



Introducción a la Aceptación Universal

Mark Svancarek y Luisa Villa

Acerca de este documento

Propósito

Las tecnologías de Internet, incluidos sus componentes de nombres, están en constante evolución y cambio. En los últimos años, la ICANN ha lanzado una gran cantidad de nuevos TLD con caracteres ASCII y dominios de alto nivel de IDN. Entre los ejemplos se incluye `.nyc`, `.hsbc`, `.eco` y `.ストア`. Sin embargo, la respuesta al cambio en el panorama de nombres no ha sido lo suficientemente rápida. Muchas aplicaciones y servicios no se están actualizando para administrar los nuevos TLD. Esto afecta la experiencia del usuario. Por ejemplo:

- no se están aceptando direcciones de correo electrónico válidas.
- Los nombres de dominio se tratan erróneamente como términos de búsqueda en la barra de direcciones del navegador.

Muchas aplicaciones y servicios no se están actualizando para administrar estos nuevos TLD. Esto afecta la experiencia del usuario.

A menos que el software reconozca y pueda procesar los nuevos dominios, un estado conocido como **Aceptación Universal**, no será posible proporcionar una experiencia coherente y positiva para los usuarios de Internet. Por tal motivo, el presente documento proporciona una extensa introducción a la Aceptación Universal para ayudar en el desarrollo de software listo para Aceptación Universal.

Destinatarios

- Desarrolladores de software
- Directores de Tecnologías (CTO)
- La comunidad técnica en general

Estructura del documento

- Parte 1 **Conceptos básicos de la Aceptación Universal**, como qué es un nombre de dominio y el Sistema de Nombres de Dominio, ASCII y Unicode, Punycode, internacionalización de direcciones de correo electrónico y otros conceptos básicos.
- Parte 2 Los **cinco criterios de Aceptación Universal** y las **prácticas recomendadas** para cada uno de estos criterios. También contiene **escenarios de usuarios** y **prácticas de no conformidad** para la Aceptación Universal, requisitos técnicos y desafíos actuales.
- Parte 3 **Temas avanzados** como códigos de escritura de derecha a izquierda, el algoritmo Bidi, normalización y conversión de mayúsculas y minúsculas.
- Parte 4 Contiene el **glosario** y **recursos en línea** útiles.

¿Necesita más información?

El UASG y la comunidad están disponibles para brindar asesoramiento a los desarrolladores de software y a los implementadores sobre lo que se necesita.

- **Contáctenos** para compartir sus ideas y sugerencias sobre el tema en info@uasg.tech
- **Participe en el debate sobre Aceptación Universal** en <http://tinyurl.com/ua-discuss>
- Para **obtener más información** acerca de la iniciativa, visite <http://www.icann.org/universalacceptance>

Índice

Introducción 5

- Una breve historia de la Internacionalización de los nombres de dominio..... 5
- La necesidad de la Aceptación Universal 5

Parte 1: Conceptos fundamentales de la Aceptación Universal..... 7

- Nombre de dominio..... 7
- Sistema de Nombres de Dominio (DNS) 7
- Dominios de Alto Nivel (TLD) 7
- Dominios Genéricos de Alto Nivel (gTLD) 8
- Conjuntos de caracteres y códigos de escritura 8
- ASCII y Unicode 8
- Nombres de Dominio Internacionalizados (IDN) y Punycode..... 9
- Correo electrónico 10
- Direcciones y la Internacionalización de las Direcciones de Correo Electrónico (EAI) 10
- Generación dinámica de vínculos (Vinculación) 11

Parte 2: La Aceptación Universal en acción..... 12

- Cinco criterios de Aceptación Universal..... 12
- Escenarios de usuarios 13
- No conformidad con las prácticas de la Aceptación Universal 15

Requisitos técnicos para la preparación de la UA 16

- Requisitos de alto nivel 16
- Consideraciones del desarrollador..... 17

 - Principio rector para lograr la Aceptación Universal: Ley de Postel..... 18
 - Prácticas recomendadas para desarrollar y actualizar el software para lograr la preparación para la Aceptación Universal 18

- Fuentes fidedignas para nombres de dominio 25

 - Zona Raíz del DNS 25
 - Lista de sufijos públicos 25

Otros desafíos..... 26

- General..... 26
- Correo electrónico estilo IDN y por qué no es lo mismo que EAI 26
- La vinculación y sus desafíos 27

Parte 3: Temas avanzados 29

- Códigos de escritura complejos 29

 - Idiomas de derecha a izquierda y conformidad con Unicode 29

Introducción a la Aceptación Universal (UASG 007)

El algoritmo Bidi.....	29
La regla Bidi para nombres de dominio.....	30
Uniones.....	30
Homóglifos y caracteres confusamente similares.....	31
Normalización y conversión de mayúsculas y minúsculas.....	32
Normalización.....	32
Conversión de mayúsculas y minúsculas.....	33
Parte 4: Glosario y otros recursos.....	35
Glosario.....	35
RFC.....	37
Normas clave.....	40
Recursos en línea.....	40
Reconocimientos.....	43
Cambios de versiones.....	44

Introducción

Una breve historia de la Internacionalización de los nombres de dominio

En la década de 1970, los caracteres disponibles para registrar nombres de dominio se limitaban a un subconjunto de caracteres **ASCII** (letras a-z, dígitos 0-9 y el guión "-"). Desde la registración .com más antigua, symbolics.com, en 1985, el número y las características de los nombres de dominio se han expandido para reflejar las necesidades del uso global cada vez mayor de Internet como recurso comunitario. Hoy en día, la mayoría de los usuarios de Internet no hablan inglés. Sin embargo, el idioma dominante que se utiliza en Internet es el inglés. Para ayudar con la internacionalización de Internet, en 2003, el Grupo de Trabajo en Ingeniería de Internet (**IETF**) comenzó a publicar normas que proporcionaban directrices técnicas para el despliegue de **Nombres de Dominio Internacionalizados (IDN)** a través de un mecanismo de traducción para admitir representaciones de nombres de dominio que no son ASCII en códigos de escritura locales geográficamente diversos (por ejemplo, 普遍接受-□□.世界, ua-test.世界, etc.).

La Junta Directiva de la Corporación de Internet para Nombres y Números Asignados (**ICANN**) aprobó el proceso para introducir nuevos Dominios de Alto Nivel Internacionalizados con Código de País (IDN ccTLD) en octubre de 2009, y los primeros IDN ccTLD se agregaron a la zona raíz en mayo de 2010. En junio de 2011, la Junta Directiva aprobó y autorizó el lanzamiento del **programa de nuevos Dominios Genéricos de Alto Nivel (gTLD)**, que incluía nuevos IDN TLD y ASCII. El primer lote de TLD de este programa se agregó a la zona raíz en 2013. La adición de IDN ccTLD y nuevos TLD ha aumentado drásticamente el ritmo en el que se agregan TLD a la zona raíz.

Una década después de que el IETF publicara sus directrices relacionadas con IDN, y gracias al Programa de Nuevos TLD de la ICANN, ahora se han lanzado más de mil TLD nuevos. Sin embargo, a pesar de todos estos esfuerzos, gran parte del software y muchas aplicaciones aún no están listos para la Aceptación Universal. Esto causa problemas a los usuarios de Internet, incluidos aquellos cuyos idiomas están escritos en códigos de escritura que incluyen caracteres que no son ASCII.

La necesidad de la Aceptación Universal

Para seguir el ritmo de este mundo de nuevos TLD, se debe crear nuevo software y se deben actualizar las aplicaciones y software antiguos. El estado de cumplir con éxito con este mundo de nuevos TLD se llama **Aceptación Universal**.

Aceptación Universal es el estado en el cual todos los nombres de dominio y direcciones de correo electrónico válidos son **aceptados, validados, almacenados, procesados y mostrados** correcta y uniformemente por todos los sistemas, dispositivos y aplicaciones habilitados para Internet. En otras palabras, cada dirección web válida se resuelve en el sitio web esperado y cada dirección de correo electrónico válida entrega el correo al destino esperado. Debido al panorama de nombres de dominio que evoluciona tan rápidamente, muchos sistemas no reconocen ni procesan apropiadamente nuevos nombres de dominio, principalmente porque pueden estar en un formato no ASCII, porque el software no tiene conocimiento del TLD recién publicado, o porque la longitud de sus TLD difiere. Lo mismo sucede con las direcciones de correo electrónico que incorporan estas nuevas extensiones.

Aceptación Universal es el estado en el cual todos los nombres de dominio y direcciones de correo electrónico válidos son aceptados, validados, almacenados, procesados y mostrados correcta y uniformemente por todos los sistemas, dispositivos y aplicaciones habilitados para Internet.

Introducción a la Aceptación Universal (UASG 007)

El **Grupo Directivo sobre Aceptación Universal (UASG)**, una iniciativa que abarca a toda la industria y dirigida por la comunidad que recibe el apoyo de la ICANN, está trabajando en crear concientización, identificar y resolver problemas asociados con la Aceptación Universal de Nombres de Dominio a fin de ayudar a garantizar una experiencia uniforme y positiva para los usuarios de Internet a nivel global.

Parte 1: Conceptos fundamentales de la Aceptación Universal

Esta sección contiene una descripción general de los términos y conceptos básicos necesarios para comprender antes de leer las secciones más avanzadas de este documento.

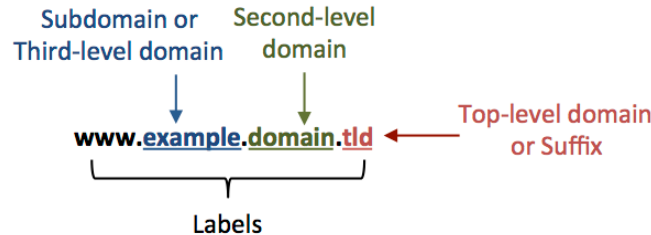
Nombre de dominio

Un nombre de dominio es una cadena de caracteres de texto punteada que se utiliza como identificador técnico amigable con el usuario para las computadoras y las redes en Internet. Por ejemplo:

`www.dominio.tld`

Cómo leer un nombre de dominio:

- Cada punto representa un **nivel** en la jerarquía del Sistema de Nombres de Dominio (DNS).
- Un Dominio de Alto Nivel (TLD) a menudo se denomina **sufijo** al final de un nombre de dominio.
- Las palabras individuales o los caracteres entre los puntos se llaman **etiquetas**. Para aquellos idiomas o códigos de escritura que se escriben de **izquierda a derecha (LTR)**,¹ la etiqueta más a la derecha representa el dominio de alto nivel.
- La segunda etiqueta desde el final representa el **dominio de segundo nivel**.
- Cualquier etiqueta que venga antes del dominio de segundo nivel se considera **subdominio** del dominio de segundo nivel (a veces denominada **dominio de tercer nivel**).



Sistema de Nombres de Dominio (DNS)

A cada recurso en Internet se le asigna una dirección para ser utilizada por el Protocolo de Internet (IP). Dado que las direcciones IP son difíciles de recordar, el DNS proporciona una correspondencia entre las direcciones IP y los nombres de dominio legibles para los humanos. Los servidores que proporcionan colectivamente un DNS público existen en direcciones conocidas en Internet.

Dominios de Alto Nivel (TLD)

Los nombres de dominio legibles para los humanos son administrados por organizaciones conocidas como **registros**. Cuando se registra un nombre de dominio, consta de múltiples cadenas de caracteres de texto que representan múltiples niveles de dominio, cada uno separado por un carácter ".". En las secuencias de códigos de escritura de LTR, el nivel de dominio más a la derecha

¹ Los idiomas o los códigos de escritura que se escriben de derecha a izquierda (RTL) se analizarán más adelante en este documento.

es el Dominio de Alto Nivel (TLD). Algunos TLD se delegan a países o territorios específicos. Estos se denominan **TLD con Código de País (ccTLD)**.

Dominios Genéricos de Alto Nivel (gTLD)

A partir de 2013, la ICANN (la organización responsable de la creación y el mantenimiento de las asignaciones de TLD) aprobó la creación de una gran cantidad de nuevos TLD. Estos nuevos TLD pueden representar marcas, comunidades de interés, comunidades geográficas (ciudades, regiones) y conceptos más genéricos. En su conjunto, todos estos nuevos TLD se conocen como Dominios Genéricos de Alto Nivel (gTLD).

Ejemplos de TLD comunes	Ejemplos de ccTLD	Ejemplos de nuevos gTLD
.com	China = .cn	.app
.gov	Alemania = .de	.lawyer
.info	Estados Unidos = .us	.shopping
.org		.panasonic
		.osaka

Conjuntos de caracteres y códigos de escritura

Los idiomas se escriben utilizando sistemas de escritura. La mayoría de los sistemas de escritura utilizan un código de escritura, que es un conjunto de caracteres gráficos que se utilizan para la forma escrita de uno o más idiomas. Una pequeña cantidad de sistemas de escritura emplea más de un código de escritura al mismo tiempo. Estos caracteres o códigos de escritura pueden ser reconocidos por los humanos. Sin embargo, no son útiles para las computadoras. En cambio, una computadora necesita que un código de escritura se codifique de una manera que pueda procesar (por ejemplo, para resolver una dirección web). El mecanismo para esto se denomina **mapeo de caracteres** o **conjunto de caracteres codificados (CCS)**, o **página de códigos**.² Un mapeo de caracteres asocia caracteres con números específicos. Muchas páginas de códigos diferentes se han creado a lo largo del tiempo para diferentes propósitos, pero para este tema nos enfocaremos solo en dos: ASCII y Unicode.

ASCII y Unicode

En los ejemplos de TLD anteriores, todas las cadenas de caracteres de texto se representan utilizando el conjunto de caracteres latinos. Este conjunto de caracteres está incluido en el esquema de codificación de caracteres del Código Estadounidense Estándar para el Intercambio de Información (ASCII o US-ASCII). ASCII es un esquema de codificación anterior y se basaba en el idioma inglés. Por razones históricas, se convirtió en el esquema de codificación de caracteres estándar en Internet. ASCII utiliza solo 7 bits por carácter, lo que limita el conjunto a 128 caracteres, de los cuales no todos pueden utilizarse en nombres de dominio. Los nombres de dominio están limitados a los caracteres A-Z, los números 0-9 y el guión "-".

² Existen sutilezas en los términos que no son directamente relevantes para el tema de la Aceptación Universal. Si está interesado en obtener más información sobre la terminología, un punto de partida útil es: <https://tools.ietf.org/html/rfc6365>

Tabla de ASCII - ISO 8859-1 (Latino-1) ³

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

Debido a que la mayoría de los sistemas de escritura no utilizan el conjunto de caracteres latinos, también se han adoptado codificaciones alternativas. Unicode, también conocido como **Conjunto Universal de Caracteres Codificados (UCS)**, es capaz de codificar más de 1 millón de caracteres. Cada uno de estos caracteres Unicode se denomina **punto de código**. La forma más común de Unicode se denomina **Formato de Transformación de Conjunto Universal de Caracteres Codificados de 8 bits (UTF-8)**.

Para ver todos los cuadros de códigos de caracteres Unicode, consulte: <http://unicode.org/charts>

Nombres de Dominio Internacionalizados (IDN) y Punycode

El uso de Unicode permite que los nombres de dominio contengan caracteres que no sean ASCII. Como se señaló anteriormente en este documento, los nombres de dominio que utilizan caracteres que no son ASCII se denominan Nombres de Dominio Internacionalizados (IDN).⁴ La parte internacionalizada de un nombre de dominio puede estar en cualquier nivel, no solo en el TLD, sino también en las demás etiquetas.

Como el propio DNS antes solo utilizaba ASCII,⁵ era necesario crear una codificación adicional para permitir que los puntos de código Unicode que no eran ASCII se convirtieran en cadenas de caracteres ASCII, y viceversa. El algoritmo que implementa esta codificación de Unicode a ASCII se denomina **Punycode**; las cadenas de caracteres de salida se denominan **Etiquetas-A**. Las Etiquetas-A se pueden distinguir de una etiqueta ASCII ordinaria porque siempre comienzan con los siguientes cuatro caracteres:

- **xn--**

Estos caracteres se denominan **prefijo ACE**.⁶

La transformación de Punycode es reversible: puede transformarse de Unicode a una Etiqueta-A y también desde una Etiqueta-A se puede volver a Unicode (conocida como Etiqueta-U).

³ Fuente: Universidad Estatal de California. 1997. *Tabla de ASCII - ISO 8859-1 (Latino-1) con nombres de entidades HTML*.
http://web.calstatela.edu/faculty/jchen13/Docs/CS120/Lectures/ASCIITable_with_HTML_Entity_Names.htm

⁴ Tenga en cuenta que no todos los caracteres que no son ASCII son un IDN.

⁵ Para ver el estado actual, consulte <http://tools.ietf.org/html/rfc6055#section-3>

⁶ El prefijo de codificación compatible con ASCII (ACE), del inglés ASCII Compatible Encoding, se utiliza para distinguir las etiquetas codificadas de Punycode de las etiquetas ASCII ordinarias.

El único uso definido⁷ por RFC del algoritmo Punycode es para expresar dominios internacionalizados. Sin embargo, en lugar de implementar Unicode, algunos desarrolladores eligen aplicar Punycode a otros escenarios.

Ejemplos de IDN (imaginarios)

ejemplo.みんな (Codificación Punycode = ejemplo.xn--q9jyb4c)

大坂.info (Codificación Punycode = xn--uesx7b.info)

みんな.大坂 (Codificación Punycode = xn--q9jyb4c.xn--uesx7b)

Para obtener más información, consulte la sección de Preguntas Frecuentes de los IDN:
<http://unicode.org/faq/idn.html>

Correo electrónico

Direcciones y la Internacionalización de las Direcciones de Correo Electrónico (EAI)

Las direcciones de correo electrónico contienen dos partes:

1. una parte local (el nombre de usuario, antes del carácter “@”)
2. un dominio (después del carácter “@”)

La parte de dominio puede contener cualquier TLD, incluido un nuevo TLD. Ambas partes pueden ser Etiquetas-U de Unicode.

Scripts Left to Right (LTR)

User Domain TLD
name part ↓
↓ ↓ ↓
user@example.app

Scripts Right to Left (RTL)

TLD Domain User
↓ part name ↓
↓ ↓ ↓
app.مثال@مستخدم

NOTA: A continuación, se analizará un formato adicional, las direcciones de correo electrónico estilo IDN.

Ejemplos de direcciones de correo electrónico (imaginarias) que incluyen IDN

usuario@ejemplo.みんな (utiliza TLD internacionalizados)

な (utiliza dominio de segundo nivel internacionalizado)

usuario@大坂.info (utiliza nombre de usuario internacionalizado y nuevo gTLD)

用口@ejemplo.lawyer

⁷ RFC: Solicitud de Comentarios. Consulte el Glosario de términos en la Parte 4 de este documento para obtener más información.

La internacionalización de las direcciones de correo electrónico (EAI) requiere el uso de Unicode en todas las partes de la dirección de correo electrónico. Cada uno de los ejemplos anteriores podría expresarse como EAI, y este es el formato preferido.

Generación dinámica de vínculos (Vinculación)

El software moderno, como las aplicaciones populares de procesadores de textos u hojas de cálculo, a veces permite a un usuario crear un hipervínculo con tan solo escribir una cadena de caracteres que se parece a una dirección web, dirección de correo electrónico o ruta de red. Por ejemplo, si se escribe "www.icann.org" en un mensaje de correo electrónico puede hacer que se cree automáticamente un vínculo a <http://www.icann.org> si la aplicación trata a "www." como un prefijo especial o a ".org" como un sufijo especial.

La vinculación debería funcionar de forma coherente para todas las direcciones web, direcciones de correo electrónico o rutas de red que se formen correctamente.

Parte 2: La Aceptación Universal en acción

Cinco criterios de Aceptación Universal

Como se describe en la sección, Aceptación Universal es el estado en el cual todos los nombres de dominio y direcciones de correo electrónico válidos son **aceptados, validados, almacenados, procesados y mostrados** correcta y uniformemente por todos los sistemas, dispositivos y aplicaciones habilitados para Internet. Estos cinco criterios se describen a continuación.

1. Aceptar⁸	<p>Acceptar es el proceso que ocurre cuando una dirección de correo electrónico o un nombre de dominio es recibido como una cadena de caracteres de una interfaz del usuario, un archivo o una API que utiliza una aplicación de software o un servicio en línea.</p> <p>Las aplicaciones y los servicios permiten que los nombres de dominio y las direcciones de correo electrónico se:</p> <ul style="list-style-type: none">• introduzcan en interfaces de usuario, Y/O• reciban de otras aplicaciones y servicios a través de las API
2. Validar⁹	<p>La validación puede ocurrir en muchos lugares siempre que una aplicación o servicio en línea reciba o emita una dirección de correo electrónico o un nombre de dominio como una cadena de caracteres.</p> <p>La validación está destinada a garantizar que la información especificada sea válida o al menos que concretamente no sea información no válida. En otras palabras, la validación asegura la corrección de la sintaxis de la información proporcionada.</p> <p>Para los nombres de dominio y direcciones de correo electrónico, muchos programadores han estado utilizando algunos métodos heurísticos (por ejemplo, verificar que un TLD tenga el número de caracteres "correcto" o que los caracteres sean del conjunto de caracteres ASCII). Sin embargo, estos métodos heurísticos ya no son aplicables porque Internet está cambiando:</p> <ul style="list-style-type: none">• los nombres de dominio y las direcciones de correo electrónico ahora pueden incluir caracteres Unicode (no ASCII)• la lista de TLD está creciendo• un TLD puede tener hasta 63 caracteres de longitud
3. Almacenar	<p>El proceso de Almacenamiento ocurre cuando una dirección de correo electrónico o un nombre de dominio se almacena como una cadena de caracteres en una base de datos o archivo utilizado por una aplicación de software o servicio en línea.</p> <p>Las aplicaciones y los servicios pueden requerir almacenamiento a largo plazo o transitorio de nombres de dominio y direcciones de correo electrónico.</p>

⁸ La aceptación se trata de forma distinta a la validación en este documento. En la práctica, las capacidades pueden superponerse.

⁹ La aceptación y el procesamiento se tratan de forma distinta a la validación en este documento. En la práctica, las capacidades pueden superponerse.

	<p>Independientemente de la duración de los datos, deben almacenarse en:</p> <ul style="list-style-type: none"> • formatos definidos por RFC, o bien • formatos alternativos que se puedan traducir fácilmente desde y hacia formatos definidos por RFC (esto es mucho menos deseable) <p>Aunque las RFC requieren el uso de UTF-8, se pueden encontrar otros formatos en el código heredado. Consulte la sección "prácticas recomendadas" a continuación.</p>
4. Procesar¹⁰	<p>El procesamiento tiene lugar siempre que una aplicación o un servicio utiliza una dirección de correo electrónico o un nombre de dominio para realizar una actividad (por ejemplo, realizar una búsqueda o clasificar una lista) o se transforma a un formato alternativo (como almacenar ASCII como Unicode).</p> <p>Procesamiento significa utilizar nombres de dominio y cadenas de caracteres de correo electrónico en una función. Puede realizarse una validación adicional durante el procesamiento. No existe un límite para la cantidad de formas en que los nombres de dominio y las direcciones de correo electrónico pueden procesarse (ejemplos: "Identificar a todas las personas asociadas con Nueva Zelanda porque tienen un nombre con un ccTLD .nz"; "Identificar a todos los farmacéuticos porque tienen una dirección de correo electrónico de <code>usuario@ejemplo.pharmacy</code>"; "Identificar cortafuegos que puedan filtrar las solicitudes del DNS que no se apliquen a sus políticas").</p>
5. Visualizar	<p>El proceso de Visualizar tiene lugar siempre que una dirección de correo electrónico o un nombre de dominio se presente dentro de una interfaz del usuario.</p> <p>La visualización de nombres de dominio y direcciones de correo electrónico suele ser sencilla cuando las cadenas de caracteres utilizadas son compatibles con el sistema operativo subyacente y cuando las cadenas de caracteres se almacenan en Unicode. Si no se cumplen estas condiciones, puede ser necesario realizar transformaciones específicas de aplicaciones.</p>

Escenarios de usuarios

Los ejemplos y definiciones anteriores pueden dar la impresión de que la Aceptación universal solo se trata de sistemas informáticos y servicios en línea. La realidad, sin embargo, es que también se trata de las personas que utilizan esos sistemas y servicios.

A continuación, se incluyen algunos ejemplos de actividades que requieren la Aceptación Universal:

Registrar un nuevo TLD	Una organización adopta un TLD de "marca" para ofrecer a sus clientes una experiencia de cliente diferenciada al proporcionar direcciones de correo electrónico en el formato, <code>nombredelcliente@ejemplo.brand</code> .
-------------------------------	--

¹⁰ El procesamiento se trata de forma distinta a la validación en este documento. En la práctica, las capacidades pueden superponerse.

	<p>La Aceptación Universal significa:</p> <ul style="list-style-type: none"> Las aplicaciones web aceptan estas nuevas direcciones de correo electrónico “@ejemplo.brand” con la misma validez que tendrían con TLD como .com, .net, .org.
Acceso a un gTLD	<p>Un usuario accede a un sitio web, cuyo nombre de dominio contiene un nuevo TLD, cuando escribe una dirección en un navegador o haciendo clic en un enlace en un documento.</p> <p>La Aceptación Universal significa:</p> <ul style="list-style-type: none"> Aunque el TLD sea nuevo, cualquier navegador que el usuario desee utilizar muestra la dirección web en su formato original y accede al sitio como lo prevé el usuario. El navegador no muestra texto codificado con Punycode al usuario a menos que beneficie al usuario de alguna manera.
Uso de una dirección de correo electrónico que contenga un nuevo gTLD como identidad en línea	<p>Un usuario adquiere una dirección de correo electrónico con la parte de dominio utilizando un nuevo gTLD y usa esta dirección de correo electrónico como su identidad para acceder a sus cuentas de fidelidad al banco y a la línea aérea.</p> <p>La Aceptación Universal significa:</p> <ul style="list-style-type: none"> Aunque el dominio utilizado en la dirección de correo electrónico es nuevo, el sitio del banco o de la aerolínea acepta la dirección exactamente como si fuera un TLD establecido, como .biz o .eu.
Acceso a un IDN	<p>Un usuario accede a una URL de IDN, si escribe una dirección en un navegador o si hace clic en un enlace en un documento.</p> <p>La Aceptación Universal significa:</p> <ul style="list-style-type: none"> Incluso si el nombre de dominio contiene caracteres diferentes a la configuración de idioma en la computadora del usuario, cualquier navegador que el usuario desee utilizar mostrará la dirección web como se esperaba y accederá al sitio con éxito.
Uso de una dirección de correo electrónico internacionalizada para el correo electrónico	<p>Un usuario ha adquirido múltiples direcciones de correo electrónico, algunas están internacionalizadas (por ejemplo, info@普遍接受-□□.世界).</p> <p>La Aceptación Universal significa:</p> <ul style="list-style-type: none"> el usuario puede enviar a y recibir de cualquier dirección de correo electrónico y utilizar cualquier cliente de correo electrónico.
Uso de una dirección de correo electrónico internacionalizada como identidad en línea	<p>Un usuario adquiere una dirección de correo electrónico de EAI y utiliza esta dirección de correo electrónico como su identidad para acceder a sus cuentas de fidelidad del banco y de la aerolínea.</p> <p>La Aceptación Universal significa:</p> <ul style="list-style-type: none"> El sitio del banco o de la aerolínea acepta la identidad de EAI exactamente como si fuera cualquier otra identidad de correo electrónico.
Creación dinámica	<p>Un usuario escribe una dirección web en un documento o mensaje de correo</p>

<p>de un hipervínculo en una aplicación</p>	<p>electrónico.</p> <p>La Aceptación Universal significa:</p> <ul style="list-style-type: none"> Las reglas utilizadas por la aplicación para generar automáticamente un hipervínculo son las mismas, incluso si la dirección es una EAI o contiene un nuevo TLD.
<p>Desarrollo de una aplicación</p>	<p>Un desarrollador escribe una aplicación que accede a los recursos web.</p> <p>La Aceptación Universal significa:</p> <ul style="list-style-type: none"> Las herramientas utilizadas por los desarrolladores incluyen bibliotecas que permiten la Aceptación Universal mediante el soporte de Unicode, IDN y EAI.

No conformidad con las prácticas de la Aceptación Universal

Se considera que las siguientes son **prácticas deficientes**:

<p>❑</p>	<p>Mostrar texto de Punycode al usuario sin un beneficio correspondiente para el usuario.</p> <p>Por ejemplo, para mostrar el mapeo entre una Etiqueta-U y una Etiqueta-A.</p>
<p>❑</p>	<p>Exigir que un usuario especifique texto con Punycode al registrarse para obtener una nueva dirección de correo electrónico o requerir que un usuario ingrese texto con Punycode al registrarse para un nuevo dominio alojado.</p>
<p>❑</p>	<p>Validar la sintaxis del nombre de dominio o la dirección de correo electrónico utilizando criterios desactualizados o recursos de nombres de dominio en línea no autorizados.</p>
<p>❑</p>	<p>Utilizar una lista desactualizada de TLD aunque se agreguen regularmente nuevos TLD.</p>
<p>❑</p>	<p>Exponer a los usuarios el uso interno del texto con Punycode.</p> <p>Por ejemplo, convirtiendo la EAI a una dirección de correo electrónico estilo IDN al responder a un usuario de EAI.</p>
<p>❑</p>	<p>Tratar algunos nombres de dominio como términos de búsqueda en lugar de como nombres de dominio porque la aplicación no los reconoce como tales.</p>
<p>❑</p>	<p>Establecer bloqueadores de spam para bloquear automáticamente los TLD completos.</p>

Requisitos técnicos para la preparación de la UA

Requisitos de alto nivel

Una aplicación o servicio que admite la aceptación universal (UA):

1. **Admite todos los nombres de dominio independientemente de la longitud o el conjunto de caracteres.**

Ver [RFC 5892](#).

2. **Permite múltiples conjuntos de caracteres que son válidos para nombres de dominio y direcciones de correo electrónico.**

Es decir, permite puntos de código Unicode.

3. **Puede representar correctamente todos los puntos de código en cadenas de caracteres Unicode.**

Ver [RFC 3490](#).

4. **Puede reproducir correctamente las cadenas de derecha a izquierda (RTL), como las de árabe y hebreo.**

Para obtener información acerca de códigos de escritura RTL, consulte [RFC 5893](#).

5. **Puede comunicar datos entre aplicaciones y servicios en formatos que admiten Unicode y son convertibles a/desde UTF-8.**

Para obtener información acerca de UTF-8, consulte [RFC 3629](#).

6. **Ofrece API públicas que admiten Unicode y UTF-8.**

7. **Ofrece API privadas que admiten Unicode y UTF-8.**

Las API privadas se aplican solo a las llamadas entre servicios del mismo proveedor.

8. **Almacena datos de usuario en formatos que admiten Unicode y es convertible a/desde UTF-8.**

Dichas conversiones serían visibles únicamente para el propietario del producto/servicio.

9. **Admite todas las cadenas de caracteres de nombres de dominio en la lista autoritativa de TLD de la ICANN y la lista de sufijos públicos proporcionada por la comunidad, independientemente de la longitud o el conjunto de caracteres.**

Ver <https://newgtlds.icann.org/en/program-status/delegated-strings>.

10. **Puede enviar y recibir correos electrónicos de los destinatarios, independientemente del nombre de dominio o conjunto de caracteres.**

Ver [RFC 6530](#).

11. **Trata las direcciones de EAI de la misma manera que sus equivalentes con Punycode (formato de correo electrónico de IDN).**

Consideraciones del desarrollador

Dado que muchos sistemas de software existentes contienen suposiciones codificadas sobre dominios y direcciones de correo electrónico, es posible que se requieran cambios de código para reconocer a los IDN y los nuevos TLD. Esta sección explica cómo los desarrolladores pueden incorporar cambios de código que permitirán la Aceptación Universal de todos los nuevos TLD.

Principio rector para lograr la Aceptación Universal: Ley de Postel

En la RFC 793, Jon Postel formuló el **Principio de Robustez**, ahora conocido como la **Ley de Postel**, como una guía de implementación para el entonces nuevo TCP. En **informática**, el Principio de Robustez es una directriz de diseño general para el software:

"Ser conservador en lo que se hace, ser liberal en lo que se acepta de los demás".

En otras palabras, ser conservador en lo que se envía y ser liberal en lo que se acepta. Este es también un buen enfoque cuando uno se enfrenta a las veleidades de la Aceptación Universal actualmente implementada en el ecosistema.

Prácticas recomendadas para desarrollar y actualizar el software para lograr la preparación para la Aceptación Universal

Aceptar	
□	Siempre ofrecer equivalentes de Unicode. Los usuarios deberían poder, pero no deber, ingresar texto con codificación compatible con ASCII (o "Punycode") en lugar de su equivalente Unicode. Sin embargo, se debería mostrar Unicode de manera predeterminada y que texto con Punycode se muestre al usuario solo cuando aporte un beneficio.
!	No generar direcciones de correo electrónico de estilo IDN, pero sí tener la capacidad de manejarlas si las presenta el software de otra persona.
□	Todo elemento de interfaz del usuario que requiere que el usuario escriba un nombre de dominio o dirección de correo electrónico debe ser compatible con Unicode, etiquetas de hasta 63 caracteres y cadenas de caracteres de nombres de dominio de hasta 253 caracteres. <ul style="list-style-type: none">• Ver RFC 1035.

Validar	
□	Validar únicamente en la medida mínima necesaria. Validar únicamente si es necesario para el funcionamiento de la aplicación o del servicio. Ésta es la forma más confiable de garantizar que todos los nombres de dominio válidos sean aceptados en sus sistemas.
□	Reconocer que las entradas sintácticamente correctas pueden no representar nombres de dominio o direcciones de correo electrónico actualmente en uso en Internet.
!	Si debe validar, considere lo siguiente: <ul style="list-style-type: none">• Compruebe la parte del TLD de un nombre de dominio con una tabla autoritativa. Algunos ejemplos de algunas tablas autoritativas que se puede utilizar son:<ul style="list-style-type: none">○ http://www.internic.net/domain/root.zone○ http://data.iana.org/TLD/tlds-alpha-by-domain.txt Consulte también: https://www.icann.org/en/system/files/files/sac-070-en.pdf

	<ul style="list-style-type: none"> • Consulte el nombre de dominio con el DNS. <ul style="list-style-type: none"> ○ Considere utilizar la API GETDNS (http://getdnsapi.net/) • Solicite la entrada repetida de una dirección de correo electrónico para descartar errores tipográficos. • Valide los caracteres en etiquetas solo en la medida para determinar que la U-Label (Etiqueta-U) no contenga puntos de código “NO PERMITIDOS” o puntos de código no asignados en su versión de Unicode. <ul style="list-style-type: none"> ○ Ver RFC 5892 • Limite la validación de etiquetas en sí a un pequeño número de reglas de etiquetas completas definidas en las RFC. <ul style="list-style-type: none"> ○ Ver RFC 5894 • Si una cadena de caracteres que se asemeja a un nombre de dominio contiene el carácter de punto arábigo “.” (U+06D4), o el carácter ideográfico del punto final “。” (U+3002), conviértalo al punto “.” (U+002E). • Asegúrese de que el producto o la función maneje los números de forma correcta. <ul style="list-style-type: none"> ○ Por ejemplo: los numerales de ASCII y las representaciones numéricas ideográficas asiáticas deberían tratarse todos como números.
--	--

Almacenar	
□	Las aplicaciones y los servicios deberían ser compatibles con los estándares Unicode adecuados.
□	<p>La información debería almacenarse en UTF-8 (Formato de Transformación Unicode) siempre que sea posible.</p> <p>Algunos sistemas pueden requerir compatibilidad con UTF-16 también pero, por lo general, se recomienda UTF-8. Deberían evitarse los formatos UTF-7 y UTF-32.</p>
!	<p>Considere todos los escenarios completos antes de convertir Etiquetas-A (Punycode) a Etiquetas-U y viceversa al momento de almacenar datos.</p> <p>Puede ser conveniente mantener solo Etiquetas-U en un archivo o base de datos, porque simplifica la búsqueda y la clasificación. Sin embargo, la conversión puede tener implicancias al interoperar con aplicaciones y servicios más antiguos no habilitados con Unicode. Considere almacenar e indexar los datos en ambos formatos.</p>
□	<p>Marque claramente las direcciones de correo electrónico y los nombres de dominio durante el almacenamiento para facilitar el acceso a ellos.</p> <p>Instancias en las que se han archivado direcciones de correo electrónico y nombres de dominio bajo el campo “autor” de un documento o “información de contacto” en un archivo de registro han llevado a la pérdida de la dirección original.</p>
□	<p>Si <i>no</i> almacena en Unicode, debe ser capaz de hacer coincidir cadenas de caracteres en múltiples formatos.</p> <p>Por ejemplo, una búsqueda de ejemplo.みんな también debería encontrar ejemplo.xn--q9jyb4c.</p>

Procesar	
□	Asegúrese de que todas las respuestas del servidor tengan especificada la opción Unicode en el tipo de contenido.
□	<p>Especifique Unicode en el encabezado de http del servidor web y directamente en un archivo web.</p> <ul style="list-style-type: none"> • Cada archivo web debería incluir el conjunto de caracteres de UTF-8. • Es importante asegurarse de que la codificación se especifique en cada respuesta.
!	<p>Considere todos los escenarios completos antes de convertir Etiquetas-A (Punycode) a Etiquetas-U y viceversa durante el procesamiento.</p> <p>Puede ser conveniente mantener solo Etiquetas-U en un archivo o base de datos, porque simplifica la búsqueda y la clasificación. Sin embargo, la conversión puede tener implicancias al interoperar con aplicaciones y servicios más antiguos no habilitados con Unicode. Considere almacenar los datos en ambos formatos.</p>
□	Asegúrese de que el producto o la función maneje el orden de clasificación, las búsquedas y la clasificación de acuerdo con las especificaciones de idioma/configuración regional, y que aborde la búsqueda y clasificación en varios idiomas.
□	<p>No utilice la codificación URL para nombres de dominio:</p> <ul style="list-style-type: none"> • ejemplo.みんな es correcto • ejemplo.%E3%81%BF%E3%82%93%E3%81%AA no es correcto
□	<p>Debido a que el estándar Unicode se expande de manera continua, los puntos de código no definidos al momento en que se creó el servicio o la aplicación deberían ser controlados a fin de garantizar que no “interrumpan” la experiencia del usuario.</p> <p>Tipos de letras faltantes en el sistema operativo subyacente pueden ocasionar caracteres que no pueden mostrarse (con frecuencia, el carácter “□” se utiliza para representar los mismos), pero esta situación no debería generar un fallo irrecuperable.</p>
□	Utilice API habilitadas con Unicode compatibles.
□	<p>Utilice la última versión del Protocolo de Nombres de Dominio Internacionalizados en Aplicaciones (IDNA) y los documentos de tablas para IDN.</p> <ul style="list-style-type: none"> • RFC 5891 • RFC 5892
□	Realice el procesamiento en formato UTF-8 siempre que sea posible.
□	<p>Actualice las aplicaciones y los servidores/servicios en conjunto.</p> <p>Si el servidor es Unicode y el cliente no es Unicode o viceversa, los datos deberán ser convertidos a cada página de código cada vez que viajen entre el servidor y el cliente.</p>
□	<p>Realice revisiones de código para evitar ataques de desbordamiento de búfer.</p> <p>Al realizar la transformación de caracteres, las cadenas de caracteres de texto pueden crecer o</p>

	reducirse considerablemente.
--	------------------------------

Visualizar	
□	<p>Visualice todos los puntos de código Unicode compatibles con el sistema operativo subyacente.</p> <p>Si una aplicación mantiene sus propios juegos de fuentes, se debería ofrecer compatibilidad integral con Unicode al conjunto de fuentes disponibles del sistema operativo.</p>
□	<p>Al desarrollar una aplicación o un servicio, considere los lenguajes compatibles y asegúrese de que los sistemas operativos y las aplicaciones incluyan dichos lenguajes.</p>
□	<p>Convierta los datos que no tienen formato Unicode a Unicode antes de visualizarlos.</p> <p>Por ejemplo, el usuario final debería ver “ejemplo.みんな” en lugar de “ejemplo.xn--q9jyb4c”. (Esta conversión es un ejemplo de procesamiento preparado para UA).</p>
□	<p>Visualice Unicode de manera predeterminada.</p> <p>Visualice texto con Punycode para el usuario solo cuando aporte un beneficio.</p>
!	<p>Tenga en cuenta que las direcciones con combinación de códigos de escritura se volverán más comunes.</p> <ul style="list-style-type: none"> Algunos caracteres Unicode pueden parecer iguales ante el ojo humano, pero diferentes para las computadoras. No suponga que las cadenas de caracteres con combinación de códigos de escritura tienen propósitos maliciosos, tales como phishing. Si la interfaz del usuario hace que las cadenas de caracteres llamen la atención del usuario, asegúrese que lo haga de manera que no resulte perjudicial a los usuarios de códigos de escritura no latinos. <p>Obtenga más información sobre las Consideraciones de seguridad de Unicode en: http://unicode.org/reports/tr36</p>
□	<p>Use el Procesamiento de Compatibilidad de IDNA para Unicode a fin de cumplir con las expectativas de los usuarios.</p> <p>Para obtener más información, consulte: http://unicode.org/reports/tr46</p>
□	<p>Tenga presente los caracteres no asignados y no permitidos para nombres de dominio.</p> <ul style="list-style-type: none"> Ver RFC 5892

Unicode	
□	Utilice API habilitadas con Unicode compatibles.
□	No construya sus propias API para:

	<ul style="list-style-type: none"> • conversiones de formato de cadenas de caracteres • determinar qué código de escritura comprende una cadena de caracteres • determinar si una cadena de caracteres contiene una combinación de códigos de escritura • normalización/descomposición Unicode
❑	<p>No utilice UTF-7 ni UTF-32.</p> <ul style="list-style-type: none"> • UTF-7 generalmente no se usa como una representación nativa dentro de las aplicaciones porque es muy difícil de procesar. A pesar de su ventaja de tamaño con respecto a la combinación de UTF-8 con quoted-printable o base64, Internet Mail Consortium recomienda que no se utilice. • La principal desventaja de UTF-32 es que es ineficiente en cuanto al espacio, y utiliza cuatro bytes por punto de código. Los caracteres no BMP son tan inusuales en la mayoría de los textos[cita requerida], también se pueden considerar inexistentes para cuestiones de tamaño, lo que hace que UTF-32 tenga el doble del tamaño de UTF-16 y hasta cuatro veces el tamaño de UTF-8.
❑	<p>Utilice Unicode en las cookies para que las aplicaciones puedan leerlas correctamente.</p>
❑	<p>Utilice los documentos de tablas y el Protocolo 2008 de IDNA:</p> <ul style="list-style-type: none"> • RFC 5891 • RFC 5892
❑	<p>No utilice IDNA 2003; en casi todos los casos han sido reemplazados por IDNA 2008.</p>
❑	<p>No asuma automáticamente que las API externas pueden consumir datos que hayan sido convertidos a NFKC¹¹.</p>
!	<p>Mantenga las tablas de IDNA y Unicode que sean compatibles en lo que respecta a las versiones.</p> <p>Por ejemplo, a menos que la aplicación realmente ejecute las reglas de clasificación en el documento de Tablas (RFC 5892), sus tablas de IDNA deben derivarse de la versión de Unicode que se admita de manera más general en el sistema. Al igual que con el registro, las tablas no necesitan reflejar la última versión de Unicode, pero deben ser compatibles.</p>
!	<p>Valide los caracteres en etiquetas solo en la medida para determinar que la U-Label (Etiqueta-U) no contenga puntos de código "NO PERMITIDOS"¹² o puntos de código no asignados en su versión de Unicode.</p>
❑	<p>Limite la validación de etiquetas en sí a un pequeño número de reglas de etiquetas completas:</p> <ul style="list-style-type: none"> • sin marcas de unión iniciales • las condiciones bidireccionales se cumplen si aparecen caracteres de derecha a izquierda • se prueban todas las reglas contextuales que estén asociadas con los caracteres de unión (y

¹¹ **NFKC** (*Composición de Compatibilidad de Formulario de Normalización*): Los caracteres se descomponen por compatibilidad, luego se recomponen por equivalencia canónica. Ver: <http://unicode.org/reports/tr15>

¹² **NO PERMITIDOS**: puntos de código que no deberían incluirse en los IDN. Ver: <https://tools.ietf.org/html/rfc5892>

	los caracteres CONTEXTJ ¹³ en general)
!	<p>No utilice UTF-16 excepto cuando se requiera explícitamente (como en ciertas API de Windows).</p> <p>Cuando utilice UTF-16, tenga en cuenta que 16 bits solo pueden contener el rango de caracteres de 0x0 a 0xFFFF, y se utiliza una complejidad adicional para almacenar valores por encima de este rango (0x10000 a 0x10FFFF). Esto se hace utilizando pares de unidades de código conocidas como sustitutos. Si el manejo de pares sustitutos no se prueba exhaustivamente, puede generar errores complicados y posibles brechas de seguridad.</p>

Vinculación

□	Si una cadena de caracteres que se asemeja a un nombre de dominio contiene el carácter de punto arábigo “.” (U+06D4), o el carácter ideográfico del punto final “。” (U+3002), conviértalo al punto “.” (U+002E).
---	--

General

□	<p>Utilice recursos autorizados para validar nombres de dominio.</p> <p>No haga suposiciones heurísticas, como "todos los TLD son de 2, 3, 4 o 6 caracteres de longitud".</p>
□	<p>Asegúrese de que el producto o la función maneje los números de forma correcta.</p> <p>Por ejemplo, los numerales de ASCII y las representaciones numéricas ideográficas asiáticas deberían tratarse todos como números.</p>
!	<p>Busque direcciones de correo en lugares inesperados:</p> <ul style="list-style-type: none">• artista/autor/fotógrafo/metadatos de derechos de autor• Metadatos de fuente• Registros de contacto del DNS• Información de versión binaria• Información de apoyo• Información de contacto de OEM• Registración, comentarios y otros formularios
!	<p>Busque posibles rutas de IRI¹⁴ en lugares inesperados:</p> <ul style="list-style-type: none">• nombres de máquinas de una sola etiqueta independientemente de la página de códigos del sistema que se haya cargado• nombres de máquinas completos independientemente de la página de códigos del sistema

¹³ CONTEXTJ: Regla contextual para controles de unión Ver: <https://tools.ietf.org/html/rfc5892>

¹⁴ IRI: Identificadores de Recursos Internacionalizados Ver: <https://www.ietf.org/rfc/rfc3987.txt>

	que se haya cargado
□	Utilice GB18030 (China) para compatibilidad ¹⁵ de idioma chino además de UTF-8.
!	<p>Restrinja los puntos de código permitidos al generar nuevos nombres de dominio y direcciones de correo electrónico:</p> <p>Todos los productos que utilizan direcciones de correo electrónico deben aceptar direcciones de correo electrónico internacionalizadas y permitir caracteres > U+007f. Es decir, no se rechaza ningún carácter > U+007f. Sin embargo, una aplicación o servicio no necesita permitir todos estos caracteres cuando un usuario crea un nuevo IDN o dirección de correo electrónico. Utilice únicamente esta lista de caracteres permitidos para los IDN: http://unicode.org/reports/tr36/idn-chars.txt</p> <p>Evitar que se creen ciertos IDN o direcciones de correo electrónico en primer lugar puede mitigar algunas posibles preocupaciones de seguridad y accesibilidad. (NOTA: La Ley de Postel aún requeriría que el software acepte dichas cadenas de caracteres si se presentan).</p>
!	<p>Tenga en cuenta que la Aceptación Universal no siempre se puede medir a través de casos de prueba automatizados solamente.</p> <p>Por ejemplo, no siempre es posible probar cómo una aplicación o protocolo gestiona los recursos de la red y, en ocasiones, es mejor verificar el cumplimiento a través de la revisión de la especificación funcional y la revisión del diseño.</p>
!	<p>No suponga automáticamente que porque un componente no invoca directamente a las API de resolución de nombres, ni utiliza directamente direcciones de correo electrónico, no significa que no lo afecten.</p> <p>Comprenda cómo el componente obtiene los nombres de red; no siempre es a través de la interacción del usuario. Los siguientes son algunos ejemplos sobre cómo el componente puede obtener un nombre de red:</p> <ul style="list-style-type: none"> • política de grupo • consulta LDAP • archivos de configuración • registro de Windows • transferido a/desde otro componente/función
□	<p>Realice revisiones de código para evitar ataques de desbordamiento de búfer.</p> <ul style="list-style-type: none"> • En Unicode, las cadenas de caracteres pueden expandirse en cuanto a las mayúsculas y minúsculas: Fluß → FLUSS → fluss • Al realizar la conversión de caracteres, el texto puede crecer o reducirse considerablemente.

¹⁵ GB 18030-2000 es un estándar del gobierno chino que especifica una página de códigos extendida para su uso en el mercado chino. Ver: <http://icu-project.org/docs/papers/unicode-gb18030-faq.html>

Fuentes fidedignas para nombres de dominio

Zona Raíz del DNS

Existen algunas opciones para la lista autoritativa de TLD. La primera opción es la zona raíz del DNS en sí. Está firmado por DNSSEC, por lo que la lista está autenticada correctamente. Puede obtener la zona raíz desde cualquiera de los siguientes enlaces:

- <http://www.internic.net/domain/root.zone>
- <http://www.dns.icann.org/services/authoritative-dns/index.html>
- <http://data.iana.org/TLD/tlds-alpha-by-domain.txt>

Lista de sufijos públicos

La Lista de sufijos públicos (PSL), que administran voluntarios de la Fundación Mozilla, proporciona una lista precisa de sufijos de nombres de dominio. Esta lista es un conjunto de nombres del DNS o comodines concatenados con puntos y codificados con UTF-8. Si necesita utilizar la PSL como una fuente autoritativa de nombres de dominio, su software debe recibir de forma periódica actualizaciones de la PSL. No cree copias estáticas de la PSL en su software sin un mecanismo de actualización. Puede utilizar el siguiente enlace para hacer que su aplicación descargue una lista actualizada de forma periódica. La lista se actualiza una vez al día desde Github:

- https://publicsuffix.org/list/public_suffix_list.dat

Otros desafíos

General

Codificación variable de los IDN	<p>En algunas aplicaciones, los IDN están codificados:</p> <ul style="list-style-type: none"> • en Punycode, según los IDNA, si el nombre se identifica como un nombre de Internet, PERO • en UTF-8, si el nombre se identifica como un nombre en la red de área local ("intranet")
Mecanismo para detectar y convertir conjuntos de caracteres	<p>Algunas aplicaciones de correo electrónico antiguas estaban codificadas en una página de códigos local y no tenían un mecanismo establecido para detectar y convertir el conjunto de caracteres según fuera necesario. Esto ocurría especialmente en el encabezado del correo electrónico (PARA, CC, CCO, Asunto).</p>
Fallas al manejar protocolos que no sean del DNS	<p>Algunas aplicaciones que generan IDNA (por ejemplo, IE7+) fallan con protocolos que no son del DNS. Esto podría afectar el acceso a recursos que utilizan protocolos que no son del DNS.</p>
Mecanismo para gestionar múltiples direcciones de correo electrónico en una sola identidad de usuario	<p>Cuando un usuario seudonimiza varias direcciones de correo electrónico, puede resultar complicado gestionar estas direcciones como una única identidad de usuario.</p> <p>Los programas de correo electrónico pueden dirigir el tráfico que va hacia dichos seudónimos al mismo buzón, pero la aplicación aún puede percibir que estos correos electrónicos pertenecen a diferentes identidades.</p>

Consejo práctico para desarrolladores de software



Cuando permita que un usuario genere un nombre de dominio o una dirección de correo electrónico, considere evitar el uso de caracteres visualmente confusos para evitar ataques homógrafos. Utilice únicamente esta lista de caracteres permitidos para los IDN: <http://unicode.org/reports/tr36/idn-chars.txt>

Correo electrónico estilo IDN y por qué no es lo mismo que EAI

La EAI se define como el uso exclusivo de Unicode; las Etiquetas-A (Punycode) no están permitidas. Sin embargo, los desarrolladores a veces han adaptado el software y los servicios de correo electrónico para manejar las direcciones de correo electrónico estilo IDN en lugar de realizar una conversión completa a Unicode.

Debido a que los IDN pueden codificarse con Punycode, algunos programas existentes permiten representar la porción del IDN de una dirección de correo electrónico en ASCII o Unicode. Por ejemplo, algunos programas tratarán estas dos direcciones de "correo electrónico estilo IDN" de manera equivalente para todos los fines (envío, recepción y búsqueda):

No todos los programas tratarán a estos dos correos electrónicos estilo IDN como funcionalmente equivalentes

`usuario@ejemplo.みんな = usuario@ejemplo.xn--q9jyb4c`

Sin embargo, algunos tipos de software no tratarán de forma robusta a estas direcciones como equivalentes, incluso si ambas son válidas, porque no existe ningún requisito para que el software procese una Etiqueta-A (por ejemplo, "xn--q9jyb4c") en su equivalente Etiqueta-U (por ejemplo "みんな") antes de comparar. Esto puede generar una experiencia de usuario impredecible. La experiencia del usuario puede volverse especialmente confusa si un software convierte Etiquetas-U en Etiquetas-A por "compatibilidad"; a medida que se respondan o reenvíen los mensajes, pueden aumentar las direcciones que sean visiblemente diferentes a un usuario o que no puedan realzar la búsqueda y clasificación como se espera.

En el siguiente ejemplo, algunos programas pueden intentar convertir incluso la parte local de la dirección de correo electrónico utilizando Punycode, y crear algo que se parezca a una ETIQUETA-A en la parte local de la dirección. Esto no está permitido conforme a las RFC existentes, y es muy probable que genere fallas para recibir correos electrónicos de ciertos sistemas y que genere dificultades de búsqueda y clasificación

Nunca convierta la parte local de una dirección de correo electrónico con Punycode

- ❑ 用口@ejemplo.みんな
- ❑ xn--youq53b@ejemplo.xn--q9jyb4c

como se explicó anteriormente.

El software y los servicios sólidos listos para la Aceptación Universal pueden ser capaces de manejar y tratar todos estos formatos de forma idéntica, incluso aquellos que no cumplan con las RFC. Sin embargo, el software listo para la Aceptación Universal no debe generar direcciones de correo electrónico de EAI verdaderas únicamente.

La vinculación y sus desafíos

El software moderno a veces permite a un usuario crear automáticamente un hipervínculo con tan solo escribir una cadena de caracteres que se parece a una dirección web, nombre de correo electrónico o ruta de red. Por ejemplo, si se escribe "www.icann.org" en un mensaje de correo electrónico puede hacer que se cree automáticamente un vínculo a <http://www.icann.org> si la aplicación trata a "www." como un prefijo especial o a ".org" como un sufijo especial.

La vinculación debería funcionar de forma coherente para todas las direcciones web, nombres de correo electrónico o rutas de red que se formen correctamente.

La vinculación es la acción donde una aplicación acepta una cadena de caracteres y determina de forma dinámica si debe crear un hipervínculo a una ubicación de Internet (URL) o una dirección de correo electrónico (<mailto:>

La vinculación utiliza algoritmos y reglas creadas por los desarrolladores de software para determinar si una cadena se debe considerar un enlace, o no. En relación a esto es la forma en que la gente puede identificar una cadena de caracteres como un nombre de dominio. Si bien los navegadores, clientes de correo electrónico y procesadores de texto son lugares obvios, existen muchas más aplicaciones que toman estas decisiones.

Recomendaciones de buenas prácticas

1. Intente vincular en base a prefijos de protocolos explícitos (por ejemplo "http://", "ftp://", "mailto:") pero complete la acción únicamente si el resto de la cadena de caracteres está formada correctamente.

Cadena de caracteres de ejemplo	Comportamiento esperado / Resultado
ejemplo.com	Sin hipervínculo porque el protocolo está ausente y no se infiere.
http://ejemplo.com	Crea hipervínculo porque el protocolo es explícito.
http:ejemplo.com	Sin hipervínculo debido a sintaxis incorrecta (falta //)
http://ejemplo.a	Sin hipervínculo porque las Políticas de la ICANN requieren que el TLD tenga al menos dos caracteres. Nota: esta sintaxis podría ser compatible dentro de una red interna.
http://ejemplo..ab	Sin hipervínculo debido a sintaxis incorrecta (puntos consecutivos)
http:// 普遍接受-□□.世界	Crea hipervínculo porque el protocolo es explícito.

2. Intente generar hipervínculos en base a prefijos de protocolos implícitos (por ejemplo, "www" infiere "http://www")

Cadena de caracteres de ejemplo	Comportamiento esperado / Resultado
www.example.com	Crea hipervínculo porque el protocolo está implícito ¹⁶ .
label@example.com	Cree mailto: etiqueta@ejemplo.com porque el protocolo está implícito.

3. Asigne el punto final ideográfico “。” (U+3002) y el carácter de punto final arábigo“.” (U+06D4), al punto final “.” (U+002E) (por ejemplo http://田中。com => http://田中.com) si la cadena de caracteres está bien formada.
4. Si los TLD se utilizan como un 'sufijo especial' para determinar la capacidad de generar el vínculo, entonces deben incluirse todos los TLD. Una lista de TLD válidos debería actualizarse de forma dinámica con frecuencia.

¹⁶ Nota: podría ser que el sitio web real requiera que los usuarios finales escriban https:// en lugar de http://. Si este es el caso, entonces el hipervínculo puede no resolverse o puede devolver una página de error.

Parte 3: Temas avanzados

Códigos de escritura complejos

Los detalles de los códigos de escritura complejos pueden no ser de interés para aquellos que no sean desarrolladores y que creen sus propias bibliotecas de análisis sintáctico de cadena de caracteres. Sin embargo, se incluye un resumen aquí para garantizar que todos los lectores tengan suficiente conocimiento para reconocer errores de código relacionados con estos códigos de escritura cuando se encuentren en las experiencias del usuario.

Idiomas de derecha a izquierda y conformidad con Unicode

La mayoría de los códigos de escritura muestran caracteres de izquierda a derecha cuando el texto se presenta en líneas horizontales. Sin embargo, también existen varios códigos de escritura, como el árabe o el hebreo, donde el orden del texto horizontal en la pantalla es de derecha a izquierda. El texto también puede ser bidireccional (de izquierda a derecha, de derecha a izquierda) cuando un código de escritura de derecha a izquierda utiliza dígitos que se escriben de izquierda a derecha o cuando utiliza palabras incrustadas del inglés u otros códigos de escritura.

Se pueden generar desafíos y ambigüedades cuando la dirección horizontal del texto no es uniforme. Para resolver este problema, existe un algoritmo para determinar la direccionalidad del texto Unicode bidireccional.

Existe un conjunto de reglas que la aplicación debe implementar para producir el orden correcto en el momento de la visualización, que se describen mediante el **Algoritmo Bidireccional de Unicode**. Generalmente nos referimos a esto como el "**algoritmo Bidi**".

El algoritmo Bidi

El algoritmo Bidi describe cómo el software debe procesar el texto que contiene secuencias de caracteres de izquierda a derecha (LTR) y de derecha a izquierda (RTL). La **dirección base**¹⁷ asignada a la frase determinará el orden en que se visualizará el texto.

Para saber si una secuencia es de izquierda a derecha o de derecha a izquierda, cada carácter en Unicode tiene una propiedad direccional asociada. La mayoría de las letras son **fuertemente tipadas (caracteres fuertes)** como LTR (de izquierda a derecha). Las letras de los códigos de escritura de derecha a izquierda están fuertemente tipadas como RTL (de derecha a izquierda). Una secuencia de caracteres RTL fuertemente tipados se visualizará de derecha a izquierda. Esto es independiente de la dirección base circundante. Por ejemplo:

(LTR) ejemplo - مثال (RTL).

El texto con diferente direccionalidad se puede combinar en línea. En dichos casos, el algoritmo Bidi produce un **sentido direccional** separado de cada secuencia de caracteres contiguos con la misma direccionalidad.

Los espacios y la puntuación no son fuertemente tipados como LTR o RTL en Unicode porque se pueden usar en cualquier tipo de código de escritura. Por lo tanto, se clasifican como **caracteres neutrales o débiles**. Los caracteres débiles son aquellos con una direccionalidad vaga. Los ejemplos de este tipo de caracteres incluyen:

¹⁷ En HTML, la dirección base se hereda de la dirección predeterminada del documento, que es de izquierda a derecha, o se establece explícitamente con el elemento primario más cercano que utiliza el atributo `dir`.

- dígitos europeos
- dígitos arábigos-índicos
- símbolos aritméticos y símbolos de divisas
- símbolos de puntuación que son comunes a muchos códigos de escritura, como los dos puntos, la coma, el punto y el espacio de no separación

La direccionalidad de los **caracteres neutrales** es indeterminada sin contexto. Algunos ejemplos incluyen:

- pestañas
- separadores de párrafos
- la mayoría de los otros caracteres de espacios en blanco

Cuando un carácter neutral se encuentra entre dos caracteres fuertemente tipados que tienen el mismo tipo direccional, también adoptará esa direccionalidad. Por ejemplo, un carácter neutral entre dos caracteres RTL se tratará como un carácter RTL en sí mismo, y tendrá el efecto de extender el sentido direccional:

- مثال.نطاق

Incluso si hay varios caracteres neutrales entre los dos caracteres fuertemente tipados, todos serán tratados de la misma manera.

Cuando un espacio o puntuación se encuentra entre dos caracteres fuertemente tipados que tienen diferente direccionalidad, el carácter neutro (o caracteres) se tratará como si tuviera la misma direccionalidad que la dirección base predominante. Por ejemplo:

- ejemplo. مثال

A menos que exista un reemplazo direccional, los **números** siempre se codifican (y se introducen) según el sistema big-endian¹⁸, y los numerales se representan de izquierda a derecha. La direccionalidad débil solo se aplica a la ubicación del número en su totalidad.

Para ver el algoritmo Bidi con mayor detalle, consulte: <http://unicode.org/reports/tr9/tr9-11.html>

La regla Bidi para nombres de dominio

Un **nombre de dominio Bidi** es un nombre de dominio que contiene al menos una etiqueta RTL. Existe una regla que determina las condiciones que deben cumplirse para las etiquetas en los nombres de dominio Bidi. Esta regla se puede encontrar en la Sección 2 de RFC 5893: <https://tools.ietf.org/html/rfc5893>

Uniones

Algunos lenguajes utilizan códigos de escritura alfabéticos en los que los fonemas individuales se escriben utilizando dos caracteres denominados **dígrafo**. En otras palabras, un dígrafo es un grupo de dos letras sucesivas que representan un solo sonido (o **fonema**).

¹⁸ "Big-endian y little-endian son términos que describen el orden en que se almacena una secuencia de bytes en la memoria de la computadora. Big-endian es un orden en el que el 'big end' (el valor más significativo de la secuencia) se almacena primero (en la dirección de almacenamiento más baja). Little-endian es un orden en el que el 'little end' (valor menos significativo en la secuencia) se almacena primero". Fuente: <http://searchnetworking.techtarget.com/definition/big-endian-and-little-endian>

Ejemplos de dígrafos en inglés

ch (como en *church*)
ph (*phone*)

th (*then*)
th (*think*)

sh (*shoe*)

Algunos dígrafos están completamente unidos como **ligaduras**. En escritura y tipografía, una ligadura se produce donde dos o más grafemas o letras se unen como un único glifo. Un ejemplo es el carácter et (&), que evolucionó a partir de las letras latinas adjuntas *e* y *t* ("et" que significa "y").

Si las ligaduras y los dígrafos tienen la misma interpretación en todos los idiomas que utilizan un código de escritura determinado, la normalización Unicode generalmente resuelve las diferencias y las hace coincidir. Cuando tienen diferentes interpretaciones, para lograr la coincidencia se deben utilizar métodos alternativos, probablemente elegidos a nivel de registro, o los usuarios deben ser instruidos para comprender que la coincidencia no tendrá lugar. Un ejemplo de interpretación diferente se puede encontrar en la Sección 4.3 de RFC 5894: <https://tools.ietf.org/html/rfc5894>

El Consorcio Unicode enumera dos estrategias principales para determinar el comportamiento de unión de un carácter en particular después de aplicar el algoritmo Bidi:

- "Cuando se configura, una implementación puede hacer referencia al almacenamiento de respaldo original para ver si había caracteres ZWNJ o ZWJ¹⁹ adyacentes.
- De forma alternativa, la implementación puede reemplazar ZWJ y ZWNJ por una propiedad de carácter fuera de banda asociada con esos caracteres adyacentes, de modo que la información no interfiera con el algoritmo Bidi y la información se conserve a través de la reorganización de esos caracteres. Una vez que se ha aplicado el algoritmo Bidi, esa información fuera de banda se puede utilizar para dar la forma adecuada".²⁰

Ante la falta de cuidado por parte de los registros sobre cómo se manejan las cadenas de caracteres que podrían tener diferentes interpretaciones conforme a IDNA2003 y la especificación actual, es posible que las diferencias se puedan utilizar como un componente de los ataques de coincidencia de nombres o confusión de nombres. Por lo tanto, dicho cuidado es apropiado.

Para obtener más información sobre las uniones, consulte la Sección 4.3 de RFC 5894: <https://tools.ietf.org/html/rfc5894>

Homóglifos y caracteres confusamente similares

Los **homóglifos** son caracteres que, debido a las similitudes en tamaño y forma, pueden parecer idénticos a primera vista.

Ejemplos de homóglifos

Carácter cirílico а	=	Número Unicode 0430
Carácter latino a	=	Número Unicode 0061

¹⁹ Para obtener más información sobre ZWNJ/ZWJ, consulte: <http://www.unicode.org/L2/L2005/05307-zwj-zwj.pdf>

²⁰ Fuente: Mark Davis, Aharon Lanin, Andrew Glass. 2015. *Unicode*. <http://unicode.org/reports/tr9>

Para evitar que se registren nombres de dominio que se vean de manera confusa, los registros pueden utilizar el procedimiento de "agrupación de homóglifos".²¹

La **agrupación de homóglifos** es cuando se registra un IDN y el sistema de registro agrupa automáticamente todos los homóglifos de ese nombre (si hay alguno). Esto significa que varios nombres de dominio están agrupados a la vez, y ninguno de los otros nombres de dominio en esa agrupación puede registrarse.

La agrupación de homóglifos es una práctica recomendada para los registros para evitar posibles prácticas de phishing que pretenden engañar al usuario con caracteres visualmente confusos.

Para obtener más información sobre los mecanismos de seguridad de Unicode para la detección de confusiones, consulte:

- http://www.unicode.org/reports/tr39/#Confusable_Detection

Para ver una lista de homóglifos, consulte:

- <http://homoglyphs.net>

Para obtener más información sobre los caracteres confusamente similares y las prácticas recomendadas, consulte:

- Tutorial y descripción general del uso indebido de Unicode del M3AAWG
<https://www.m3aawg.org/sites/default/files/m3aawg-unicode-tutorial-2016-02.pdf>
- Prácticas recomendadas del M3AAWG para la prevención del uso indebido de Unicode
<https://www.m3aawg.org/sites/default/files/m3aawg-unicode-best-practices-2016-02.pdf>

Normalización y conversión de mayúsculas y minúsculas

Normalización

La normalización de Unicode ayuda a determinar si dos cadenas de caracteres de Unicode son equivalentes entre sí. Algunos caracteres pueden ser representados en Unicode por varias secuencias de códigos. Esto se denomina **equivalencia Unicode**. Unicode proporciona dos tipos de equivalencias:

- Canónica (NFD)
- Compatibilidad (NFK)

Las secuencias que representan el mismo carácter se denominan **canónicamente equivalentes**. Estas secuencias tienen el mismo aspecto y significado cuando se imprimen o visualizan. Por ejemplo:

Ejemplos de caracteres canónicamente equivalentes

U+006E (“n” latina minúscula) seguido de U+0303 (la combinación de la tilde “̃”)	=	ñ
U+00F1 (letra “ñ” minúscula del alfabeto español)	=	ñ

²¹ <https://www.icann.org/resources/pages/idn-guidelines-2011-09-02-en>

Los **equivalentes de compatibilidad** son secuencias que pueden tener diferentes apariencias, pero en algunos contextos tienen el mismo significado. Es un tipo de equivalencia más débil entre caracteres o secuencias de caracteres.

Ejemplos de caracteres canónicamente equivalentes

U+FB00 (la ligadura tipográfica "ff")	=	ff
U+0066 U+0066 (dos letras "f" latinas)	=	ff

En el ejemplo anterior, el punto de código U+FB00 se define como compatible, pero no canónicamente equivalente a la secuencia U+0066 U+0066. Las secuencias que son canónicamente equivalentes también son compatibles, pero lo contrario no es necesariamente cierto.

Para evitar problemas de interoperabilidad derivados del uso de secuencias de caracteres canónicamente equivalentes, pero diferentes, el W3C recomienda utilizar el Formulario de Normalización C²² para todo el contenido.

Para ver una lista de todos los caracteres que pueden cambiar en cualquiera de los Formularios de Normalización, consulte: <http://www.unicode.org/charts/normalization>

Algunos otros puntos a tener en cuenta:

- Solo las cadenas NO transformadas por NFKC²³ son válidas.
- Cuando dos aplicaciones comparten datos Unicode, pero los normalizan de forma diferente, pueden producirse errores y pérdida de datos.
- Los formularios de normalización deben permanecer estables con el transcurso del tiempo. En otras palabras, una cadena de caracteres debe permanecer normalizada en todas las versiones futuras de Unicode (compatibilidad con versiones anteriores).

Consejo práctico para desarrolladores de software



No realice la normalización convirtiendo a mayúsculas o ignorando los caracteres que no sean espacios, ya que esto también puede dificultar la clasificación, la copia de datos, la importación y exportación de datos, la recuperación de datos por parte de las aplicaciones cliente y la pérdida o corrupción de datos.

Para obtener más información sobre formularios de normalización, consulte: <http://www.unicode.org/reports/tr15>

Conversión de mayúsculas y minúsculas

La **conversión de mayúsculas y minúsculas** es el proceso de hacer que dos textos, que difieren en cuanto a mayúsculas y minúsculas pero son "iguales", sean idénticos. La correspondencia de [a-z] a [A-Z] funciona para la mayoría de los documentos de texto simples únicamente con ASCII. Sin embargo, comienza a descomponerse con idiomas que utilizan caracteres adicionales.

²² NFC: descomposición canónica, seguida de composición canónica.

²³ NFKC: descomposición canónica, seguida de composición canónica.

Unicode define la correspondencia predeterminada de la conversión de mayúsculas y minúsculas para cada punto de código Unicode. Existen **correspondencias de conversión de mayúsculas y minúsculas completas y comunes**:

- **Las correspondencias de conversión de mayúsculas y minúsculas comunes** son aquellas que tienen una correspondencia sencilla y directa a un solo punto de código de coincidencia (principalmente minúsculas).
- **Las correspondencias de conversión completas** son aquellas que normalmente requerirían más de un carácter Unicode.

Una consideración importante, según el W3C,²⁴ es si los valores están restringidos al subconjunto ASCII de Unicode o si el vocabulario permite el uso de caracteres (como acentos en letras latinas o una amplia gama de Unicode incluidos códigos de escritura no latinos) que potencialmente tiene requisitos más complejos de conversión de mayúsculas y minúsculas.²⁵

Consejo práctico para desarrolladores de software

□ Considere la normalización de Unicode además de la conversión de mayúsculas y minúsculas.

Para obtener más información sobre la normalización de Unicode, consulte:

- <http://www.w3.org/TR/charmod-norm>
- <http://unicode.org/reports/tr15>

Para obtener recomendaciones sobre la conversión de mayúsculas y minúsculas, consulte:

- https://www.w3.org/International/wiki/Case_folding

²⁴ W3C: El Consorcio Mundial de Internet (W3C) es una comunidad internacional en la cual las **organizaciones Miembro**, el **personal** de tiempo completo y el público trabajan en conjunto para desarrollar **estándares para la Web**. Ver: <https://www.w3.org>

²⁵ Fuente: A Phillips. 2015. *Modelo de caracteres para la World Wide Web: Coincidencia y búsqueda de cadena de caracteres*. <https://www.w3.org/TR/charmod-norm>

Parte 4: Glosario y otros recursos

Glosario

Etiqueta-A	Representación codificada compatible con el Código ASCII (ACE) de un nombre de dominio internacionalizado; por ejemplo, como se lo transmite internamente dentro del protocolo del DNS. Las Etiquetas-A siempre comienzan con el prefijo "xn-". Comparar con etiquetas U.
Prefijo ACE	Prefijo de codificación compatible con ASCII.
Caracteres ASCII	Código Estadounidense Estándar para el Intercambio de Información. Estos son caracteres del alfabeto latino básico junto con dígitos europeo-arábigos. También están incluidos en los "caracteres Unicode", los cuales son la base de los IDN.
API	Una Interfaz de Programación de Aplicaciones (API) es un conjunto de rutinas, protocolos y herramientas para crear software y aplicaciones. Una API puede ser para un sistema basado en web, sistema operativo o sistema de base de datos, y proporciona instalaciones para desarrollar aplicaciones para ese sistema con un lenguaje de programación determinado.
Espacio de código	Rango que define los límites inferior y superior para una codificación.
Puntos de código	Un punto de código o posición de código es cualquiera de los valores numéricos que componen el espacio de código. Se utilizan para distinguir ambos, el número de una codificación como una secuencia de bits, y el carácter abstracto de una representación gráfica particular (glifo).
Zona Raíz del DNS	La zona raíz es el directorio central para el DNS, que es un componente clave en la traducción de nombres de host legibles a direcciones IP numéricas.
EAI	La internacionalización de las direcciones de correo electrónico es una dirección de correo electrónico que requiere el uso de Unicode en todas las partes de la dirección de correo electrónico.
IANA	Autoridad de Números Asignados en Internet. Sus funciones incluyen: <ul style="list-style-type: none"> • mantenimiento del registro de los parámetros técnicos de protocolo de Internet • administración de ciertas responsabilidades asociadas con la Zona Raíz del DNS de Internet • asignación de recursos numéricos de Internet
ICANN	La Corporación para la Asignación de Nombres y Números en Internet (ICANN) es una corporación internacional sin fines de lucro responsable de la asignación del espacio de direcciones de Protocolo de Internet (IP), la asignación de identificadores de protocolo, la administración del sistema de nombres de dominio genéricos de alto nivel (gTLD) y de dominios de alto nivel con código de país (ccTLD), y las funciones de administración del sistema del servidor raíz.
IDN	Nombres de Dominio Internacionalizados Los IDN son nombres de dominio que

	incluyen caracteres correspondientes a la representación local de idiomas cuya escritura no utiliza las veintiséis letras básicas del alfabeto latino “a-z”, los números 0-9 y el guión “-“
IDNA	Nombres de Dominio Internacionalizados en Aplicaciones.
IDN ccTLD	Dominio de Alto Nivel con Código de País que incluye caracteres correspondientes a la representación local de idiomas cuya escritura no utiliza las veintiséis letras básicas del alfabeto latino “a-z”. Ejemplos: <ul style="list-style-type: none"> • .рф (Rusia) • .صر (Egipto) • .السعودية (Arabia Saudita)
IETF	El Grupo de Trabajo en Ingeniería de Internet (IETF) es una gran comunidad internacional abierta de diseñadores de red, operadores, proveedores e investigadores involucrados en la evolución de la arquitectura y el funcionamiento continuo de Internet. Está abierto a la todas las personas interesadas. El IETF desarrolla estándares de Internet y en particular estándares relacionados con el conjunto de protocolos de Internet (TCP/IP).
Lenguaje	El método de comunicación humana, ya sea hablado o escrito, que consiste en el uso de palabras de una manera estructurada y convencional.
Punycode	Es un algoritmo para representar Unicode con el subconjunto de caracteres limitados de ASCII admitido por el Sistema de Nombres de Dominio. Punycode está destinado a la codificación de etiquetas en el marco de los Nombres de Dominio Internacionalizados en Aplicaciones (IDNA).
Registrador	Una organización donde los usuarios registran nombres de dominio. El registrador guarda registros de la información de contacto y presenta la información técnica a un directorio central conocido como "registro".
Registro	La base de datos principal y acreditada de todos los nombres de dominio registrados en cada dominio de alto nivel.
RFC	Una Solicitud de Comentarios (RFC) es un documento formal del Grupo de Trabajo en Ingeniería de Internet (IETF) que es el resultado de la redacción del comité y la posterior revisión de las partes interesadas.
Código de escritura	El conjunto de letras o caracteres utilizados en la escritura, que representan los sonidos de un idioma.
Nombre de dominio de segundo nivel	En la jerarquía del Sistema de Nombres de Dominio (DNS), un dominio de segundo nivel (SLD o 2LD) es un dominio que está directamente debajo de un dominio de nivel superior (TLD). Por ejemplo, en ejemplo.com, ejemplo es el dominio de segundo nivel del TLD .com.
Etiqueta-U	Una "etiqueta U" es una cadena de caracteres Unicode válida según IDNA que incluye al menos un carácter no ASCII. Las conversiones entre Etiquetas-U y Etiquetas-A se realizan de acuerdo con la especificación de Punycode [RFC3492].
Software listo para	Software que está listo para la Aceptación Universal. Es un software que tiene la

la Aceptación Universal	capacidad de Aceptar, Almacenar, Procesar, Validar y Visualizar todos los Dominios de Alto Nivel por igual y todos los IDN, hipervínculos y direcciones de correo electrónico por igual.
Unicode	Un estándar universal de codificación de caracteres. Define la forma en que se representan los caracteres individuales en archivos de texto, páginas web y otros tipos de documentos. Unicode fue diseñado para admitir caracteres de todos los idiomas del mundo. Puede admitir aproximadamente 1 000 000 de caracteres y admite hasta 4 bytes para cada carácter. Ver: http://unicode.org
UTF	Formato de Transformación Unicode. Es una forma de transformar puntos de código Unicode en una secuencia de bytes. UTF-8 es el UTF preferido para manejar IDN y EAI. UTF-8 convierte Unicode a bytes de 8 bits.
M3AAWG	El Grupo de Trabajo Anti-Abuso de Mensajes, Malware y Móvil (M ³ AAWG) es donde la industria se reúne para trabajar contra: botnets, malware, spam, virus, ataques de denegación de servicio (DoS) y otras acciones abusivas en línea. Ver: https://www.m3aawg.org/
W3C	El Consorcio Mundial de Internet (W3C) es una comunidad internacional en la cual las organizaciones Miembro , el personal de tiempo completo y el público trabajan en conjunto para desarrollar estándares para la Web . Ver: https://www.w3.org/
ZWJ	Zero-Width Joiner o, Unión de longitud nula, es un carácter no imprimible que se utiliza en la composición tipográfica computarizada de algunos códigos de escritura complejos, como el código de escritura árabe o cualquier código de escritura índico. Cuando se coloca entre dos caracteres que de otro modo no estarían conectados, un ZWJ hace que se impriman en sus formas conectadas.
ZWNJ	Zero-Width Non-Joiner o Separador de longitud nula es un carácter no imprimible utilizado en la informatización de sistemas de escritura que utilizan ligaduras. Cuando se coloca entre dos caracteres que de otro modo no estarían conectados en una ligadura, un ZWNJ hace que se impriman en sus formas finales e iniciales, respectivamente. Esto también es un efecto de un carácter de espacio, pero se utiliza un ZWNJ cuando es deseable mantener las palabras más juntas o para conectar una palabra con su morfema.

Para obtener un glosario completo de la ICANN, consulte: <https://www.icann.org/resources/pages/glossary-2014-02-03-en>

RFC

RFC de PUNYCODE	
RFC 3492	<p>Punycode: Una codificación Bootstring de Unicode para Nombres de Dominio Internacionalizados en Aplicaciones (IDNA)</p> <p>La RFC 3492 describe a Punycode como:</p> <p><i>"una sintaxis simple y eficiente de codificación de transferencia diseñada para su</i></p>

	<p><i>uso con Nombres de Dominio Internacionalizados en Aplicaciones (IDNA)"</i></p> <p>Punycode transforma de forma única y reversible una cadena de caracteres Unicode en una cadena de caracteres ASCII. Esta RFC define un algoritmo general denominado Bootstring. Este algoritmo permite que una cadena de puntos de código básicos represente de manera única cualquier cadena de puntos de código extraídos de un conjunto más grande.</p> <p>https://tools.ietf.org/html/rfc3492</p>
RFC de IDN	
RFC 5890	<p>Nombres de Dominio Internacionalizados en Aplicaciones (IDNA): Definiciones y marco del documento</p> <p>Esta RFC describe el contexto de uso y el protocolo para una revisión de Nombres de Dominio Internacionalizados en Aplicaciones (IDNA).</p> <p>https://tools.ietf.org/html/rfc5890</p>
RFC 5891	<p>Protocolo de Nombres de Dominio Internacionalizados en Aplicaciones (IDNA)</p> <p>Esta RFC especifica el mecanismo de protocolo, denominado Nombres de Dominio Internacionalizados en Aplicaciones (IDNA), para registrar y buscar los IDN de una manera que no requiera cambios en el propio DNS.</p> <p>https://tools.ietf.org/html/rfc5891</p>
RFC 5892	<p>Los puntos de código Unicode y los Nombres de Dominio Internacionalizados para Aplicaciones (IDNA)</p> <p>La RFC 5892 especifica reglas para decidir si un punto de código, considerado de forma aislada o en contexto, es un candidato para su inclusión en un Nombre de Dominio Internacionalizado (IDN).</p> <p>https://tools.ietf.org/html/rfc5892</p>
RFC 5893	<p>Códigos de escritura de derecha a izquierda para Nombres de Dominio Internacionalizados en Aplicaciones (IDNA)</p> <p>Esta RFC proporciona una nueva regla Bidi para etiquetas de Nombres de Dominio Internacionalizados en Aplicaciones (IDNA), para el uso de códigos de escritura de derecha a izquierda en Nombres de Dominio Internacionalizados.</p> <p>https://tools.ietf.org/html/rfc5893</p>
RFC 5894	<p>Nombres de Dominio Internacionalizados en Aplicaciones (IDNA): antecedentes, explicación y fundamento</p> <p>Este documento informativo proporciona una descripción general de un sistema revisado para abordar las versiones más recientes de Unicode y proporciona material explicativo para sus componentes.</p> <p>https://tools.ietf.org/html/rfc5894</p>

<p>RFC 5895</p>	<p>Mapeado de caracteres para Nombres de Dominio Internacionalizados en Aplicaciones (IDNA) 2008</p> <p>Esta RFC describe las acciones que una implementación puede realizar entre la recepción de la entrada del usuario y la aprobación de puntos de código permitidos para el nuevo protocolo de IDNA (2008). Describe una operación que se aplicará a la entrada del usuario para preparar esa entrada de usuario para su uso en un protocolo "en la red". También incluye un procedimiento de implementación general para la asignación.</p> <p>https://tools.ietf.org/html/rfc5895</p>
<p>RFC de EAI</p>	
<p>RFC 6530</p>	<p>Descripción general y marco para el correo electrónico internacionalizado</p> <p>Este estándar presenta una serie de especificaciones que definen los mecanismos y las extensiones de protocolos necesarios para proporcionar compatibilidad completa a las direcciones de correo electrónico internacionalizadas. Este documento describe cómo encajan entre sí los diversos elementos de la internacionalización de las direcciones de correo electrónico y las relaciones entre las especificaciones principales asociadas con el transporte de mensajes, formatos de encabezado y manejo.</p> <p>https://tools.ietf.org/html/rfc6530</p>
<p>RFC 6531</p>	<p>Extensión SMTP para correo electrónico internacionalizado</p> <p>El documento define una extensión de Protocolo de transferencia de correo simple para que los servidores puedan anunciar la capacidad de aceptar y procesar direcciones de correo electrónico internacionalizadas y encabezados de correo electrónico internacionalizados.</p> <p>https://tools.ietf.org/html/rfc6531</p>
<p>RFC 6532</p>	<p>Encabezados de correo electrónico internacionalizado</p> <p>Este documento especifica una mejora del formato de mensaje de Internet y MIME que permite el uso de Unicode en las direcciones de correo y la mayoría del contenido del campo del encabezado. Este documento especifica una mejora del formato de mensaje de Internet (RFC 5322) y MIME que permite el uso directo de UTF-8, en lugar de ASCII, en valores de campo de encabezado, incluidas las direcciones de correo. Se define un nuevo tipo de medio, mensaje/global, para los mensajes que utilizan este formato extendido. Esta especificación también levanta la restricción MIME al tener codificaciones de transferencia de contenido sin identidad en cualquier subtipo del tipo de mensaje de nivel superior, de modo que las partes de mensaje/global se puedan transmitir de manera segura a través de la infraestructura de correo existente.</p> <p>https://tools.ietf.org/html/rfc6532</p>
<p>RFC 6533</p>	<p>Notificaciones internacionalizadas de estado de entrega y eliminación</p> <p>Esta especificación agrega un nuevo tipo de dirección para las direcciones de correo electrónico internacionales para que una dirección de destinatario original con caracteres que no sean ASCII pueda conservarse de forma correcta incluso después de la degradación. Esto también proporciona tipos de medios de retorno de contenido actualizado para</p>

	<p>notificaciones de estado de entrega y notificaciones de eliminación de mensajes para admitir el uso del nuevo tipo de dirección.</p> <p>https://tools.ietf.org/html/rfc6533</p>
--	--

Normas clave

<p>ISO 10646 (Unicode)</p>	<p>Para proporcionar una base técnica común para el procesamiento de información electrónica en varios idiomas, la Organización Internacional de Normalización (ISO) ha desarrollado una norma de codificación internacional denominada ISO 10646. La norma ISO 10646 proporciona un estándar unificado para la codificación de caracteres en todos los principales idiomas del mundo, incluidos los caracteres del chino tradicional y simplificado. Este gran conjunto de caracteres se denomina Conjunto Universal de Caracteres (UCS). El mismo conjunto de caracteres está definido por el estándar Unicode, que además define las propiedades de caracteres adicionales y otros detalles de la aplicación de gran interés para los implementadores.</p> <p>Unicode es un sistema de codificación de caracteres diseñado por el Consorcio Unicode para apoyar el intercambio, procesamiento y visualización de los textos escritos de los principales idiomas del mundo. ISO 10646 y Unicode definen varias formas de codificación de su repertorio común: UTF-8, UCS-2, UTF-16, UCS-4 y UTF-32.</p> <p>http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=63182</p>
<p>GB18030 (China)</p>	<p>GB 18030-2000 es un estándar del gobierno chino que especifica una página de códigos extendida para su uso en el mercado chino, además de UTF-8. El código de procesamiento interno para el repertorio de caracteres puede y debe ser Unicode; sin embargo, la norma estipula que los proveedores de software deben garantizar un ida y vuelta exitoso entre GB18030 y el código de procesamiento interno. Todos los productos que actualmente se venden o que se venderán en China deben planificar la migración de la página de códigos para admitir GB18030 sin excepción. GB18030 es un "estándar obligatorio" y el gobierno chino regula el proceso de certificación para reforzar la implementación de GB18030.</p> <p>http://icu-project.org/docs/papers/unicode-gb18030-faq.html</p>
<p>Norma técnica de Unicode 46: Procesamiento de compatibilidad de IDNA de Unicode</p>	<p>Esta especificación define una correspondencia consistente con los requisitos normativos del protocolo IDNA 2008, y que es lo más compatible posible con IDNA 2003. Para el software del cliente, esto proporciona un comportamiento que es más coherente con las expectativas del usuario sobre el manejo de los nombres de dominio con los datos existentes.</p> <p>http://unicode.org/reports/tr46/</p>

Recursos en línea

<p>API</p>	<p>API de Windows</p> <p>https://www.msdn.microsoft.com/enus/library/windows/desktop/ff818516%28v=vs.</p>
-------------------	--

	<p>85%29.aspx</p> <p>API de SharePoint https://msdn.microsoft.com/en-us/library/office/jj860569.aspx</p> <p>Lista de sufijos públicos https://publicsuffix.org/list/public_suffix_list.dat</p> <p>Lista de TLD autoritativa de la ICANN http://data.iana.org/TLD/tlds-alpha-by-domain.txt</p> <p>API de Android http://developer.android.com/guide/index.html</p> <p>API de MAC IOS https://developer.apple.com/library/mac/navigation</p> <p>Marco de .Net https://msdn.microsoft.com/en-us/library/system.text.encoding(v=vs.110).aspx</p>
Seguridad de Unicode	<p>Consideraciones de seguridad de Unicode http://www.unicode.org/reports/tr36</p> <p>Mecanismos de seguridad de Unicode http://www.unicode.org/reports/tr39</p>
Agrupaciones de caracteres de Unicode	<p>Planos de código Unicode http://en.wikipedia.org/wiki/Mapping_of_Unicode_character_planes</p> <p>Descripción general de GB18030 http://en.wikipedia.org/wiki/GB_18030</p> <p>Tabla autoritativa de correspondencia entre BG18030-2000 y Unicode http://source.icu-project.org/repos/icu/data/trunk/charset/data/xml/gb-18030-2000.xml</p> <p>Normalización de Unicode https://en.wikipedia.org/wiki/Unicode_equivalence</p>
Ataques de Unicode	<p>Sección 3.1, "Ataques de UTF-8" en el Informe técnico N.º 36 de Unicode http://unicode.org/reports/tr36/#UTF-8_Exploit</p> <p>Prácticas recomendadas del M3AAGW para la prevención del uso indebido de Unicode https://www.m3aawg.org/sites/default/files/m3aawg-unicode-best-practices-2016-02.pdf</p> <p>Tutorial y descripción general del uso indebido de Unicode del M3AAGW https://www.m3aawg.org/sites/default/files/m3aawg-unicode-tutorial-2016-02.pdf</p> <p>Consulte también: http://www.unicode.org</p>
Misceláneos	<p>URI</p>

<http://tools.ietf.org/html/rfc3986>

El Sistema de Nombres de Dominio: Una explicación no técnica: por qué la resolución universal es importante

<http://www.internic.net/faqs/authoritative-dns.html>

Glosario de la ICANN

<https://www.icann.org/resources/pages/glossary-2014-02-03-en>

Reconocimientos

Los autores agradecen a las siguientes personas por sus aportes y colaboración en este documento:

Eleeza Agopian
Gwen Carlson
Edmon Chung
Samantha Dickinson
Don Hollander
Chantal Lebrument
Antonietta Mangiacotti
Richard Merdinger
Ram Mohan
David Morrison
Carolyn Nguyen
Michael D. Palage
Kurt Pritz
André Schappo
Zheng Song
Lars Steffen
Andrew Sullivan
Dennis Tan
Winnie Yu

Cambios de versiones

De la versión 8 a la versión 9

- Transformaciones sugeridas corregidas de puntos de Unicode
- Se eliminó un enlace irrelevante de fuentes fidedignas