convergencia privada del gobierno. Para ello se conectarán a través de un LER utilizando una conexión de fibra óptica.

La **capa de transporte** está compuesta por el core IP/GMPLS que basado en el protocolo LDP y la señalización SS7, etiquetan todos los paquetes IP que circulan por la red. En ella se encuentran los Media Gateways y los LER para del flujo de paquetes que ingresan de otras redes y los equipos LSR en su interior. Esta capa también utiliza fibra óptica y la tecnología DWDM.

La **capa de control** está compuesta por el softswitch que controla y gestiona la conmutación de los paquetes IP y de los servicios. Es una intermediaria entre la capa de Gestión, Servicios y aplicación y la convergencia en la capa de transporte. El softswitch aporta a la red los equipos Media Gateways, Signalling Gateway, entre otros.

La **capa de Gestión, Servicios y Aplicación** contiene el equipamiento general con los cuales se proporcionan los servicios y aplicación a la red de transporte, los cuales se definieron como Nodos T. Está relacionada con el control de la arquitectura a través de softswitch.

Los **nodos T** se interconectarán utilizando el paradigma NGN, que se presentó en el capítulo 8. La conexión se realizará con fibra óptica, utilizando DWDM como protocolo de enlace, e IP/GMPLS como protocolo de transporte.

Toda la arquitectura en general se controlará y gestionará a partir de la implementación de los protocolos IMS y de los componentes del esquema de aplicación de ASON y el Softswitch.

Etapa 6: Diseño de la accesibilidad para la plataforma de E-GovG

El acceso a la plataforma E-GovG varía de acuerdo al tipo de actor (Fig. 10.12). El **acceso de los ciudadanos** a la plataforma de E-GovG se realizará utilizando Internet, y en la plataforma a través inicio de sesión simple.

El **acceso de las empresas** se realiza a través de una nube de Internet que pueda ingresar a la plataforma de dos maneras: un **inicio de sesión único**, para servidores de la empresa, o un **inicio con seguridad, autenticación y autorización** según los servicios y procesos que requieran.

La comunicación de los **organismos de gobierno y la plataforma** se apoya en la red privada de gobierno y el acceso puede ser de dos maneras, de acuerdo al servicio solicitado: con un inicio de sesión único, para servidores público; o un inicio con seguridad, autenticación y autorización según los servicios y procesos que requieran. El tráfico de los paquetes organismos y gobierno está permitido libremente, sin embargo debe ser habilitado por un equipo de soporte del E-GovG.

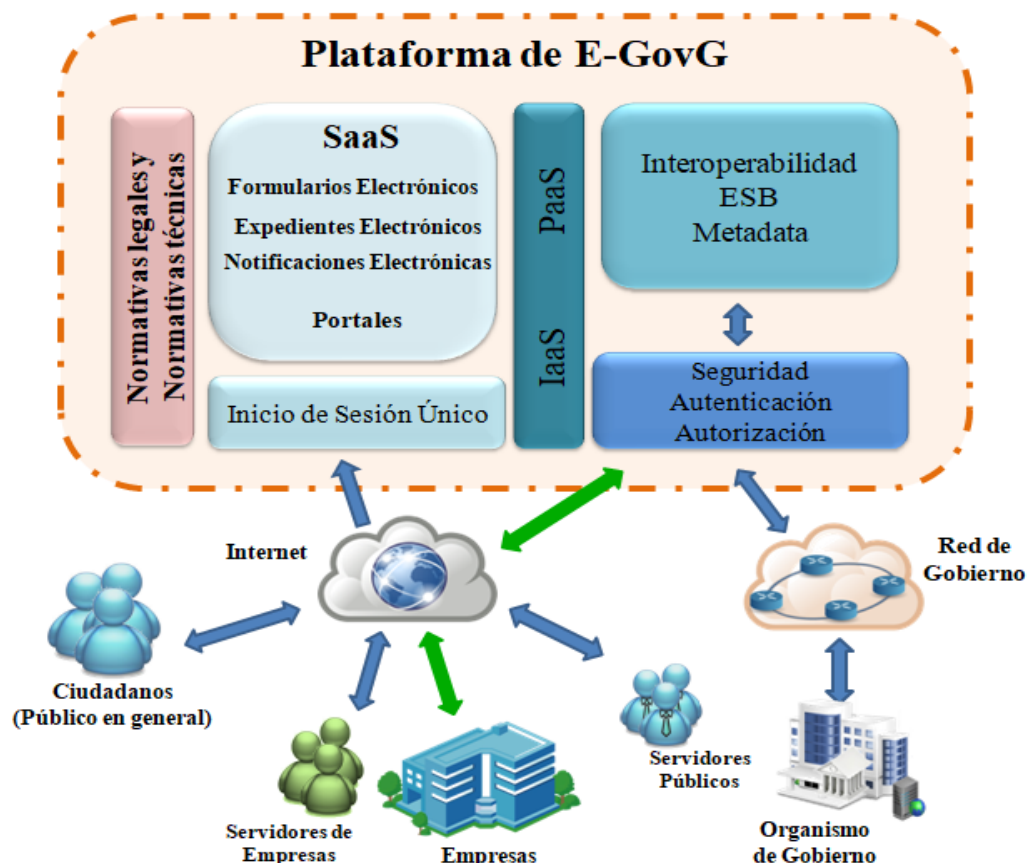


Figura 10.12: Capa de acceso a la red E-GovG.

Etapa 7: Implantación del modelo SOA con servicios web

Según lo estudiado en los **capítulos 4, 5 y 6**, se propone implantar el modelo SOA con servicios web para la comunicación de acuerdo a los estándares básicos descritos en el

marco teórico. Para ellos se define el modelo de protocolos básico para E-GovG en la Fig. 10.13.

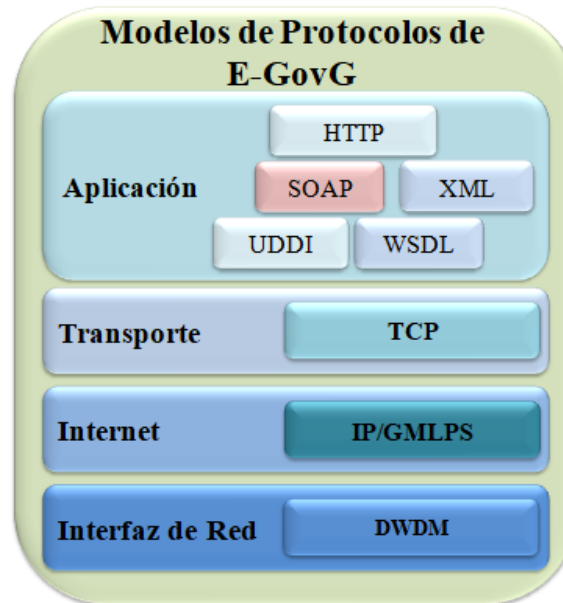


Figura 10.13: Modelo de Protocolos de E-GovG.

El funcionamiento de los servicios web con los protocolos definidos para SOA, puede representarse en 7 pasos:

1. El proveedor de servicio define la descripción abstracta de un servicio utilizando WSDL.
2. Luego se crea un servicio concreto a partir de la descripción (Fig. 10.14).

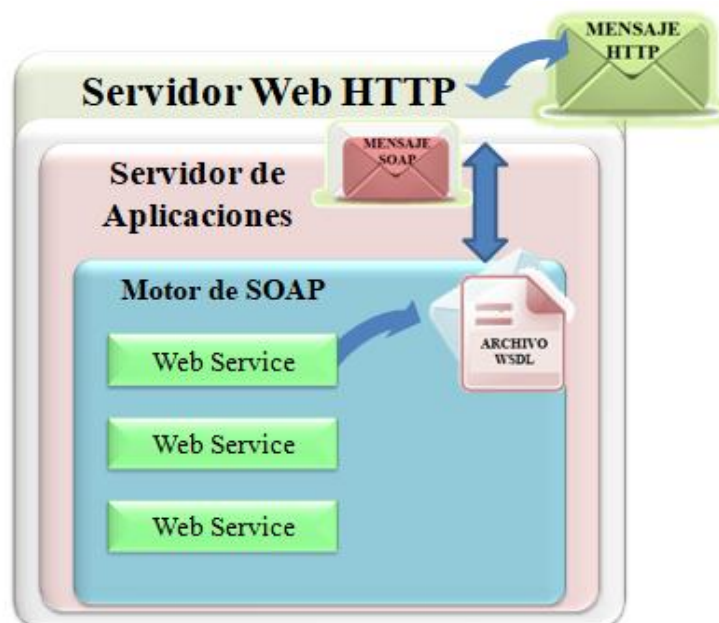


Figura 10.14: Implantación del Modelo SOA en Servicios Web. Paso 1 y 2.

3. El proveedor de servicio publica en el UDDI la descripción del servicio creado (Fig. 10.15).

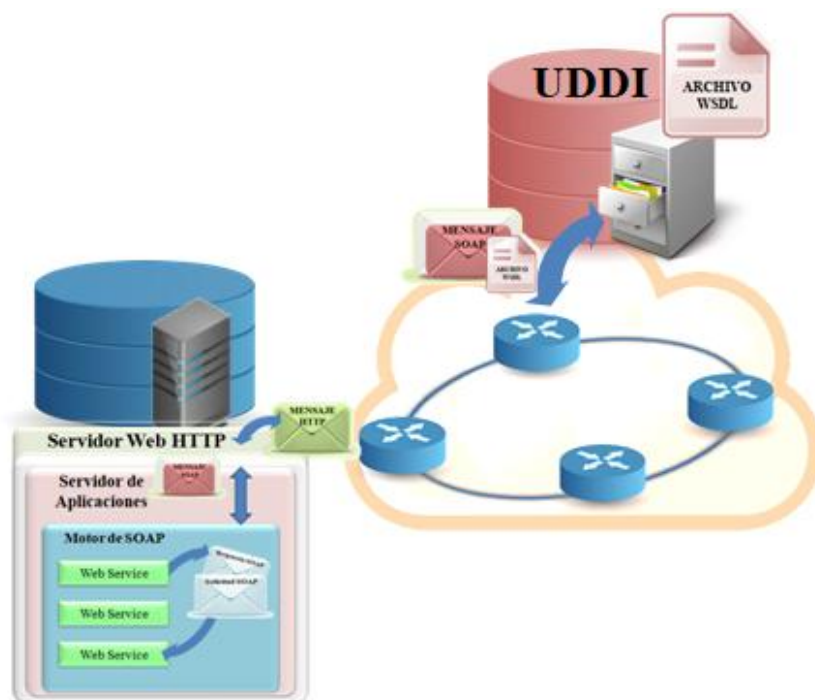


Figura 10.15: Implantación del Modelos SOA en Servicios Web. Paso 3.

4. La aplicación cliente requiere un servicio web. Para ello solicita al UDDI la descripción del servicio. El UDDI recibe la solicitud y envía la descripción del servicio correspondiente (Fig. 10.16).

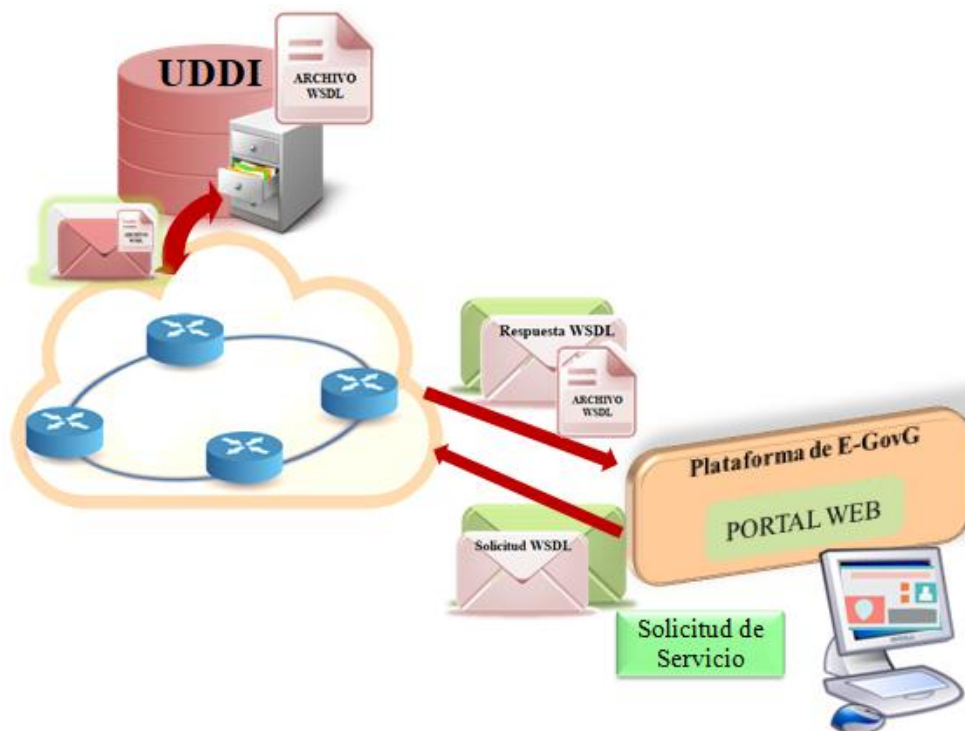


Figura 10.16: Implantación del Modelos SOA en Servicios Web. Paso 4.

5. La aplicación cliente basándose en la descripción, envía una solicitud de servicio al Proveedor de Servicio (Fig. 10.17).

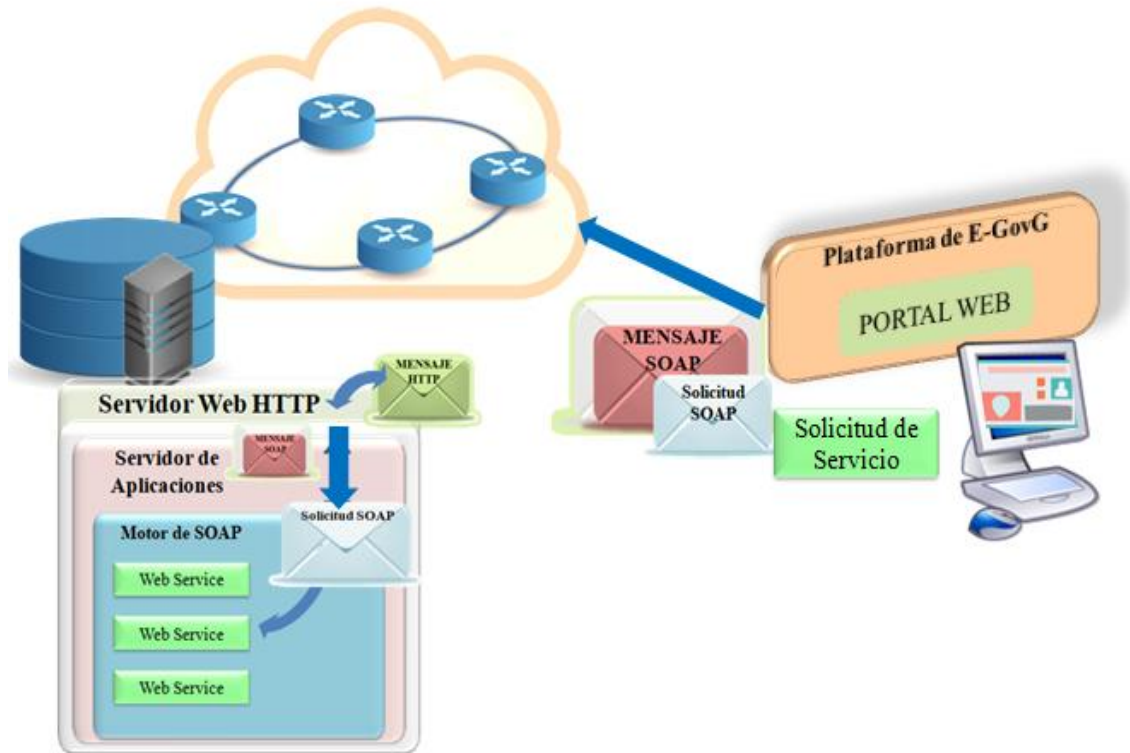


Figura 10.17: Implantación del Modelos SOA en Servicios Web. Paso 5.

6. El proveedor valida la solicitud de servicio y envía el dato estructurado del servicio en formato XML (documento XML).
7. La aplicación cliente valida el archivo XML utilizando un archivo XSD (Fig. 10.18). Finalmente, puede ejecutar el servicio.

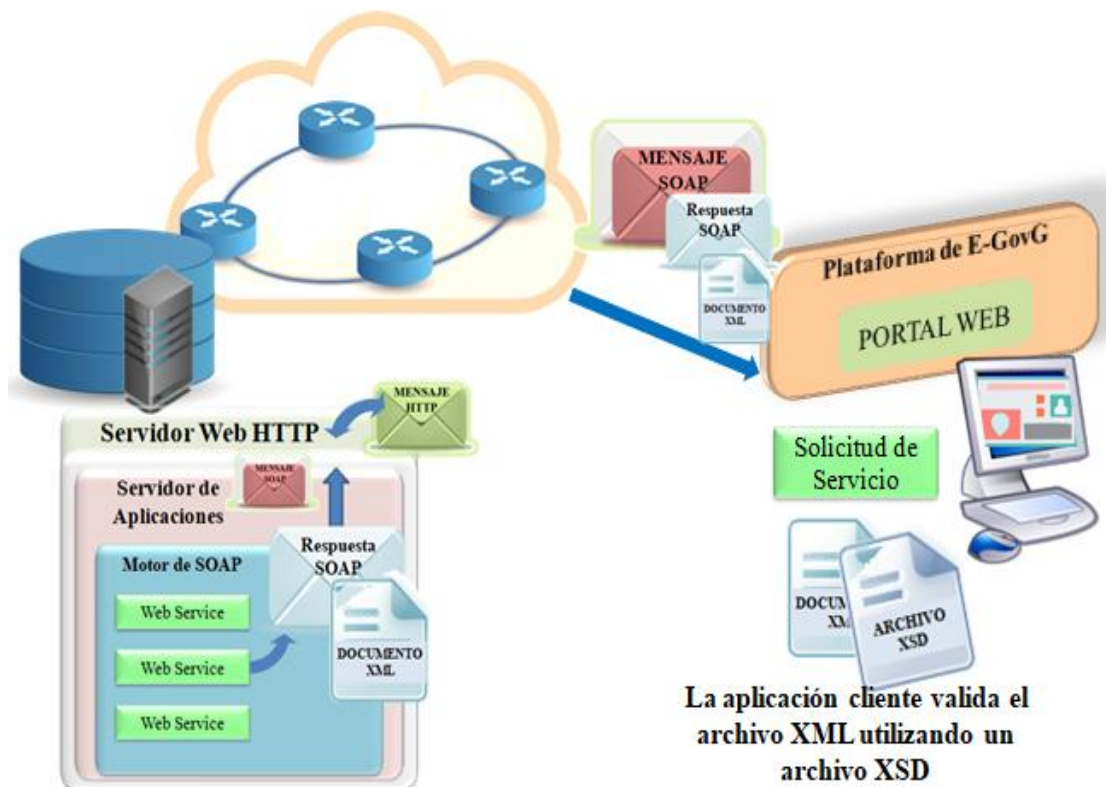


Figura 10.18: Implantación del Modelos SOA en Servicios Web. Paso 6 y 7.

10.3 – Ventajas del modelo E-GovG

El modelo e-GovG propone la implementación de un red de telecomunicaciones, escalable y aplicado a un e-Gov, independiente del ámbito de gobierno (municipal, provincial, nacional), con o sin recursos propios.

La principal ventaja de este modelo escalable es la **expandibilidad funcional** de la administración pública “electrónica” usando redes de telecomunicaciones. Sobre todo el hecho de agregar funcionalidades de una manera rápida, que pueden realizarse utilizando los nodos, que trabajan a nivel de software y/o hardware.

Todo este modelo apunta a que el beneficiario directo y principal sea el ciudadano. Los servicios ágiles permitirán una mejor atención, personalizada y con transparencia. Además del ahorro de tiempo significativo que se invierte realizando largas colas de espera. Según un estudio realizado en Estonia, la persona promedio ahorra dos semanas al año en esperas, para ser atendidos, gracias al e-Gov implementado.

En el capítulo 2 se vio las etapas de desarrollo de e-Gov según el grupo Gartner. Según este modelo, la primera fase tiene que ver con la **presencia** del gobierno en la web a nivel informativo, procurando configurar el portal con datos e información acerca de las autoridades, entes, organismos, ministerios, etc. Para esclarecer mejor la cuestión pueden visitar el portal de Salta y el portal de Orán.

Cuando ya se ha superado la etapa inicial, los gobiernos procuran sistematizar algunas funciones administrativas, para iniciar la intercomunicación con el ciudadano. En esta nueva etapa, denominada **interacción**, se abre un canal de comunicación de doble vía, donde se brinda información y también se reciben opiniones y comentarios, ya sea a través de foros o envío de formularios. Un ejemplo claro de esta etapa es el hecho de sacar turno por Internet, desde el portal de la policía de Salta, para realizar el trámite de antecedentes penales en las oficinas físicas. Si bien todavía existe un tiempo de espera, este es ahora menor, ya que el turno especifica el rango horario en el que el ciudadano debe presentarse. Para realizar esta acción puede hacerse uso de un nodo F, que contenga el servicio o la aplicación que genere un número y registre al ciudadano que lo solicita, y no requerirá inversión de hardware, ya que simplemente se trata de software sencillo que aprovecha los recursos disponibles.

Para las etapas 3 y 4, que son **transacción** y **transformación**, la interacción Gobierno-ciudadano es mayor. Existen trámites que requieren la interrelación de diferentes oficinas, por lo que tener un e-Gov más desarrollado significa un gran beneficio en cuestión de agilidad administrativa, atención personalizada y gastos de tiempo innecesario. Lo que implica la alfabetización tecnológica de la sociedad en este ámbito y varios cambios en la gestión y la administración. Un ejemplo de estas etapas puede ser la implementación de un e-voto, un voto electrónico, que permita a cualquier ciudadano, sin importar la ubicación geográfica, participar de el proceso de sufragio directamente desde una computadora conectada a Internet. Esta parte de la expandibilidad requiere de altos costos de infraestructura y software, que pueden ser cubiertos con cualquier tipo de nodo, que según el ejemplo podrían ser de tipos NP o NI de la escalabilidad en la Nube, ya que todo lo que implica el proceso electoral es una actividad que se realiza cada 2 o 4 años.

La transformación también implica una democracia participativa donde los ciudadanos participan en el gobierno con ideas, sugerencias, propuestas, etc. Esto influenciaría en la toma de decisiones para la administración del bien común, de manera permanente, moldeando así el futuro del país.

Por último, pero no menos importante, nombrar las siguientes ventajas:

- La interoperabilidad que aportan los servicios web tanto entre las aplicaciones como en las plataformas de software independientes o de propietarios diferentes.
- La flexibilidad que aporta la utilización de los estándares básicos, como XML, WSDL, UDDI, HTTP y SOAP y que en conjunto ofrecen seguridad y confiabilidad a todos los aspectos que rodean a los servicios web y la comunicación entre los elementos de software de la red.
- La combinación de los sistemas distribuidos y de los servicios web a través de SOA, que permite la reutilización del software y la creación de sistemas altamente escalables.
- La flexibilidad e inteligencia de la red NGN a partir de la posibilidad de multiservicios integrados basados en IP, GMPLS, ASON y DWDM.

CONCLUSIÓN

En este trabajo se realizó una investigación acerca del concepto de e-Gov y todo lo que abarca, estudiando casos de aplicación en el mundo actual. Se pudo encontrar que las potencias de referencia a nivel mundial son: Estonia y Uruguay.

Estonia es el primer país que declara que el acceso a Internet es un derecho humano y sus ciudadanos tienen un fuerte nivel de educación desde primer grado en el área de informática y TICs. Su E-Gov permite realizar gran mayoría de los trámites desde el portal y solo 3 requieren la presencia del ciudadano. Aun así, sigue avanzando, definiendo y desarrollando etapas nuevas.

Uruguay, un país vecino, es el único en Latinoamérica con puntaje EDGI alto y, recientemente, fue nombrado líder del grupo D9. Para el desarrollo de su e-Gov no se limitó a utilizar solo sus recursos disponibles, también adoptó las herramientas disponibles de la Computación en la Nube, logrando escalabilidad y un gran sistema hiperconvergente para su desarrollo.

En este trabajo, también se llevó a cabo un marco teórico que incluye todas las tecnologías y estándares posibles de utilizar para el desarrollo de un e-Gov. Los servicios web aportan al diseño interoperabilidad, que al emplearse en conjunto con los estándares básicos, permiten obtener una independencia significativa, seguridad, confiabilidad y calidad.

La implementación de un e-Gov de acuerdo al paradigma SOA, no solo aporta el beneficio de la reusabilidad de software, sino también la creación de sistemas altamente escalables que posibilitan la agilidad organizativa, la expansión y consolidación de las organizaciones.

La escalabilidad es fundamental para la expansión y el crecimiento del e-Gov y, a partir de los casos de estudio y del marco teórico, se pudo definir un concepto de escalabilidad para este ámbito. Una escalabilidad horizontal que aporta beneficios como alta disponibilidad de los recursos, funcionamiento continuo del sistema (aunque falle algún nodo), posibilidad de combinarse con un futuro escalamiento vertical y un crecimiento prácticamente infinito. Y una escalabilidad en la nube, que no solo permite una escalabilidad horizontal y vertical, sino que permite un desarrollo y una implementación paulatina de un e-Gov. Todo esto sustentado en una arquitectura e infraestructura actualizada, basadas en las redes NGN, ASON y en el protocolo IP, como principal, para lograr un sistema hiperconvergente.

La implementación de un e-Gov puede realizarse según las etapas del modelo Gartner y obtener las ventajas, descritas en el capítulo 10, paso a paso. El beneficio para el ciudadano debe ser siempre el objetivo principal del proyecto. Aunque, el éxito de la implementación del e-Gov dependerá de dos factores fundamentales: una decisión política concreta del Gobierno de turno y del nivel de alfabetización tecnológica de todos los ciudadanos. Ese fue el gran acierto de Estonia. .

Estonia de estar devastada de la guerra, con un 10% menos de población, sin estructura organizativa gubernamental y sin infraestructura propia, luego de tres décadas, hoy es el primer país digital del mundo y han logrado: un e-Gov con un 99% de trámites online, una conectividad segura y ágil tanto en el medio del mar como en el medio de las montañas, con el mejor ancho de banda y con el mejor sistema de seguridad y criptografía (desarrollado por ellos mismo). Es un gran desafío para el gobierno argentino lograr un avance tan significativo como éste.

Estos casos de estudio, y marco teórico, son una inspiración para el desarrollo del e-Gov en Argentina y una propuesta importante de cómo es posible empezar su implementación, aún con pocos recursos, para obtener un país tecnológicamente más desarrollado. El nivel y la calidad de formación de los profesionales en este país, sobre todo en ciencias, es reconocido a nivel mundial, y por qué no pensar en ello como la fuente de origen del futuro de Argentina.

El tema de este trabajo no está terminado, puede tomarse como iniciativa para otros trabajos y proyectos de investigación y desarrollo. Sobre todo la idea de una primera fase de implementación utilizando la computación en la nube y aprovechando los grandes beneficios de la misma. Un proyecto de E-Gov es muy extenso y requiere de una gama amplia de profesionales, con los que se podría decir, que Argentina cuenta.

ACRÓNIMOS

e-Gov	Gobierno Electrónico
AGESIC	Agencia para el Desarrollo del Gobierno de Gestión Electrónica y de la Sociedad de la Información y del Conocimiento
AGW	Access Gateway
AIG	Arquitectura Integrada de Gobierno
API	Application Programming Interface
ASON	Automatic Switched Optical Network
ASP	Active Server Pages
ATM	Asynchronous Transfer Mode
BAM	Business Activity Monitoring
BEA	Bureau of Economic Analysis
BI	Business Intelligence
BICSI	Building Industry Consulting Service International
BPEL	Business Process Execution Language
BPM	Business Process Modeling
BPM	Business Process Management
CCI	Connection Controller Interface
CCITT	Consultative Committee for International Telegraphy and Telephony
CDMA	Code Division Multiple Access
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
CP	Categoría de Performance
CRM	Customer Relationship Management
CSCF	Call State Control Function
DBMS	DataBase Management System
DESA	(United Nations) Department of Economic and Social Affairs
DNS	Domain Name System
DPADM	Division for Public Administration and Development Management
DSL	Digital Subscriber Line
DSP	Digital Signal Processor
DTD	Document Type Definition
DWDM	Dense Wavelength Division Multiplexing
EAI	Enterprise Application Integration
EGDI	E-Government Development Index
EJB	Enterprise JavaBeans
E-NNI	External Network-to-Network Interface
EPI	Electronic Participation Index
ERP	Enterprise Resource Planning
ESB	Enterprise Service Bus
FEC	Forwarding Equivalence Class
FSC	Fiber- Switch Capable
GDE	Gestión Documental Electrónica

GII	Global Information Infrastructure
GMPLS	Generalized Multi-Protocol Label Switching
GPF	Generic Framing Protocol
GPRS	General Packet Radio Service
HCI	Human Capital Index
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IaaS	Infrastructure as a Service
IARAF	Instituto Argentino de Análisis Fiscal
ICREA	International Computer Room Experts Association
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
I-NNI	Internal Network-to-Network Interface
IP	Internet Protocol
IP MS	IP Media Server
ISAGEL	Índice de Servicios y Asistencia Gubernamental en Línea
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
ISP	Internet Service Provider
ITU-T	International Telecommunications Union – Telecommunication Standardization Sector
IVR	Interactive Voice Response
JSON	Java Scripts Object Notation
JSP	Java Server Pages
KSI	Keyless Signature Infrastructure
LCAS	Link Capacity Adjustment Scheme
LDP	Label Distribution Protocol
LER	Label Edge Router
LIB	Label Information Base
LSC	Lambda Switch Capable
LSP	Label Switched Path
LSR	Label Switching Router
MPLS	Multi-Protocol Label Switching
MRA	Modelo de Referencia de Aplicaciones
MRD	Modelo de Referencia de Datos
MRDe	Modelo de Referencia de Desempeño
MRF	Multimedia Resource Function
MRN	Modelo de Referencia de Negocio
MRS	Modelo de Referencia de Seguridad
MRT	Modelo de Referencia Tecnológico
MSA	Micro-Services Architecture
NGN	Next Generation Networks
NII	National Information Infrastructure

NMI	Network Management Interface
NPR	National Performance Review
OASIS	Organization of the Advancement Structured Information Standards
OCDE	Organización para la Cooperación y el Desarrollo Económicos
ONU	Organización de las Naciones Unidas
OSI	Online Service Index
OSI	Open System Interconnection
OTAN	Organización del Tratado del Atlántico Norte
OXC	Optical Cross Connect
PaaS	Platform as a Service
PDI	Plataforma de Interoperabilidad
PGE	Plataforma de Gobierno Electrónico
PSC	Packet Switch Capable
PSL	Provisión de Servicios en Línea
PSTN	Public Switched Telephone Network
QoS	Quality of Service
REST	Representational State Transfer
RIKS	RIIGI INFOKOMMUNIKATSIOONI SIHTASUTUS (En inglés: State Infocommunication Foundation)
RPC	Remote Procedure Call
RSVP	Resource Reservation Protocol
SA	Server Application
SaaS	Software as a Service
SDH	Synchronous Digital Hierarchy
SGML	Standard Generalized Markup Language
SI	Sociedad de la Información
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SOA	Arquitectura Orientada a Servicios
SOAP	Simple Object Access Protocol
SQA	Software Quality Assurance
SS7	Signaling System 7
SSL	Secure Socket Layer
STM	Synchronous Transport Module
TDM	Time Division Multiplexing
TDMC	Time Division Multiplexing Capable
TGW	Terminating Gateway
TIA	Telecommunications Industry Association
TIC	Tecnologías de la Información y la Comunicación
TII	Telecommunication Infrastructure Index
TLS	Transport Layer Protocol
UDDI	Universal Description Discovery and Integration
UMTS	Universal Mobile Telecommunications System
UNeGovDD	United Nations E-Government Development Database

UNI	User-Network Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UTRAN	UMTS Terrestrial Radio Access Network
VC	Virtual Container
VCAT	Virtual Concatenation
W3C	World Wide Web Consortium
WDM	Wavelength Division Multiplexing
WLAN	Wireless Local Area Network
WS	Web Services
WS-BPEL	Web Service – Business Process Execution Language
WS-CDL	Web Service – Choreography Description Language
WSDL	Web Services Description Language
WSFL	Web Service Flow Language
WS-I	Web Services Interoperability
WSIO	Web Services Interoperability Organization
WWW	World Wide Web
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language
XSD	XML Schema Definition

BIBLIOGRAFIA

Libros

- Beltrán, M., Sevillano, F., 2013. Cloud Computing – Tecnología y negocio. Editorial Ediciones Nobel, S.A.
- Cruz del Valle, Ernesto 2017. Ejecutivos de la Informática - Temario Materias Específicas. Editorial Lulu.com.
- Erl T., 2005. Service-Oriented Architecture: Concepts, Technology And Design. Edit. Pearson Education.
- Erl T., 2007. SOA Principles of Service Desing. Edit. Prentice Hall.
- Gartner Group. BAUM, C., & DI MARIO, A. , 2000. “Gartner's four phases of e-Government model”.
- Márquez, 2007. La Web Semántica. Editorial Lulu.com
- Silberschatz, A., Korth, H., Sudarshan, S. 2002. Fundamentos de Bases de Datos – Cuarta edición. 787 páginas. Editorial McGrawHill.

Revistas científicas y Trabajos de Tesis de grado y Postgrado

- Alvarez Ballestas, K. El. Y Rodelo Calvo, L.A., 2008. Estado del arte de SOA, principios, técnicas, tecnologías, patrones y metodologías. Universidad Tecnológica de Bolívar. <http://biblioteca.unitecnologica.edu.co/notas/tesis/0043898.pdf>. consultado noviembre de 2018.
- Cardona, D., 2002. El gobierno electrónico: Una revisión desde la perspectiva de la prestación de servicios. I Congreso Catalán de Gestión Pública, Barcelona, 26 y 27 de septiembre.
- Consortio del proyecto Atlántico, 2003. La Administración Pública. España.
- DCCIA, 2014. Introducción a los Servicios Web. Invocación de servicios web SOAP – Departamento de Ciencia de la Computación e Inteligencia Artificial. Universidad de Alicante. <http://www.jtech.ua.es/j2ee/publico/servc-web-2012-13/sesion01-apuntes.html#WSDL>.
- García I., Polo M., Ruiz F., Piattini M., 2005 – Servicios Web – Informe Técnico UCLM DIAB-05-05-1 – Departamento de Informática – Universidad de Castilla. <http://www.info-ab.uclm.es/descargas/technicalreports/DIAB-05-01-1/Servicios%20Web.pdf>.
- Gil Aros, 2009. Los web services y características de calidad. Revista N° 10, Avances, Investigación en Ingeniería. . http://www.unilibre.edu.co/revistaavances/avances_10/r10_art7.pdf . Consultado el 20 noviembre de 2018.
- Rico-Bautista, D., Arévalo Ascanio J.G. y Bayona Trillo, R. A., 2017. Tecnologías de Banda Ancha. https://www.researchgate.net/figure/Arquitectura-genrica-de-NGN_fig2_283053635. Consultado en noviembre 2018.
- Rivera Rubio, J. M., 2008. INTRODUCCIÓN TEÓRICA AL IP MULTIMEDIA SUBSYSTEM, IMS .<http://bibing.us.es/proyectos/abreproy/11794/>. Consultado en noviembre de 2018.
- Siva Rama Prasad R. S. y RaghavaRao Atukuri V., 2012. Cloud Computing Technology for effective e-Governance. (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (1) , 2012,3241-3244. ISSN 0975 9646.
- Tricas Lamana, F., 2007. El gobierno electrónico: servicios públicos y participación ciudadana. Documento de trabajo 109/2007, Fundación Alternativas, ISBN: 978-84-96653-56-6.

Documentos Oficiales electrónicos consultados por internet

- Boletín Oficial, 2016. <http://servicios.infoleg.gob.ar/infolegInternet/verNorma.do?id=259082>. Consultado el 18 de octubre de 2018.
- CLAD, 2007. Carta Iberoamericana de E-Gov. <http://old.clad.org/documentos/declaraciones/cartagobelec.pdf>. Consultado el 23 de setiembre de 2018.
- Cubo Velazquez, 2018. SOAP y WSDL. <http://bibing.us.es/proyectos/abreproy/11247/fichero/Memoria%252F7-SOAP+y+WSDL.pdf>. Consultado Noviembre de 2018.
- EURLex, 2003. Documento 52003DC0567. <https://eur-lex.europa.eu/legalcontent/EN/TXT/?uri=celex:52003DC0567>. Consultado el 23 de setiembre de 2018.
- GPO, 2002. www.gpo.gov/fdsys/pkg/PLAW-107publ347/pdf/PLAW-107publ347.pdf. Consultado el 23 de setiembre de 2018.
- IARAF, 2017. <https://www.iaraf.org/index.php/informes-economicos/publicaciones-otras-areas-de-estudio/192-informe-economico-n-387>. Consultado el 05 de noviembre de 2018.
- IETF, 2014. RFC 7231 – Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. <https://tools.ietf.org/html/rfc7231>.
- OASIS, 2004. UDDI Version 3.0..2 – UDDI Spec Technical Committee Draft, Dated 20041019. <https://www.oasis-open.org/committees/uddi-spec/doc/spec/v3/uddi-v3.0.2-20041019.html>
- OCDE, 2003. The e-government imperative: main findings, Policy Brief OECD Observer, <http://unpan1.un.org/intradoc/groups/public/documents/APCITY/UNPAN015120.pdf>
- Reporte UN, 2004. UN GLOBAL E-GOVERNMENT READINESS REPORT 2004. <https://publicadministration.un.org/egovkb/portals/egovkb/Documents/un/2004-Survey/Complete-Survey.pdf>. Consultado el 24 de setiembre de 2018.
- Ruelas A., Arámburo P., 2006. El gobierno electrónico: su estudio y perspectivas de desarrollo. <http://www.razonypalabra.org.mx/anteriores/n52/27RuelasyAramburo.pdf>. Consultado el 30 de setiembre de 2018.
- The Open Group, 2018. Arquitectura Orientada a Servicios. http://www.opengroup.org/soa/source-book/soa/p1.htm#soa_definition. Consultado el 15 de octubre de 2018.
- UN E-Government Knowlegdebase, 2018. “UN eGovernment survey 2018”. <https://publicadministration.un.org/egovkb/en-us/Reports/UN-E-Government-Survey-2018> Consultado el 17 de noviembre de 2018.
- VOORMANN, 2006. Servicios Web. https://es.wikipedia.org/wiki/Servicio_web. Consultado el 18 de octubre de 2018.
- W3C, 1999-2017. RFC 2616 – Protocolo de transferencia de hipertexto - HTTP/1.1. <https://www.w3.org>

Portales web y otras referencias de internet

- ADSLFAQS, 2018. <https://www.adslfaqs.com/>. Consultado noviembre de 2018.
- AGESIC, 2018. Agencia para el Desarrollo del Gobierno de Gestión Electrónica y la Sociedad de la Información y del Conocimiento. https://www.agesic.gub.uy/innovaportal/v/1496/1/agesic/plataforma_de_gobierno_electronico_.html?menuderecho=5. Consultado el 27 de Noviembre de 2018.
- Arcitura, 2018. <http://serviceorientation.com/>. Consultado en octubre de 2018.
- Bitec, 2018. www.bitec.es/servicios-cloud/adoptando-la-nube-nube-hibrida-privada-iaas-paas-o-saas/. Consultado diciembre 2018.-

- CISCO, 2015. Arquitectura de Internet. <http://ciscobgl.blogspot.com/p/arquitectura-de-internet-ccna.html>. Consultado noviembre de 2018.
- Desarrolloconsoa, 2014. <http://desarrolloconsoa.blogspot.com/2014/02/conceptos-basicos-de-servicios-web-soap.html>, consultado el 30 de noviembre de 2018.
- Diario DPI, 2016. E-Government: la Situación en Argentina. <http://dpicuantico.com/sitio/wp-content/uploads/2016/04/doctrina-dyt-nro-15.pdf>. Suplemento N° 15. Consultado el 25 de octubre de 2018.
- e-Estonia, 2018. Servicios de interoperabilidad. <https://e-estonia.com/solutions/interoperability-services/>. Consultado el 28 de noviembre de 2018.
- El País, 2018. Estonia, el primer país digital del mundo. https://elpais.com/elpais/2018/04/05/eps/1522927807_984041.html. Consultado el 27 de noviembre de 2018.
- Escobar G., Passerini N., Arias J. y Blanco S., 2006. Arquitecturas de Integración. http://apit.wdfiles.com/local--files/start/03_apit_integracion.pdf. Consultado el 15 de octubre de 2018.
- Interoute, 2018. <https://www.interoute.es>. consultado diciembre 2018.-
- La Nación, 2018. Estonia, el primer país digital. <https://www.lanacion.com.ar/2146490-estonia-el-primer-pais-digital>. Consultado el 27 de noviembre de 2018.
- Infobae, 2018. <https://www.infobae.com/economia/2018/08/09/ya-no-sera-necesario-llevar-documentacion-en-papel-para-hacer-tramites-en-organismos-del-estado/>. Consultado el 9 de agosto de 2018.
- OEA-Organización de los Estados Americano. <http://portal.oas.org/Portal/Sector/SAP/DptodeModernizaci%C3%B3ndelEstadoyGobenabilidad/NPA/SobreProgramadeeGobierno/tabid/811/Default.aspx>. Consultado el 23 de setiembre de 2018.
- OSI-españa, 2014. <https://www.osi.es/es/actualidad/blog/2014/02/28/que-pasa-si-una-pagina-web-no-utiliza-https>. consultado el 30 de noviembre de 2018.
- Paredes, Francisco, 2018. XML – Websarrolladores. <http://websarrolladores.com/2018/09/10/xml/#more-657>.
- Rojas D., 2009. Capas del Modelo Conceptual de SOA. <http://icomparable.blogspot.com/2009/01/capas-del-modelo-conceptual-de-soa.html>. Consultado el 16 de octubre de 2018.
- Portal de Canadá, 2017. <https://digital.canada.ca/beginning-the-conversation/full-report/> Consultado el 21 de noviembre de 2018.
- PowerData, 2017. Qué es SOA y cuál es su diferencia con los Microservicios?. <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/que-es-soa-y-cual-es-su-diferencia-con-los-microservicios>. Consultado octubre 2018.
- Xertica, 2018. <https://www.xertica.com/blog/computacion-en-la-nube-definicion-ejemplos-y-usos-practicos>. Consultado diciembre de 2018.
- Castillo, Echenique y García, 2017. Trabajo Monográfico de “DISEÑO DE INTEGRACIÓN DE TECNOLOGÍAS” – Materia: Sistemas de Telecomunicaciones – Ingeniería en Telecomunicaciones – UCASAL.