



PRESIDENCIA DE LA REPÚBLICA



LINEAMIENTO DE DESARROLLO DE PROYECTOS DE SOFTWARE

Bogotá D.C. octubre 2017



TABLA DE CONTENIDO

INTRODUCCIÓN	3
1. OBJETIVO	3
2. ALCANCE	3
3. TÉRMINOS Y DEFINICIONES	3
4. CONSIDERACIONES SOBRE LA SEGURIDAD PARA LAS APLICACIONES WEB	6
5. PREFIJOS Y SUFIJOS	6
5.1 Reglas de generación:	6
5.2 Ejemplos de prefijos:	7
5.3 Ejemplos de sufijos	7
5.4 Estilo de escritura o sintaxis de los prefijos o sufijos.	7
6. NOMENCLATURA BASES DE DATOS	7
6.1 Bases de datos, nombres y rutas de archivo	7
6.2 Tablas y vistas	8
6.3 Campos	8
7. NOMENCLATURA DE PROGRAMACION	9
7.1 Definición de Variables	9
7.2 Definición de Funciones	9
7.3 Definición de Procedimientos	10
7.4 Definición de Clases	10
7.5 Definición de Objetos	10
8. TABLA RESUMEN NOMBRAMIENTOS	11
9. ACCESIBILIDAD Y USABILIDAD	12
9.1 Diseño Centrado en el Usuario.	12
9.2 Directrices de Usabilidad Web.	12
9.3 Diseño de Interfaz de Usuario.	13
9.4 Diseño de interacción	14
9.5 Contenido	15
10. INTEROPERABILIDAD	15
10.1 Plataforma de interoperabilidad	16
11. MARCO LEGAL	17
12. REQUISITOS TÉCNICOS	17
13. DOCUMENTOS ASOCIADOS	17
14. RESPONSABLE DEL DOCUMENTO	17



INTRODUCCIÓN

Este documento describe los estándares y lineamientos de codificación que son establecidos para la elaboración e interpretación de productos de software, que son desarrollados por el Área de Tecnologías y Sistemas de Información del Departamento Administrativo de la Presidencia de la República, los proveedores externos que desarrollen productos para la Presidencia, aplicando las mejores prácticas y la normatividad vigente. Todos los lineamientos internos deben seguir y complementar las estrategias de Gobierno en Línea lideradas por el Ministerio de Tecnologías de la Información y las Comunicaciones.

1. OBJETIVO

Unificar la metodología y el lenguaje utilizado en la construcción de productos de software que se despliegan en la plataforma de la Presidencia de la República. Mejorar la calidad de los desarrollos y la mantenibilidad de estos, dando cumplimiento a los principios y fundamentos de la Estrategia de Gobierno en Línea, como la innovación, confianza y seguridad.

2. ALCANCE

Los lineamientos aplican para el desarrollo de todos los proyectos de software que se estén ejecutando en la entidad. Deben ser acogidos por el Área de Tecnologías y Sistemas de Información y se debe reflejar en cada producto puesto en producción, ya sean nuevos desarrollos o ajustes de sistemas existentes. Los desarrollos de software deben cumplir con el **Manual de Gobierno en Línea** y el **Marco de referencia de arquitectura empresarial para la gestión de las Tecnologías de Información**.

3. TÉRMINOS Y DEFINICIONES

Assembly: se refiere a uno a más archivos que son la salida de la compilación de un programa. Fuente de esta información.

AssemblyInfo: se refiere aun a clase que provee atributos para obtener información sobre el programa, tal como número de versión, descripción, etc. Fuente de esta información.

Bases de datos: Se refiere a un conjunto de datos o archivos que tiene una estructura en común, la cual está organizada de tal forma que el ordenador pueda fácilmente encontrar la información.

CamelCase: Es un estilo de escritura que se aplica a frases o palabras compuestas y se asemejan a las jorobas de un camello. La letra inicial es minúscula. Ej: presidenciaDeLaRepublica, areaInformacionSistemas, grupoDesarrollo.

Casing: Se refiere a estilos de nomenclatura de palabras y oraciones de código.



C#: Es un lenguaje de programación orientado a objetos, desarrollado y estandarizado por Microsoft, forma parte de la plataforma .NET. Fuente de esta información.

Excepción: Se refiere a la indicación de un problema en la ejecución de un programa desarrollado, es poco frecuente, pero es importante contar con un manejo adecuado de estas. Fuente de esta información.

GEL-XML: Lenguaje común para que los diferentes sistemas de información puedan entenderse e intercambiar información de manera adecuada y eficiente. Define conceptos y la estructura de los servicios de forma que puedan ser entendidos por todos los involucrados.

GEL-POINT: Conjunto de principios, políticas y estándares para la efectiva implementación de interoperabilidad en los sistemas de los diferentes actores que prestan y/o demandan trámites y/o servicios del Estado Colombiano.

Indentación: Se refiere a mover un bloque de texto hacia la derecha con espacios o tabulaciones para distinguirlo del texto adyacente. Fuente de esta información.

keyword internal: Se refiere a una palabra clave que referencia a un modificador de acceso a los tipos y miembros de una clase que pertenecen a un mismo conjunto de Librerías y clases. Fuente de esta información.

kebab-case: nomenclatura que consiste en escribir todas las palabras en letras minúsculas separadas con un guion: Ejm: presidencia-de-la-republica.

Loop: se refiere a un bucle de programación. Fuente de esta información.

Namespace: Se refiere a un sistema de organización lógico que proporciona una forma de presentación de los elementos del programa que son expuestos hacia otros. Fuente de esta información.

Namespace System: Se refiere a un sistema de organización lógico que contiene clases base que definen tipos de datos de uso común, eventos, controladores de eventos, interfaces, atributos y excepciones de procesamiento. Fuente de esta información.

Pascal Casing: Se refiere a guías de nomenclatura de código fuente. Fuente de esta información.

PascalCase: Es un estilo de escritura que se aplica a frases o palabras compuestas y se asemejan a las jorobas de un camello. La letra inicial es mayúscula. Ej: PresidenciaDeLaRepublica, ArealInformacionSistemas, GrupoDesarrollo.

Pascal_Snake_Case: combina la nomenclatura PascalCase y snake_case, cada palabra debe iniciar con mayúscula y además deben estar separadas por el guion bajo. Ej: BDS_Proyectos_Desarrollo, VWS_Planillas_Pago, PLN_PK_Planilla.



SEO: La Sigla SEO (Search Engine Optimization) se refiere a la optimización que se debe aplicar a un sitio web , para garantizar su aparición en los primeros resultados de los motores de búsqueda.

snake_case: Es un estilo de escritura que se aplica a frases o palabras compuestas. Las letras son minúsculas y las palabras están separadas por un guion bajo (_). Ej: presidencia_de_la_republica, area_informacion_sistemas, grupo_desarrollo.

SCREAMING_SNAKE_CASE: Es un estilo de escritura que se aplica a frases o palabras compuestas. Las letras son mayúsculas y las palabras están separadas por raya al piso. Ej: PRESIDENCIA_DE_LA_REPUBLICA, AREA_INFORMACION_SISTEMAS, GRUPO_DESARROLLO.

String: Tipo de dato que representa una serie de caracteres Unicode. Fuente de esta información.

String.Equals: Se refiere a un método que determina si dos cadenas de texto tienen el mismo valor. Fuente de esta información.

String.Empty: Se refiere a un método que representa una cadena de texto vacía. Este campo es de solo lectura. Fuente de esta información.

StringComparison: se refiere a un método de tipo enumerador, el cual es utilizado para especificar si la comparación de una cadena de texto debe usar la referencia actual cultural, palabras o reglas de ordenación, y distinción entre mayúsculas o minúsculas. Fuente de esta información.

StringBuilder: Se refiere a una clase que representa una cadena de texto como un objeto. Fuente de esta información.

Sockets: Se refiere a un método para la comunicación entre un programa de cliente y un programa del servidor en una red. Fuente de esta información.

Streams: Se refiere a una técnica para transferir datos que pueden ser procesados como un flujo regular y continuo. Fuente de esta información.

Throw: Se refiere a la palabra clave (Keyword Internal), que se utiliza para señalar la aparición de una situación o evento anómalo durante la ejecución de un programa. Fuente de esta información.

Unicode: se refiere a un estándar de codificación de caracteres con la intención de representar texto de múltiples lenguajes, el término Unicode proviene de las palabras: universalidad, uniformidad y unicidad. Fuente de esta información.

Using: Se refiere a la palabra clave (Keyword Internal) la cual permite el uso de tipos en un espacio de nombres (namespace). Fuente de esta información.



4. CONSIDERACIONES SOBRE LA SEGURIDAD PARA LAS APLICACIONES WEB

Las aplicaciones deben realizarse atendiendo el marco que aplique a cada desarrollo dentro de lo que establece el estándar de seguridad de Aplicaciones Web denominado OWASP, mediante el cual se publica y mantiene actualizado tanto una guía para el desarrollo de aplicaciones seguras, como un listado de los diez riesgos de seguridad más importantes sobre los cuales recomienda a las organizaciones orientar la prevención y protección en el código que contienen sus aplicaciones.

- Toda la información en detalle acerca del proyecto y el estándar, puede consultarse en el sitio web oficial:

https://www.owasp.org/index.php/Sobre_OWASP

- El documento “Guía para Construir Aplicaciones y Servicios Web Seguros” liberado, con la versión más reciente puede consultarse y descargarse- aquí:

https://www.owasp.org/images/b/b2/OWASP_Development_Guide_2.0.1_Spanish.pdf

- El documento guía en su versión borrador más actualizado puede consultarse y descargarse aquí:

<http://owasp.cvs.sourceforge.net/viewvc/owasp/guide/current%20draft.pdf>

- El listado actualizado, sus históricos y candidatos para liberación pueden consultarse aquí:

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

5. PREFIJOS Y SUFIJOS

Es obligatorio utilizarlos para denominar cada tipo de elemento identificado como un artefacto de desarrollo, su función es identificar diferentes tipos de objetos, datos, entradas o salidas. Se puede aplicar para:

- Directorios y estructura de archivos.
- Archivos
- Clases, Interfaces, Servicios, Formularios, Apis, Enumeraciones, Etc.
- Proyectos, Soluciones, Espacios de Nombres.

5.1 Reglas de generación:

Los prefijos y sufijos deben contener exactamente 3 letras.

Si la palabra tiene al menos 3 consonantes, se escogen las primeras 3 consonantes, ejemplo:



- Desarrollo: dsr
- Metodología: mtd
- Trabajo: trb

Si la primera palabra tiene menos de 3 consonantes y empieza por una vocal, se escoge la primera vocal y las siguientes consonantes o vocales.

- Anual: anl
- Opera: opr
- Único: unc

Si la palabra tiene menos de 3 consonantes y empieza por una consonante, se escogen las consonantes y vocales en el orden que aparezcan.

- Tema: tem
- Cueva: cuv
- Trio: tri

5.2 Ejemplos de prefijos:

- fnt indicando que se trata de una función
- int, dbl, bol, indicando el tipo de datos de una variable.
- txt, cmb, ddl, opt, gvw, indicando el tipo de componente
- TBL, BDS, VWS, indicando el tipo de objeto de Base de Datos.

5.3 Ejemplos de sufijos

- lst, obj, int, str indicando que una el resultado de una función es una lista de objetos o la instancia de un objeto.

5.4 Estilo de escritura o sintaxis de los prefijos o sufijos.

Los prefijos y sufijos se toman como una nueva palabra adicionada al elemento que se quiere describir y por lo tanto deben respetar la sintaxis del elemento.

6. NOMENCLATURA BASES DE DATOS

6.1 Bases de datos, nombres y rutas de archivo

El nombre de la base de datos debe venir precedido del prefijo BDS (Base de datos) y seguido por el carácter (_) underline.



El nombre debe reflejar de manera precisa y exacta el contenido y su función. La nomenclatura utilizada es **Pascal_Snake_Case**: No se deben utilizar espacios en el nombre de archivo.

El nombre del archivo debe ser igual al nombre de la base de datos:

PREFIJO	EJEMPLO
BDS_	[Servidor].[Schema].[BDS_Plan_Compras] [Servidor].[Schema].[BDS_SIG]

6.2 Tablas y vistas

El nombre de la tabla debe venir precedido del prefijo **TBL (Tabla)** y seguido por el carácter (_) underline.

Los nombres de las tablas deben reflejar de manera precisa su contenido y función. Se recomienda la notación **Pascal_Snake_Case**: No se deben utilizar espacios en el nombre de archivo.

Los nombres de las vistas siguen las mismas convenciones que los nombres de las tablas. Se recomienda utilizar el prefijo **"VWS"**.

PREFIJO	EJEMPLO
TBL_	[Servidor].[Schema].[Base_Datos].[TBL_Cursos] [Servidor].[Schema].[Base_Datos].[TBL_Pagos]

6.3 Campos

Las siguientes recomendaciones aplican a todos los campos.

El nombre de los campos deben venir precedido del prefijo **PK o FK**, seguido por el carácter (_) underline.

Cada nombre de campo debe ser único dentro de su tabla correspondiente.

Cada nombre de campo debe derivarse del identificador utilizado durante el proceso de análisis de negocio.

No se deben utilizar palabras reservadas como nombres de campos.

Campos	PREFIJO	EJEMPLO
Campos propios	[servidor].[sch_Plan_Compras].[BDS_Plan_Compras].[TBL_Pagos] [servidor].[sch_Plan_Compras].[BDS_Plan_Compras].[TBL_Contabilidad] [servidor].[sch_Plan_Compras].[BDS_Plan_Compras].[TBL_Personas]	PGS_Descripcion CTB_Valor_Moneda PSN_Observaciones



7. NOMENCLATURA DE PROGRAMACION

7.1 Definición de Variables

Comience el nombre de la variable con una letra.

La primera letra de cada palabra en mayúscula.

Debe describir claramente el propósito de la variable.

Usar un prefijo que indique el tipo de dato.

Usar comentarios que describan el ámbito en el que se encuentran definidas las variables.

La nomenclatura que se debe usar es **camelCase**

TIPO DE DATO	PREFIJO	EJEMPLO
Bool, Booleano (boolean)	bol/bln	bolEsMenor
Byte	byt	bytEdad
(fecha-hora) date-time	dtm/dat	datFechaInicial
doble (double)	dbl	dblValorDescuento
Int, Integer (entero)	Int	intEdad
Long (entero largo)	Ing	IngTotalEmpleados
Char (caracteres unicode)	chr	chrNombre
String (cadena de caracteres)	str	strNombre
Float	flt	fltTotalDescuento
Decimal	dcm	dcmToneladas

7.2 Definición de Funciones

Usar el prefijo: fnt

Usar el sufijo: tipo de dato que retorna, deben ser tan largos como sea necesario para describir su funcionalidad, la Primera letra del nombre en Mayúscula. Nomenclatura **camelCase**.

Ej: *fntNombreFuncionInt, fntCalcularSumaDbl, fntResolverNombreStr*

Si la función no retorna nada, no se usará sufijo.

Ej: *fntNombreFuncion, fntNotificarUsuario*



7.3 Definición de Procedimientos

Usar el prefijo *fnt*, Primera letra del nombre en mayúscula. Incluir además de la acción, la entidad que será devuelta: Verbo-sustantivo (*CalcularDesperdicio()*), nomenclatura **camelCase**.

Ej: fntCalcularDesperdicio

7.4 Definición de Clases

Usar el prefijo *Cls*. La primera letra de cada palabra en mayúscula. Nomenclatura **PascalCase**.

Ej: ClsCuentaBancaria

7.5 Definición de Objetos

Utilizar prefijos para definir los objetos:

Prefijo	Tipo y Significado	Ejemplos
lbl	Label	lblNombre
lnk	LinkLabel	lnkNombre
btn	Button	btnNombre
txt	TextBox	txtNombre
mnm	MainMenu	mnmNombre
chk	CheckBox	chkNombre
rdb	RadioButton	rdbNombre
gbx	GroupBox	gbxNombre
pct	PictureBox	pctNombre
pnl	Panel	pnlNombre
dtg	DataGrid	dtgNombre
lbx	ListBox	lbxNombre
cbx	ComboBox	cbxNombre
lsv	ListView	lsvNombre
tvw	TreeView	tvwNombre
gvw	GridView	gvwNombre
tbc	TabControl	tbcNombre
prg	ProgressBar	prgNombre
rtb	RichTextBox	rtbNombre
img	ImageList	imgNombre
stb	StatusBar	stbNombre
dtp	DateTimePicker	dtpNombre
ddl	DropDownList	ddlNombre
hdf	HiddenField	hdfNombre
cld	Calendar	cldNombre
rep	Repeater	repNombre



rfv	RequiredFieldValidator	rfvNombre
smg	ScriptManager	smgNombre
upl	UpdatePanel	uplNombre
ups	UpdateProgress	upsNombre
crv	CrystalReportViewer	crvNombre

8. TABLA RESUMEN NOMBRAMIENTOS

	DESCRIPCIÓN	PREFIJO	EJEMPLO
VARIABLES	Bool (boolean)	bol	bolEsMenor
	Byte	byt	bytEdad
	Date (date-time)	dtm	dtmFechaInicial
	Double (double)	dbl	dblValorDescuento
	Integer (entero)	int	intEdad
	Long (entero largo)	lng	lngTotalEmpleados
	Char (caracteres)	chr	chrNombre
	String (cadena)	str	strNombre
	Float	flt	fltTotalDescuento
	Decimal	dec	decToneladas
FUNCIONES	Sin retorno	fnt	fntNombre
	Retorna Int	fnt	fntNombreInt
	Retorna Objeto	fnt	fntNombreObj
	Retorna Double	fnt	fntNombreDbI
	Retorna String	fnt	fntNombreStr
PROCEDIMIENTOS	Procedimientos	fnv	fnvNombreProcedimiento
CLASES	Clases	cls	ClsNombreClase
OBJETOS	Label	lbl	lblNombre
	LinkLabel	lnk	lnkNombre
	Button	btn	btnNombre
	TextBox	txt	txtNombre
	MainMenu	mnm	mnmNombre
	CheckBox	chk	chkNombre
	RadioButton	rdb	rdbNombre
	GroupBox	grp	grpNombre
	PictureBox	pct	pctNombre
	Panel	pnl	pnlNombre
	DataGrid	dtg	dtgNombre
	ListBox	lbx	lbxNombre
	ComboBox	cbx	cbxNombre
	ListView	lsv	lsvNombre
	TreeView	tvw	tvwNombre
	GridView	gvw	gvwNombre
	TabControl	tbc	tbcNombre



	ProgressBar	prg	prgNombre
	RichTextBox	rtb	rtbNombre
	ImageList	img	imgNombre
	StatusBar	stb	stbNombre
	DateTimePicker	dtp	dtpNombre
FORMULARIOS	WEB	frm	frmNombreFormulario
	MAESTRA	mtr	mtrNombreMaestra
REPORTES	Reportes	rpt	rptNombreReporte
WEBSERVICES	WebServices	wsv	wsvNombreWebService
SOLUCIONES	Soluciones	sln	slnNombreSolución
PROYECTOS	Proyectos	prj	prjNombreProyecto
ESTILOS	Estilos	sty	styNombreEstilo

9. ACCESIBILIDAD Y USABILIDAD

Siguiendo los Lineamientos y metodologías de Usabilidad para Gobierno en línea. Se debe garantizar el acceso universal a la Web, independientemente del tipo de hardware, software, infraestructura de red, idioma, cultura, localización geográfica y capacidades de los usuarios para lo cual se debe tener en cuenta las siguientes recomendaciones:

9.1 Diseño Centrado en el Usuario.

En términos generales, el diseño centrado en el usuario es una filosofía y proceso de diseño en el que las necesidades, los deseos y las limitaciones del usuario final de una interfaz o documento toman una atención y relevancia considerable en cada nivel del proceso de diseño.

9.2 Directrices de Usabilidad Web.

Las directrices más relevantes, ayudan a tener un portal, sitio o aplicación orientada al usuario con información presentada de forma clara y concisa.

Objetivos del Portal Web: Las necesidades concretas de la organización y lo que esta espera conseguir a través del sitio web se ven plasmadas en los objetivos. Serán estos, en armonía con las necesidades de los usuarios, los que guiarán todo el proceso de conceptualización, diseño e implementación del portal.

Navegación global consistente. Proporcione una navegación global visible y consistente a lo largo del sitio web.

URL limpios. El URL es el la abreviatura de (Uniform Resource Locator) Localizador Uniforme de Recursos, es el nombre técnico con el que se conoce a la comúnmente llamada —direcciónII.

Un esquema de URL limpios tiene las siguientes características:

- Corresponde a la jerarquía del sitio



- No incluye caracteres especiales como \$, &, ?, = entre otros.

9.3 Diseño de Interfaz de Usuario.

Ubicación del logotipo: Ubique el logotipo de la entidad en el mismo lugar en todas las páginas y vincúlelo con la página de inicio.

Interfaces en movimiento. Evite diseñar interfaces en movimiento. Las posibilidades de movimiento de elementos de las interfaces, a partir de tecnologías como Flash, animaron a algunos diseñadores a proponer páginas web más llamativas visualmente, por la misma novedad que supone la diferencia. Sin embargo, este recurso conlleva dificultades, particularmente en la facilidad de uso y en la accesibilidad. El siguiente es un claro ejemplo, a través de un sitio de venta de repuestos para computadores.

Contraste en brillo y color: Verifique que el texto y las imágenes de texto tengan suficiente contraste de brillo y color con el fondo.

Información transmitida a través de color: Debido a que un porcentaje de la población (8% de hombres y 1% de mujeres) tienen una percepción visual diferente, llamado daltonismo, el diseño de las páginas web debe asegurar que cuando se transmita información a través del color, se provean mecanismos alternativos para que toda la población entienda.

Esta medida beneficiaría no solo a la población con daltonismo, sino también a personas que utilicen pantallas con deficiencia cromática, monocromáticas o en condiciones donde la percepción de luz pueda verse alterada.

Justificación del texto: Evite la alineación justificada del texto de prosa al margen izquierdo y derecho a la vez. El diseño tipográfico que contempla la alineación justificada (margen derecho e izquierdo al mismo tiempo), es una práctica habitual en la industria de los impresos (libros, revistas, folletos).

Ancho del cuerpo de texto. Utilice un ancho promedio entre 60 y 80 cpl (caracteres por línea) para el cuerpo de texto. Un renglón corto en cpl implica que el ojo tiene que realizar constantemente saltos de línea, que lo lleva rápidamente a cansarse. De otra parte, para un renglón demasiado largo la vista deba realizar un recorrido muy largo, produciendo de igual forma fatiga visual.

Para los sitios web del Estado colombiano, se ha seleccionado un tamaño medio, a partir de la investigación de García y Short (2002) que sugiere el renglón ideal entre 60-80 cpl.

Desplazamiento horizontal: La resolución mínima para monitores de computador que debe tenerse en cuenta en la actualidad para los sitios del Estado colombiano, es de 1024 x 768 pixeles. Sobre esta medida, el diseñador debe probar su diseño para que en ningún navegador aparezca la barra de desplazamiento horizontal.

Por ello se sugiere una medida de 960 pixeles de ancho, que encaja perfectamente en la resolución establecida en esta directriz y que no generará barra de desplazamiento horizontal.



Hojas de estilo para diferentes formatos: Asigne estilos para lectura en pantalla e impresión en papel.

Independencia de navegador: Aunque Internet Explorer sea actualmente el navegador más utilizado, esta tendencia cada día está a la baja, con la aparición de excelentes alternativas libres, como Firefox, Opera, Chrome y Safari, navegadores que cada día demuestran más rapidez en su desempeño y, sobre todo, más respeto por los estándares web.

Calidad del código: Verifique que el código HTML y CSS de su sitio cumpla estándares. Codificar de acuerdo con los estándares es en principio una forma de garantizar que el sitio web se verá adecuadamente en la mayoría de navegadores y dispositivos. Escribir código válido no solo es un indicador de calidad, sino una clara manifestación de que quien diseña el sitio hace un esfuerzo por generar un sitio interoperable y accesible.

Para verificar que el código se escribe de acuerdo con el estándar emitido por la W3C, existen diversas herramientas de validación.

9.4 Diseño de interacción

Campos obligatorios: Distinga claramente los campos obligatorios de los opcionales. Uno de los elementos que puede afectar la experiencia de usuario es una mala señalización de los campos de formulario. El usuario se ha habituado a encontrar una convención para aquellos campos que son requeridos, por eso en ocasiones los campos sin señalización pueden ser interpretados como opcionales.

Asociación de etiquetas y campos: Asocie claramente las etiquetas con los campos de formulario. Para que el usuario tenga una interacción fluida con los formularios se hace necesario que conozca los datos que el sistema espera que ingrese en cada campo. Un adecuado rotulado de los campos de formulario, permite una lectura rápida y un ingreso ágil de la información.

Validación dinámica de datos: Proporcione una validación dinámica de datos, antes de que el usuario envíe un formulario. Los lenguajes de programación ejecutados del lado del cliente pueden beneficiar mucho la experiencia de los usuarios al diligenciar formularios. Proporcionar una validación dinámica en línea antes de enviar el formulario completo hace más sencillo el flujo de trabajo y evita al usuario errores innecesarios.

Error de página no encontrada: Diseñe e implemente una página específica para el error de página no encontrada (404). Que no exista ningún tipo de error cuando los usuarios interactúan con un sitio web es algo difícil de garantizar. Un simple enlace mal formulado, una URL que cambia, un dato ingresado de manera incorrecta e incluso una acción equivocada del usuario pueden desencadenar en un error. Un error común es el de página no encontrada, conocido a veces por su código (404). Es un error que aparece cuando el usuario intenta ingresar a una página del sitio que no existe o ha sido eliminada.

Ejemplos en los campos de formulario: Proporcione ejemplos en los campos de formulario que sean de difícil comprensión. Los ejemplos son una forma sencilla y clara de orientar al usuario. El visitante entiende



claramente la forma de diligenciar un formulario a través de ejemplos, especialmente en aquellos campos que requieren algún formato especial.

Páginas de confirmación: Redacte páginas de confirmaciones claras e informativas. Uno de los principios claves para garantizar una buena experiencia de usuario es la retroalimentación. La interacción entre una persona y el ordenador se constituye en un diálogo, por tanto el usuario espera del computador una respuesta a cada acción realizada. Una buena retroalimentación es fundamental para que el usuario conozca en todo momento el estado del sistema, y además obtenga la seguridad de que éste se encuentra en funcionamiento.

9.5 Contenido

Listas: Prefiera el uso de listas ordenadas, listas sin orden y listas de definiciones, en vez de párrafo en prosa.

Escaneado de contenido: Estructure el contenido para que pueda ser fácilmente escaneado por cualquier tipo de usuario. Se sugiere entonces utilizar enfatizar palabras y frases a través de las etiquetas (énfasis) y (énfasis fuerte). Sin embargo, la mejor forma de estructurar el contenido, será precisamente el uso de las etiquetas adecuadas de HTML. Por ejemplo, los encabezados <h1>, las listas desordenadas , los párrafos <p>, las citas en bloque <blockquote>, las tablas <table>, etc.

Vínculos rotos: Asegúrese de que no existan vínculos rotos.

10. INTEROPERABILIDAD

El modelo de interoperabilidad de Colombia está diseñado para apoyar la implementación de la Estrategia de Gobierno en línea (GEL) que desarrolla el Ministerio TIC (Tecnologías de Información y las Comunicaciones), a partir de cuatro componentes: TIC para la gestión, TIC para servicios, TIC para el gobierno abierto y Seguridad y privacidad de la información.

Hasta el momento, el Programa Gobierno en línea ha identificado como los escenarios de interoperabilidad más comunes los siguientes:

- 1: Necesidad de modernizar procesos interinstitucionales
- 2: Su entidad requiere información de otra entidad.
- 3: Solicitud de otras entidades para compartir información.
- 4: Desarrollo o mantenimiento de una nueva solución informática.
- 5: Obtener voluntad política para el intercambio de información.
- 6: Aumento inesperado en la cantidad de información que debe intercambiar la entidad.

Si se identifica que la solución de software que se está desarrollando cae en cualquiera de los anteriores escenarios, se debe aplicar el marco de interoperabilidad GEL.



A continuación se debe evaluar el grado de madurez en el que se encuentra la entidad, inicial, definido u optimizado.

10.1 Plataforma de interoperabilidad

La Plataforma De Interoperabilidad – PDI Es el conjunto de herramientas necesarias que permite que los sistemas de información del Estado conversen entre sí mediante interfaces estándar de Comunicación entre procesos y sistemas de información, un Lenguaje común para el intercambio de información entre aplicaciones (GEL XML) y varias Políticas de Interoperabilidad (GEL-POINT), ver: <http://lenguaje.intranet.gov.co>.

IMPORTANCIA DEL ESTÁNDAR GEL-XML

Entre las ventajas de la adopción del estándar GEL-XML se encuentran:

- Elementos de dato definidos en consenso, en virtud a que independiente de la entidad responsable de la definición del mismo, el usuario perciba el mismo concepto de información.
- Información definida teniendo en cuenta aspectos normativos: constitución, leyes, decretos, actos administrativos, reglamentación interna, etc.
- Información definida con un solo significado.
- Información definida con una sola sintaxis, única forma de describir el elemento de dato técnicamente.
- Apropiación de conocimiento de estándares internacionales para el intercambio de información.
- Definición de la información de forma única sin importar el proveedor.
- Establecimiento de buenas prácticas en el intercambio de información.
- Evolución ordenada y controlada de los conceptos que hacen parte del estándar.
- Construcción de una visión a futuro en el desarrollo de los sistemas de información del Estado Colombiano.
- Eficiencia y transparencia de las entidades.
- Estrategia uniforme en las entidades para el intercambio de información.

BENEFICIOS

Teniendo en cuenta algunos de los aspectos que constituyen la problemática actual que se da en los procesos de intercambio de información, y los objetivos que se han trazado para el estándar GEL-XML, es posible identificar que éste PUEDE ofrecer:

- Entendimiento del negocio: información definida con las características requeridas para soportar la misión de las entidades y no soluciones técnicas particulares.
- Reutilización de la información: capacidad de incrementar el uso sinérgico de la información, de tal forma que soporte su administración mediante mecanismos que ayuden al desarrollo de la misión de las entidades.



- Intercambio de información: identificación de la información que se quiere compartir e intercambiar entre entidades, con el sector productivo y con otros países.
- Normalización de la información: disponer de un modelo confiable y único para definir los conceptos para el intercambio de información en el Estado Colombiano.
- Información con semántica: entender la información de forma conceptual sin tener en cuenta los elementos técnicos de los sistemas de información.
- Diccionario de conceptos: biblioteca de elementos definidos en el Estado que ayuda al desarrollo de nuevos sistemas de información.
- Interoperabilidad internacional: elementos desarrollados teniendo en cuenta estándares internacionales y utilizando elementos para temas específicos (salud, comercio electrónico, reporte de información financiera, etc.).
- Otros contextos de trabajo: si bien es cierto el estándar GEL-XML, se ha concebido para ser utilizado en la implementación de procesos electrónicos de intercambio de información, se puede considerar extender su aplicabilidad en dos ámbitos: definición, desarrollo e implementación de nuevos sistemas de información, y en la definición de protocolos de intercambio de información manual.

11. MARCO LEGAL

Se puede consultar en el aplicativo SIGEPRE – Mapa de procesos – Tecnología de Información y Comunicaciones – Normograma o en el campo de documentos asociados cuando se consulta el documento.

12. REQUISITOS TÉCNICOS

Norma Técnica ISO/IEC 23270:2006 especifica la forma y establece la interpretación de los programas escritos en el lenguaje de programación C#.

Norma Técnica Colombiana NTC 5854, Accesibilidad a Páginas Web.

C# Language Specification ECMA INTERNATIONAL Standard ECMA-334 4th edición – Junio 2006 ver: <https://www.iso.org/obp/ui/es/#iso:std:iso-iec:23270:ed-2:v1:en>.

13. DOCUMENTOS ASOCIADOS

Se puede consultar en el aplicativo SIGEPRE – Mapa de procesos – Documentos y formatos o en el campo de documentos asociados cuando se consulta el documento.

14. RESPONSABLE DEL DOCUMENTO

Jefe Área de Tecnologías y Sistemas de Información