Introdução à Aceitação Universal

Mark Svancarek e Luisa Villa

Grupo de Gestão de Aceitação Universal

Sobre este documento

Objetivo

As tecnologias da Internet, inclusive dos componentes de nomenclatura, estão em constante evolução e transformação. Nos últimos anos, a ICANN tem lançado um grande número de TLDs com caracteres ASCII e domínios IDN de primeiro nível. Alguns exemplos são .nyc, .hsbc, .eco e .ストア. No entanto, a resposta para a mudança nesse cenário de nomenclaturas não tem sido rápida o bastante. Muitos aplicativos e serviços não estão sendo atualizados para gerenciar os novos TLDs. Isso afeta a experiência do usuário. Por exemplo:

- endereços de e-mail válidos não estão sendo aceitos;
- nomes de domínio são tratados incorretamente como termos de pesquisa na barra de endereços do navegador.

Muitos aplicativos e serviços não estão sendo atualizados para gerenciar esses novos TLDs. Isso afeta a experiência do usuário.

A menos que os softwares consigam reconhecer e processar os novos domínios, um estado conhecido como **Aceitação Universal**, não será possível proporcionar uma experiência consistente e positiva para os usuários da Internet. Sendo assim, este documento contém uma introdução geral à Aceitação Universal para ajudar no desenvolvimento de softwares habilitados para a Aceitação Universal.

Público-alvo

- Desenvolvedores de software
- Diretores técnicos (CTOs)
- A comunidade técnica em geral

Estrutura do documento

Parte 1 **Conceitos básicos da Aceitação Universal**, como o que é um nome de domínio e o DNS (Domain Name System, Sistema de Nomes de Domínio), ASCII e Unicode, Punycode, internacionalização de endereços de e-mail, entre outros.

Parte 2 Os **cinco critérios da Aceitação Universal**, bem como as **práticas recomendadas** para cada um desses critérios. Também contém **cenários de usuários** e **práticas que não estão em conformidade** com a Aceitação Universal, requisitos técnicos e dificuldades existentes no momento.

Parte 3 **Tópicos avançados**, como escritas da direita para a esquerda, algoritmo Bidi, normalização e case folding.

Parte 4 Contém o **glossário** e **recursos on-line úteis**.

Precisa de mais informações?

O UASG e a comunidade estão disponíveis para ajudar implementadores e desenvolvedores de software a saber o que é necessário.

- **Compartilhe** conosco suas ideias e sugestões sobre o assunto pelo e-mail info@uasg.tech
- Junte-se à **discussão sobre Aceitação Universal** em http://tinyurl.com/ua-discuss
- Para **saber mais** sobre esse trabalho, acesse http://www.icann.org/universalacceptance

Índice

Introdução

Um breve apanhado sobre a internacionalização de nomes de domínio

Na década de 70, os caracteres disponíveis para o registro de nomes de domínio eram limitados a um subconjunto de caracteres **ASCII** (letras a-z, dígitos 0-9 e o hífen "-"). Desde o primeiro registro .com, symbolics.com, em 1985, o número e as características dos nomes de domínio expandiram para refletir as necessidades do uso global cada vez maior da Internet como um recurso comum. Hoje em dia, a maioria dos usuários da Internet não tem o inglês como língua materna. No entanto, o idioma predominante na Internet é o inglês. Para ajudar na internacionalização da Internet, em 2003, a IETF (Internet Engineering Task Force, Força-tarefa de Engenharia da Internet) começou a publicar normas com diretrizes técnicas para a implantação de **IDNs (Internationalized Domain Names, Nomes de Domínio Internacionalizados)** usando um mecanismo de tradução para possibilitar o uso de representações não ASCII de nomes de domínio em várias escritas locais de diferentes regiões (por exemplo, 普遍接受-□□.世界, ua-test.世界 etc.).

Aceitação Universal é o estado em que todos os nomes de domínio e endereços de e-mail válidos são aceitos, validados, armazenados, processados e exibidos de maneira correta e consistente por todos os aplicativos, dispositivos e sistemas que utilizam a Internet.

A Diretoria da ICANN (Internet Corporation for Assigned Names and Numbers, Corporação da Internet para Atribuição de Nomes e Números) aprovou p processo para a introdução de novos ccTLDs (country code Top-Level Domains, Domínios de Primeiro Nível com Código de País) de IDNs em outubro de 2009, sendo que o primeiro deles foi adicionado à zona raiz em maio de 2010. Em junho de 2011, a Diretoria aprovou e autorizou o lançamento do Programa de Novos gTLDs (generic Top-Level Domains, Domínios Genéricos de Primeiro Nível), que incluiu novos ASCII e TLDs de IDNs. O primeiro lote de TLDs desse programa foi adicionado à zona raiz em 2013. O acréscimo de ccTLDs de IDNs e novos TLDs aumentou drasticamente o ritmo de lançamentos de TLDs na zona raiz.

Uma década após a publicação das diretrizes para IDNs da IETF, e graças ao Programa de Novos gTLDs da ICANN, mais de mil novos TLDs foram lançados. No entanto, apesar desse trabalho, muitos softwares e aplicativos ainda não estão prontos para a Aceitação Universal. Isso gera problemas para os usuários da Internet, inclusive para aqueles cujos idiomas são representados em escritas que incluem caracteres não ASCII.

A importância da Aceitação Universal

Para acompanhar o ritmo desse novo mundo de TLDs, é necessário construir novos softwares e atualizar os softwares e aplicativos antigos. O estado de conformidade plena com esse novo mundo de TLDs é chamado de **Aceitação Universal**.

A **Aceitação Universal** é o estado em que todos os nomes de domínio e endereços de e-mail válidos são **aceitos**, **validados**, **armazenados**, **processados** e **exibidos** de maneira correta e consistente por todos os aplicativos, dispositivos e sistemas que utilizam a Internet. Em outras palavras, todos os endereços da web válidos são resolvidos para o site esperado, e todos os endereços de e-mail válidos entregam as mensagens no destino esperado. Devido ao cenário dos nomes de domínio que está em constante mudança, muitos sistemas não reconhecem ou não processam adequadamente os novos nomes de domínio, principalmente porque é possível que estejam em um formato não ASCII, porque o software não conhece o TLD lançado recentemente ou porque o comprimento do TLD varia. O mesmo acontece com endereços de e-mail que incorporam essas novas extensões.

O **UASG (Universal Acceptance Steering Group, Grupo de Gestão de Aceitação Universal)**, uma iniciativa liderada pela comunidade, que abrange todo o setor e tem o suporte da ICANN, está trabalhando para promover o conhecimento, identificar e solucionar problemas associados à Aceitação Universal de Nomes de Domínio, a fim de ajudar a garantir uma experiência consistente e positiva para os usuários da Internet no mundo todo.

Parte 1: Conceitos básicos da Aceitação Universal

Esta seção contém uma visão geral dos termos e conceitos básicos, que devem ser entendidos antes de você seguir para as seções mais avançadas deste documento.

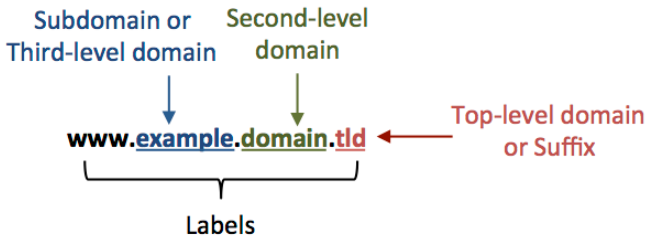
Nome de domínio

Um nome de domínio é uma cadeia de caracteres de texto com pontos usada como um identificador técnico (de fácil compreensão para os humanos) para computadores e redes na Internet. Por exemplo:

```
www.domain.tld
```

Como ler um nome de domínio:

- Cada ponto representa um **nível** na hierarquia do DNS (Domain Name System, Sistema de Nomes de Domínio).
- Um TLD (Top-Level Domain, Domínio de Primeiro Nível) é geralmente chamado de **sufixo**, porque aparece no final de um nome de domínio.
- As palavras ou caracteres individuais entre os pontos são chamados de **rótulos**. Para os idiomas ou as escritas que são lidos da **esquerda para a direita (LTR, left to right)**,[1] o rótulo mais à direita representa o domínio de primeiro nível.
- O segundo rótulo a partir do fim representa do **domínio de segundo nível**.
- Todos os rótulos que vierem antes do domínio de segundo nível são considerados **subdomínios** do domínio de segundo nível (às vezes chamados de **domínios de terceiro nível**).



DNS (Sistema de Nomes de Domínio)

Cada recurso na Internet tem um endereço atribuído a ele que é usado pelo IP (Internet Protocol, Protocolo da Internet). Como é difícil memorizar os endereços IP, o DNS oferece um mapeamento entre os endereços IP e os nomes de domínio que podem ser lidos pelas pessoas. Os servidores que coletivamente fornecem um DNS público existem em endereços conhecidos na Internet.

TLDs (Domínios de Primeiro Nível)

Os nomes de domínio que podem ser lidos pelas pessoas são gerenciados por organizações conhecidas como **registros.** Quando um nome de domínio é registrado, ele consiste em várias cadeias de caracteres de texto que representam vários níveis do domínio, cada um separado por um

[1] Os idiomas ou as escritas que são lidos da direita para a esquerda (RTL, right to left) serão tratados mais adiante.

caractere "**.**". Nas escritas LTR, o nível de domínio à direita é o TLD (Top-Level Domain, Domínio de Primeiro Nível). Alguns TLDs são delegados a países ou territórios específicos. Eles são chamados de **ccTLDs (Country Code TLDs, TLDs com código de país)**.

gTLDs (Domínios Genéricos de Primeiro Nível)

Desde 2013, a ICANN (a organização responsável pela criação e manutenção de atribuições de TLDs) tem aprovado a criação de um grande número de novos TLDs. Esses novos TLDs podem representar marcas, comunidades de interesse, comunidades geográficas (cidades, regiões) e conceitos mais genéricos. Coletivamente, todos esses novos TLDs são conhecidos como gTLDs (generic Top-Level Domains, Domínios Genéricos de Primeiro Nível).

Exemplos de TLDs comuns	Exemplos de ccTLDs	Exemplos de novos gTLDs
.com	China = .cn	.app
.gov	Alemanha = .de	.lawyer
.info	Estados Unidos = .us	.shopping
.org		.panasonic
		.osaka

Escritas e conjuntos de caracteres

Os idiomas são escritos usando sistemas de escrita. A maioria desses sistemas usa uma escrita, que é um conjunto de caracteres gráficos usados para a forma escrita de um ou mais idiomas. Um pequeno número de sistemas utiliza mais de uma escrita ao mesmo tempo. Esses caracteres ou escritas são reconhecidos pelas pessoas. No entanto, eles não podem ser usados por computadores. As escritas precisam ser codificadas de uma forma específica para serem processadas em um computador (por exemplo, para resolver um endereço da Web). O mecanismo para isso é chamado de **mapeamento de caracteres** ou **CCS (Coded Character Set, Conjunto de Caracteres Codificados)**, ou ainda uma **página de códigos**.[2] O mapeamento de caracteres associa caracteres a números específicos. Muitas páginas de códigos diferentes foram criadas com o tempo para diversas finalidades, mas, neste tópico, falaremos apenas de duas: ASCII e Unicode.

ASCII e Unicode

Nos exemplos de TLDs acima, todas as cadeias de caracteres de texto são representadas usando o conjunto de caracteres latinos. Esse conjunto de caracteres está incluído no esquema de codificação de caracteres ASCII (American Standard Code for Information Interchange, Código Padrão Americano para o Intercâmbio de Informação), também conhecido como US-ASCII. O ASCII é um esquema de codificação mais antigo, originalmente baseado no alfabeto inglês. Por motivos históricos, ele se tornou o esquema padrão para a codificação de caracteres na Internet. O ASCII usa apenas 7 bits por caractere, o que limita o conjunto a 128 caracteres, e nem todos eles podem ser usados em nomes de domínio. Os nomes de domínio são limitados aos caracteres A-Z, os números 0-9 e o hífen "-".

[2] Existem algumas sutilezas nesses termos que não são diretamente relevantes para a Aceitação Universal. Se quiser mais informações sobre a terminologia, um ponto de partida útil é: https://tools.ietf.org/html/rfc6365

Tabela ASCII - ISO 8859-1 (Latim-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	{	|	}	~	

Como a maioria dos sistemas de escrita não usa o conjunto de caracteres latinos, outras codificações alternativas também foram adotadas. O Unicode, também conhecido como **UCS (Universal Coded Character Set, Conjunto Universal de Caracteres Codificados)**, é capaz de codificar mais de um milhão de caracteres. Cada um desses caracteres Unicode é chamado de **ponto de código**. A forma mais comum do Unicode é chamada de **UTF-8 (Universal Coded Character Set Transform Format 8-bit, Formato de Transformação em 8 bits do Conjunto Universal de Caracteres Codificados).**

Para ver todas as tabelas de caracteres Unicode codificados acesse: http://unicode.org/charts

Nomes de Domínio Internacionalizados (IDNs) e Punycode

O uso do Unicode permite que os nomes de domínio contenham caracteres não ASCII. Conforme mencionado antes neste documento, os nomes de domínio que usam caracteres não ASCII são chamados de IDNs (Internationalized Domain Names, Nomes de Domínio Internacionalizados).[4] A parte internacionalizada de um nome de domínio pode estar em qualquer nível – não apenas no TLD, mas também nos outros rótulos.

Como o próprio DNS antigamente usava apenas ASCII,[5] foi necessário criar uma codificação adicional para que os pontos de código Unicode não ASCII fossem convertidos em cadeias de caracteres ASCII, e vice-versa. O algoritmo que implementa essa codificação de Unicode para ASCII é chamado de **Punycode;** as cadeias de caracteres resultantes são chamadas de **Rótulos A (A-Labels)**. É possível diferenciar os Rótulos A de um rótulo ASCII comum, porque eles sempre começam com os quatro caracteres seguintes:

- **xn--**

Esses caracteres são chamados de **prefixo ACE**.[6]

É possível reverter a transformação do Punycode: ele pode transformar de Unicode para Rótulo A e também de Rótulo A para Unicode (chamado de Rótulo U [U-Label]).

[3] Fonte: California State University. 1997. *ASCII - ISO 8859-1 (Latin-1) Table with HTML Entity Names.* http://web.calstatela.edu/faculty/jchen13/Docs/CS120/Lectures/ASCIITable_with_HTML_Entity_Names.htm

[4] É importante observar que nem todo caractere não ASCII é um IDN.

[5] Para saber o status atual, consulte http://tools.ietf.org/html/rfc6055#section-3

[6] O prefixo ACE (ASCII Compatible Encoding, Codificação Compatível com ASCII) é usado para distinguir os rótulos codificados por Punycode dos rótulos ASCII comuns.

O único uso definido por uma RFC[7] do algoritmo Punycode é para representar domínios internacionalizados. No entanto, em vez de implementar o Unicode, alguns desenvolvedores preferem usar o Punycode em outros cenários.

Exemplos de IDNs (fictícios)

exemplo.みんな	(Codificação Punycode = exemplo.xn--q9jyb4c)
大坂.info	(Codificação Punycode = xn--uesx7b.info)
みんな.大坂	(Codificação Punycode = xn--q9jyb4c.xn--uesx7b)

Para saber mais, veja as Perguntas Frequentes sobre IDNs: http://unicode.org/faq/idn.html

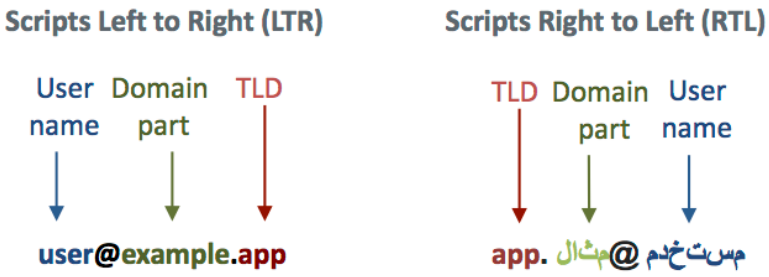
E-mail

Endereços e Internacionalização de Endereços de E-mail (EAI)

Os endereços de e-mail são divididos em duas partes.

1. Uma parte local (o nome do usuário, antes do caractere "@")
2. Um domínio (depois do caractere "@")

A parte do domínio pode conter qualquer TLD, inclusive um novo TLD. As duas partes podem ser Rótulos U Unicode.



OBSERVAÇÃO: outro formato, os Endereços de E-mail Estilo IDN, será discutido abaixo.

[7] RFC: (Request for Comments, Solicitação de Comentários). Consulte o Glossário de termos na Parte 4 deste documento para mais informações.

Exemplos de endereços de e-mail que incluem IDNs (fictícios)

usuário@exemplo.みんな	(Usa TLD internacionalizado)
usuário@大坂.info	(Usa domínio de segundo nível internacionalizado)
用口@exemplo.lawyer	(Usa nome de usuário internacionalizado e novo gTLD)

A EAI (Email Address Internationalization, Internacionalização de Endereços de E-mail) requer o uso do Unicode em todas as partes do endereço de e-mail. Todos os exemplos acima poderiam ser representados como EAI, e ele é o formato mais recomendado.

Geração de links dinâmicos (Linkification)

Softwares modernos, como aplicativos populares de processamento de texto ou planilhas, às vezes permitem ao usuário criar um hyperlink apenas digitando uma cadeia de caracteres semelhante a um endereço da Web, endereço de e-mail ou caminho de rede. Por exemplo, se você digitar "www.icann.org" em uma mensagem de e-mail, é possível que seja criado automaticamente um link clicável para `http://www.icann.org`, se o aplicativo tratar o "www." como um prefixo especial ou o".org" como um sufixo especial.

A "linkificação" (em tradução livre do inglês "linkification") deve funcionar de maneira consistente para todos os endereços da Web, endereços de e-mail ou caminhos de rede que têm o formato correto.

Parte 2: A Aceitação Universal em ação

Cinco critérios da Aceitação Universal

Conforme descrito na seção anterior, a Aceitação Universal é o estado em que todos os nomes de domínio e endereços de e-mail válidos são **aceitos, validados, armazenados, processados** e **exibidos** de maneira correta e consistente por todos os aplicativos, dispositivos e sistemas que utilizam a Internet. Esses cinco critérios são descritos abaixo.

1. Aceitar[8]	***Aceitar* ocorre sempre que um endereço de e-mail ou um nome de domínio é recebido como uma cadeia de caracteres de uma interface de usuário, arquivo ou API (Application Program Interface, Interface entre Programa e Aplicativo) usado por um aplicativo de software ou serviço on-line.** Os aplicativos e serviços permitem que os nomes de domínio e endereços de e-mail sejam: • Inseridos em interfaces do usuário, E/OU • Recebidos de outros aplicativos e serviços pelas APIs
2. Validar[9]	***Validar* pode ocorrer em muitos locais sempre que um endereço de e-mail ou nome de domínio é recebido ou enviado como uma cadeia de caracteres por um aplicativo ou serviço on-line.** A validação tem a função de garantir que as informações inseridas sejam válidas ou pelo menos definitivamente inválidas. Em outras palavras, a validação garante que a sintaxe da informação relevante esteja correta. No caso de nomes de domínio e endereços de e-mail, muitos programadores têm usado algumas heurísticas (por exemplo, para verificar se um TLD tem o número "correto" de caracteres ou se os caracteres são do conjunto de caracteres ASCII). No entanto, essas heurísticas não se aplicam mais, porque a Internet está mudando: • Os nomes de domínio e endereços de e-mail agora podem incluir caracteres Unicode (não ASCII) • A lista de TLDs está aumentando • Um TLD pode ter até 63 caracteres
3. Armazenar	***Armazenar* é o processo que ocorre sempre que um endereço de e-mail ou um nome de domínio é armazenado como uma cadeia de caracteres em um banco de dados ou arquivo usado por um aplicativo de software ou serviço on-line.** Os aplicativos e serviços podem exigir o armazenamento temporário e/ou por períodos mais longos de nomes de domínio e endereços de e-mail.

[8] O termo "aceitar" é tratado de maneira diferente de "validar" neste documento. Na prática, as funcionalidades podem se sobrepor.

[9] Os termos "aceitar" e "processar" são tratados de maneira diferente de "validar" neste documento. Na prática, as funcionalidades podem se sobrepor.

	Independentemente do tempo de vida dos dados, eles podem ser armazenados em: • Formatos definidos por RFC, OU • Outros formatos que possam ser traduzidos facilmente em formatos definidos por RFC nas duas direções (menos recomendado) Embora as RFCs exijam o uso do UTF-8, é possível encontrar outros formatos em codificações antigas. Consulte a seção "Práticas recomendadas" abaixo.
4. Processar[10]	***Processar* ocorre sempre que um endereço de e-mail ou nome de domínio é usado por um aplicativo ou serviço para executar uma atividade (por exemplo, para pesquisar ou classificar uma lista) ou é transformado em um formato alternativo (por exemplo, para armazenar ACSII como Unicode).** Processar significa usar nomes de domínio e cadeias de caracteres de e-mails em um recurso. É possível que seja realizada validação adicional durante o processamento. Não há um limite para o número de diferentes maneiras de processar nomes de domínio e endereços de e-mail (exemplos: "identificar todas as pessoas associadas à Nova Zelândia, porque elas têm um nome com um ccTLD `.nz`"; "identificar todos os farmacêuticos, porque eles têm um endereço de e-mail usuários@`exemplo.pharmacy`"; "identificar firewalls que podem filtrar solicitações do DNS que não se aplicam às políticas").
5. Exibir	**O processo *exibir* ocorre sempre que um endereço de e-mail ou um nome de domínio é renderizado em uma interface de usuário.** A exibição de nomes de domínio e endereços de e-mail é geralmente algo simples quando as escritas usadas são compatíveis com o sistema operacional em execução e quando as cadeias de caracteres são armazenadas em Unicode. Se essas condições não forem atendidas, talvez sejam necessárias transformações específicas para determinados aplicativos.

Cenários de usuários

Os exemplos e as definições acima podem dar a impressão de que a Aceitação Universal diz respeito apenas a sistemas de computadores e serviços on-line. Na realidade, ela também envolve as pessoas que usam esses sistemas e serviços.

Seguem abaixo alguns exemplos de atividades que exigem a Aceitação Universal:

Registrar um novo TLD	Uma organização adota um TLD de "marca" para dar aos clientes uma experiência diferenciada com endereços de e-mail no formato `nomedocliente`@`exemplo.marca`. Aceitação Universal significa:

[10] O termo "processar" é tratado de maneira diferente de "validar" neste documento. Na prática, as funcionalidades podem se sobrepor.

	• Os aplicativos da Web aceitam esses novos endereços de e-mail "@exemplo.marca" como válidos, da mesma forma que aceitariam TLDs como .com, .net ou .org.
Acessar um gTLD	Um usuário acessa um site, cujo nome de domínio contém um novo TLD, digitando um endereço no navegador ou clicando no link em um documento. Aceitação Universal significa: • Embora o TLD seja novo, qualquer navegador escolhido pelo usuário exibe o endereço da Web na sua forma nativa e acessa o site da maneira esperada. O navegador não exibe o texto em Punycode, a menos que isso seja útil ao usuário de alguma forma.
Usar um endereço de e-mail que contenha um novo gTLD como uma identidade on-line	Um usuário adquire um endereço de e-mail com a parte do domínio usando um novo gTLD e usa esse endereço de e-mail como identidade para acessar suas contas on-line em bancos e companhias aéreas. Aceitação Universal significa: • Embora o domínio usado no endereço de e-mail seja novo, o site do banco ou da companhia aérea aceita o endereço exatamente como se fosse um TLD já estabelecido, como .biz ou .eu.
Acessar um IDN	Um usuário acessa um URL de IDN digitando um endereço no navegador ou clicando no link em um documento. Aceitação Universal significa: • Mesmo que o nome de domínio contenha caracteres diferentes das configurações de idioma no computador do usuário, qualquer navegador escolhido pelo usuário exibe o endereço da Web conforme esperado e acessa o site sem problemas.
Usar um endereço de e-mail internacionalizado para enviar e receber mensagens	Um usuário adquire vários endereços de e-mail, e alguns deles são internacionalizados (por exemplo, ĩnfo@普遍接受-测试.世界). Aceitação Universal significa: • O usuário consegue enviar e receber mensagens por qualquer endereço de e-mail com qualquer cliente de e-mail.
Usar um endereço de e-mail internacionalizado como identidade on-line	Um usuário adquire um endereço de e-mail EAI e usa esse endereço de e-mail como identidade para acessar suas contas on-line em bancos e companhias aéreas. Aceitação Universal significa: • O site do banco ou da companhia aérea aceita a identidade EAI exatamente como se fosse qualquer outra identidade de e-mail.

Criar um hyperlink dinamicamente em um aplicativo	Um usuário digita um endereço da Web em um documento ou mensagem de e-mail. Aceitação Universal significa: • As regras usadas pelo aplicativo para gerar um hyperlink automaticamente são as mesmas, mesmo que o endereço seja do tipo EAI ou contenha um novo TLD.
Desenvolver um aplicativo	Um desenvolvedor escreve um aplicativo que acesse recursos da Web. Aceitação Universal significa: • As ferramentas usadas pelo desenvolvedor incluem bibliotecas habilitadas para a Aceitação Universal com suporte para Unicode, IDNs e EAI.

Não conformidade com as práticas da Aceitação Universal

Os exemplos a seguir são considerados **práticas negativas**:

▯	Exibir texto em Punycode ao usuário sem um benefício relevante para ele. Por exemplo, mostrar o mapeamento entre um Rótulo U e um Rótulo A.
▯	Exigir que um usuário insira texto em Punycode ao registrar um novo endereço de e-mail ou exigir que um usuário insira texto em Punycode ao registrar um novo domínio hospedado.
▯	Validar a sintaxe do nome de domínio ou do endereço de e-mail usando critérios desatualizados ou recursos on-line de nomes de domínio não oficiais.
▯	Usar uma lista desatualizada de TLDs mesmo que novos TLDs sejam adicionados regularmente.
▯	Expor o uso interno de texto em Punycode aos usuários. Por exemplo, converter de EAI para um endereço de e-mail estilo IDN ao responder a um usuários de EAI.
▯	Tratar alguns nomes de domínio como termos de pesquisa, em vez de nomes de domínio, porque o aplicativo não os reconhece como tal.
▯	Definir bloqueadores de spam para bloquear automaticamente TLDs inteiros.

Requisitos técnicos para a Aceitação Universal

Requisitos gerais

Um aplicativo ou serviço compatível com a UA (Universal Acceptance, Aceitação Universal):

1. **É compatível com todos os nomes de domínio, independentemente do conjunto ou do número de caracteres.**

   Consulte a RFC 5892.

2. **Aceita vários conjuntos de caracteres válidos para nomes de domínio e endereços de e-mail.**

   Ou seja, aceita o uso de pontos de código Unicode.

3. **Renderiza corretamente todos os pontos de código em cadeias de caracteres Unicode.**

   Consulte a RFC 3490.

4. **Renderiza corretamente cadeias de caracteres RTL (Right to Left, Direita para a Esquerda), como no árabe e hebraico.**

   Para mais informações sobre a escrita RTL, consulte a RFC 5893.

5. **Comunica os dados entre aplicativos e serviços em formatos compatíveis com Unicode e que podem ser convertidos para/de UTF-8.**

   Para mais informações sobre o UTF-8, consulte a RFC 3629.

6. **Oferece APIs públicas compatíveis com Unicode e UTF-8.**

7. **Oferece APIs particulares compatíveis com Unicode e UTF-8.**

   As APIs particulares se aplicam apenas a chamadas de atendimento internas feitas pelo mesmo fornecedor.

8. **Armazena os dados de usuários em formatos compatíveis com Unicode e que podem ser convertidos para/de UTF-8.**

   Essas conversões podem ser visualizadas apenas pelo proprietário do serviço/produto.

9. **É compatível com todas as cadeias de caracteres de nomes de domínio na lista oficial de TLDs da ICANN e na lista de sufixos públicos da comunidade, independentemente do conjunto ou do número de caracteres.**

   Consulte https://newgtlds.icann.org/en/program-status/delegated-strings.

10. **Envia e recebe e-mails de destinatários, independentemente do nome de domínio ou do conjunto de caracteres.**

    Consulte a RFC 6530.

11. **Trata os endereços EAI do mesmo modo que seus equivalentes em Punycode (formato de e-mail IDN).**

Considerações para desenvolvedores

Como muitos sistemas de software existentes contêm suposições de códigos incorporados sobre domínios e endereços de e-mail, talvez sejam necessárias mudanças nos códigos para reconhecer IDNs e novos TLDs. Esta seção explica como os desenvolvedores podem incorporar mudanças nos códigos que habilitarão a Aceitação Universal de todos os novos TLDs.

Um princípio importante para habilitar a Aceitação Universal: A Lei de Postel

Na RFC 793, Jon Postel formulou o **Robustness Principle (Princípio da Robustez)**, agora conhecido com **Lei de Postel**, como uma diretriz para a implementação do então novo TCP. Na computação, o Princípio de Robustez é uma diretriz de design geral para softwares:

> "Seja conservador no que você faz; seja liberal no que você aceita dos outros."

Em outras palavras, seja conservador no que você envia e seja liberal no que aceita. Essa também é uma boa abordagem ao lidar com as peculiaridades da Aceitação Universal implementadas atualmente no ecossistema.

Práticas recomendadas para desenvolver e atualizar softwares habilitados para a UA

Aceitar	
▯	Sempre ofereça equivalentes em Unicode. Os usuários devem ter permissão, mas não obrigação, de inserir texto compatível com codificação ASCII (ou "Punycode") no lugar do seu equivalente em Unicode. No entanto, o Unicode deve ser exibido por padrão, sendo que o texto em Punycode é exibido para o usuário apenas quando for vantajoso para ele.
!	**Não** gere endereços de e-mail estilo IDN, mas **tenha suporte** para eles, caso sejam apresentados por algum software de terceiros.
▯	Qualquer elemento da interface do usuário que exigir que o usuário digite um nome de domínio ou endereço de e-mail precisa ser compatível com Unicode, rótulos com até 63 caracteres, e cadeias de caracteres com até 253 caracteres. • Consulte a RFC 1035.

Validar	
▯	Faça o mínimo de validação necessária. **Faça apenas validações se forem necessárias para a operação do aplicativo ou do serviço.** Essa é a maneira mais eficiente de garantir que todos os nomes de domínio válidos sejam aceitos nos sistemas.
▯	Reconheça que entradas sintaticamente corretas podem não representar nomes de domínio nem endereços de e-mail usados atualmente na Internet.
!	Se a validação for necessária, considere o seguinte: • Verifique a parte do TLD de um nome de domínio em comparação a uma tabela oficial: Alguns exemplos de tabelas oficiais que podem ser usadas são: o http://www.internic.net/domain/root.zone o http://data.iana.org/TLD/tlds-alpha-by-domain.txt Consulte também: https://www.icann.org/en/system/files/files/sac-070-en.pdf • Faça uma consulta do nome de domínio com relação ao DNS.

	○ Considere usar a GETDNS API (http://getdnsapi.net/). • Exija que o endereço de e-mail seja informado mais de uma vez para descartar os erros de digitação. • Valide os caracteres nos rótulos apenas para determinar se o Rótulo U não contém pontos de código "NÃO PERMITIDO" ("DISALLOWED") ou pontos de código que não estão atribuídos na sua versão em Unicode. ○ Consulte a RFC 5892. • Limite a validação de rótulos para uma quantidade pequena de regras de rótulos inteiros definidas nas RFCs. ○ Consulte a RFC 5894. • Se uma cadeia de caracteres semelhante a um nome de domínio contiver o caractere árabe do ponto final "۔" (U+06D4) ou o caractere ideográfico do ponto final "。" (U+3002), converta-o em ponto final "." (U+002E). • Certifique-se de que o produto ou recurso reconheça números corretamente. ○ Por exemplo: numerais ASCII e representações asiáticas ideográficas de números devem ser tratados como números.

Armazenar	
▯	Os aplicativos e serviços devem ser compatíveis com os padrões adequados de Unicode.
▯	As informações devem ser armazenadas em UTF-8 (Unicode Transformation Format, Formato de Transformação Unicode) sempre que possível. Alguns sistemas também podem exigir compatibilidade com o UTF-16, mas geralmente há uma preferência pelo UTF-8. UTF-7 e UTF-32 devem ser evitados.
!	Considere todos os cenários completos antes de converter Rótulos A (Punycode) em Rótulos U, e vice-versa, ao armazenar. Pode ser interessante manter apenas Rótulos U em um arquivo ou banco de dados, porque isso simplifica as pesquisas e classificações. No entanto, a conversão pode ter implicações na interoperabilidade com aplicativos e serviços mais antigos que não usam Unicode. Considere armazenar e indexar nos dois formatos.
▯	Marque claramente os endereços de e-mail e nomes de domínio durante o armazenamento para facilitar o acesso. Instâncias em que os endereços de e-mail e nomes de domínio foram arquivados no campo "author" (autor) de um documento ou "contact info" (informações de contato) em um arquivo de registro resultaram na perda do endereço original.
▯	Se você *não* armazenar em Unicode, deverá ser capaz de fazer a correspondência das cadeias de caracteres em vários formatos. Por exemplo, uma pesquisa por `exemplo.みんな` também deverá encontrar `exemplo.xn--q9jyb4c`.

Processar	
▯	Certifique-se de que todas as respostas do servidor têm Unicode especificado no tipo de conteúdo.
▯	Especifique Unicode no cabeçalho HTTP do servidor da Web e diretamente em um arquivo da Web. • Todos os arquivos da Web devem incluir o conjunto de caracteres UTF-8. • É importante garantir que a codificação seja especificada em todas as respostas.
!	Considere todos os cenários completos antes de converter Rótulos A (Punycode) em Rótulos U, e vice-versa, ao processar. Pode ser interessante manter apenas Rótulos U em um arquivo ou banco de dados, porque isso simplifica as pesquisas e classificações. No entanto, a conversão pode ter implicações na interoperabilidade com aplicativos e serviços mais antigos que não usam Unicode. Considere armazenar nos dois formatos.
▯	Certifique-se de que o produto ou recurso seja compatível com classificação, pesquisa e agrupamento, de acordo com as especificações do local/idioma, e que ofereça capacidade de pesquisa e classificação em vários idiomas.
▯	Não use a codificação URL para nomes de domínio: • `exemplo.みんな` está correto • `exemplo.%E3%81%BF%E3%82%93%E3%81%AA` não está correto
▯	Uma vez que o padrão Unicode é expandido continuamente, os pontos de código não definidos quando o aplicativo ou serviço foi criado devem ser verificados para garantir que eles não "interromperão" a experiência do usuário. Fontes ausentes no sistema operacional subjacente podem fazer com que alguns caracteres não sejam exibidos (muitas vezes o caractere "□" é usado para representá-los), mas essa situação não deve resultar em uma falha fatal.
▯	Use APIs compatíveis com Unicode.
▯	Use os documentos mais recentes do protocolo e tabelas de IDNA (Internationalized Domain Names in Applications, Nomes de Domínio Internacionalizados em Aplicativos) para IDNs: • RFC 5891 • RFC 5892
▯	Processe no formato UTF-8 sempre que possível.
▯	Faça o upgrade de aplicativos e servidores/serviços juntos. Se o servidor usar Unicode e o cliente não usar Unicode, ou vice-versa, será necessário converter os dados para cada página de código sempre que os dados forem transferidos entre o servidor e o cliente.
▯	Faça análises da codificação para enviar ataques de *buffer overflow*.

	Ao fazer a transformação de caracteres, as cadeias de texto podem aumentar ou diminuir significativamente.

Exibir	
▯	Exiba todos os códigos de pontos Unicode compatíveis com o sistema operacional subjacente. Se um aplicativo tiver seus próprios conjuntos de fontes, é necessário oferecer um suporte abrangente de Unicode para o conjunto de fontes disponíveis no sistema operacional.
▯	Ao desenvolver um aplicativo ou um serviço, considere os idiomas compatíveis e certifique-se de que os sistemas operacionais e os aplicativos aceitem esses idiomas.
▯	Converta dados diferentes de Unicode para Unicode antes de exibir. Por exemplo, o usuário final deve ver “exemplo.みんな”, em vez de “exemplo.xn--q9jyb4c”. (Essa conversão é um exemplo de processamento compatível com a UA.)
▯	Exiba Unicode por padrão. Exiba o texto em Punycode para o usuário apenas quando isso for vantajoso para ele.
!	Lembre-se de que endereços que usam uma combinação de escritas serão mais comuns. • Alguns caracteres Unicode podem parecer iguais ao olho humano, mas diferente para os computadores. • Não presuma que cadeias de caracteres com mais de uma escrita sejam destinadas a fins maliciosos, como phishing. • Se a interface do usuário chamar a atenção do usuário para as cadeias de caracteres, certifique-se de que isso seja feito de maneira que não prejudique os usuários de escritas não latinas. Saiba mais sobre considerações de segurança com Unicode em: http://unicode.org/reports/tr36
▯	Use o processamento de compatibilidade de IDNA para Unicode a fim atender às expectativas dos usuários. Para saber mais, acesse: http://unicode.org/reports/tr46
▯	Esteja atento a caracteres não atribuídos ou não permitidos para nomes de domínio. • Consulte a RFC 5892.

Unicode	
▯	Use APIs compatíveis com Unicode.
▯	Não crie suas próprias APIs para: • Converter formatos de cadeias de caracteres

	• Determinar a escrita usada em uma cadeia de caracteres • Determinar se uma cadeia de caracteres contém uma combinação de escritas • Decomposição/normalização de Unicode
▯	Não use UTF-7 nem UTF-32. • O UTF-7 geralmente não é usado como uma representação nativa nos aplicativos, porque é um formato complicado de processar. Apesar da vantagem de ser menor, em comparação à combinação do UTF-8 com codificação QP ou Base64, o Internet Mail Consortium não recomenda o uso desse formato. • A principal desvantagem do UTF-32 é o uso excessivo de espaço: quatro bytes por ponto de código. Os caracteres não BMP são tão raros na maioria dos textos[citação necessária], que eles podem ser considerados inexistentes em termos de tamanho, fazendo com que o UTF-32 tenha o dobro do tamanho do UTF-16 e até quatro vezes mais que o UTF-8.
▯	Use o Unicode em cookies para que eles sejam lidos corretamente pelos aplicativos.
▯	Use os documentos do protocolo e tabela de IDNA de 2008: • RFC 5891 • RFC 5892
▯	Não use o IDNA de 2003; em quase todos os casos ele foi substituído pelo IDNA de 2008.
▯	Não suponha automaticamente que as APIs externas aceitem dados que tenham sido convertidos por [11]NFKC.
!	Mantenha as tabelas de IDNA e Unicode consistentes com as versões. Por exemplo, a menos que o aplicativo realmente execute as regras de classificação no documento Tabelas (RFC 5892), as tabelas de IDNA dele deverão ser derivadas da versão de Unicode que seja mais compatível no sistema. Assim como no registro, as tabelas não precisam refletir a versão mais recente de Unicode, mas devem ser consistentes.
!	Valide os caracteres nos rótulos apenas para determinar se o Rótulo U não contém pontos de código "NÃO PERMITIDO" ("DISALLOWED")[12] ou pontos de código que não estão atribuídos na sua versão em Unicode.
▯	Limite a validação de rótulos para uma quantidade pequena de regras de rótulos inteiros: • Sem marcas combinadas no início • Condições bidirecionais atendidas, se forem exibidos caracteres da direita para a esquerda

[11] **NFKC** (*Normalization Form Compatibility Composition, Composição de Compatibilidade para Forma de Normalização)*: os caracteres são decompostos por compatibilidade e, em seguida, recompostos por equivalência canônica. Consulte: http://unicode.org/reports/tr15

[12] DISALLOWED (NÃO PERMITIDO): pontos de código que não devem ser incluídos em IDNs. Consulte: https://tools.ietf.org/html/rfc5892

	• Todas as regras contextuais associadas aos caracteres de ligação (e caracteres CONTEXTJ[13] em geral) são testadas
!	Não use o UTF-16, a menos que seja explicitamente necessário (como no caso de certas APIs do Windows). Ao usar o UTF-16, observe que os 16 bits só podem conter caracteres no intervalo de 0x0 a 0xFFFF, e uma complexidade maior é usada para armazenar valores fora desse intervalo (0x10000 a 0x10FFFF). Isso é feito usando pares de unidades de códigos conhecidas como "surrogates" ("substitutos"). Se o uso de pares substitutos não for testado corretamente, isso pode resultar em erros complexos e possíveis brechas na segurança.

Linkificação ("Linkification")	
▯	Se uma cadeia de caracteres semelhante a um nome de domínio contiver o caractere árabe do ponto final "-" (U+06D4) ou o caractere ideográfico do ponto final "。" (U+3002), converta-o em ponto final "." (U+002E).

Geral	
▯	Use recursos oficiais para validar nomes de domínio. Não faça suposições heurísticas, como "todos os TLDs têm 2, 3, 4 ou 6 caracteres".
▯	Certifique-se de que o produto ou recurso reconheça números corretamente. Por exemplo, numerais ASCII e representações de números ideográficas asiáticas devem ser tratados como números.
!	Procure endereços de e-mail em lugares inesperados: • Artista/autor/fotógrafo/metadados de direitos autorais • Metadados de fonte • Registros de contato do DNS • Informações da versão binária • Informações de suporte • Informações de contato de OEM • Inscrição, feedback e outros tipos de formulários
!	Procure possíveis caminhos de IRI[14] em lugares inesperados:

[13] CONTEXTJ: regra contextual para controles de ligação. Consulte: https://tools.ietf.org/html/rfc5892

[14] IRI: (Internationalized Resource Identifiers, Identificadores de Recursos Internacionalizados). Consulte: https://www.ietf.org/rfc/rfc3987.txt

	• Nomes de máquinas com um único rótulo, independentemente da página de códigos carregada no sistema • Nomes de máquinas totalmente qualificados, independentemente da página de códigos carregada no sistema
▯	Use o GB18030 (China) para suporte ao idioma chinês[15], além do UTF-8.
!	Restrinja os pontos de códigos permitidos ao gerar novos nomes de domínio e endereços de e-mail: Todos os produtos que usam endereços de e-mail devem aceitar endereços de e-mail internacionalizados, permitindo o uso de caracteres > U+007f. Ou seja, nenhum caractere > U+007f deve ser "não permitido". No entanto, um aplicativo ou serviço não precisa permitir o uso de todos esses caracteres quando um usuário criar um novo IDN ou endereço de e-mail. Use apenas esta lista de caracteres permitidos para IDNs: http://unicode.org/reports/tr36/idn-chars.txt Impedir que certos IDNs ou endereços de e-mail sejam criados pode evitar algumas possíveis preocupações com a segurança e acessibilidade. (OBSERVAÇÃO: a Lei de Postel ainda exige que os softwares aceitem essas cadeias de caracteres se elas forem apresentadas.)
!	Lembre-se de que a Aceitação Universal não pode ser sempre medida apenas por casos de testes automatizados. Por exemplo, nem sempre é possível testar como um aplicativo ou protocolo usa os recursos de rede e, às vezes, é melhor verificar a conformidade por meio de uma revisão de especificações funcionais e revisão de design.
!	Não suponha automaticamente que, só porque um componente não chama diretamente APIs para a resolução de nomes nem usa diretamente endereços de e-mail, significa que ele não é afetado por eles. Entenda como os nomes de rede são obtidos pelo componente; nem sempre é pela a interação do usuário. Seguem abaixo alguns exemplos de como um componente pode obter um nome de rede: • Política de grupo • Consulta LDAP • Arquivos de configuração • Registro do Windows • Transferência para/de outro componente/recurso
▯	Faça análises da codificação para enviar ataques de *buffer overflow*. • No Unicode, as cadeias de caracteres podem ser ampliadas com o uso variado de letras maiúsculas e minúsculas: Fluß → FLUSS → fluss • Ao fazer a conversão de caracteres, o texto pode aumentar ou diminuir significativamente.

[15] O GB 18030-2000 é um padrão do governo chinês que especifica uma página de códigos estendida para ser usada no mercado chinês. Consulte: http://icu-project.org/docs/papers/unicode-gb18030-faq.html

Fontes oficiais para nomes de domínio

Zona raiz do DNS

Existem algumas opções para lista oficial de TLDs. A primeira opção é a própria zona raiz do DNS. Ela é assinada por DNSSEC, então, a lista é autenticada adequadamente. Você pode obter a zona raiz em qualquer um dos links a seguir:

- 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 sufixos públicos

A PSL (Public Suffix List, Lista de Sufixos Públicos), gerenciada por voluntários da Mozilla Foundation, é uma lista atualizada de sufixos de nomes de domínio. Essa lista é um conjunto de nomes do DNS ou de caracteres curinga concatenados com pontos e codificados usando o UTF-8. Se você precisar usar a PSL como uma fonte oficial para nomes de domínio, seu software deve receber atualizações da PSL regularmente. Não incorpore cópias estáticas da PSL no software sem nenhum mecanismo de atualização. Você pode usar o link abaixo para que seu aplicativo faça o download da lista atualizada periodicamente. A lista é atualizada uma vez ao dia pelo Github:

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

Outros desafios

Geral

Codificação variada de IDNs	Em alguns aplicativos, os IDNs são codificados: • Em Punycode, conforme o IDNA, se o nome for identificado como um nome da Internet, MAS • Em UTF-8, se o nome for identificado como um nome na rede de área local (“intranet”)
Mecanismo para detectar e converter conjuntos de caracteres	Alguns aplicativos de e-mail mais antigos eram codificados em uma página de códigos local e não tinham um mecanismo definido para detectar e converter conjuntos de caracteres da maneira necessária. Isso ocorria principalmente no cabeçalho de e-mails (PARA, CC, CCO, Assunto).
Falha ao executar protocolos não DNS	Alguns aplicativos que aceitam o IDNA (por exemplo, IE7 e versões posteriores) apresentam erros ao executar protocolos não DNS. Isso pode afetar o acesso a recursos que usam protocolos não DNS.
Mecanismo para gerenciar vários endereços de e-mail em uma única identidade de usuário	Quando um usuário usa vários endereços de e-mail como alias, pode ser difícil gerenciar esses endereços como uma única identidade de usuário. Os programas de e-mail podem direcionar o tráfego para esses alias para a mesma caixa de entrada, mas o aplicativo ainda pode entender que esses e-mails pertencem a identidades diferentes.

Dica para desenvolvedores de software	
▯	Ao permitir que um usuário gere um nome de domínio ou endereço de e-mail, considere a possibilidade de bloquear o uso de caracteres muito parecidos visualmente a fim de evitar o risco de ataques homográficos. Use apenas esta lista de caracteres permitidos para IDNs: http://unicode.org/reports/tr36/idn-chars.txt

E-mails estilo IDN e porque não são o mesmo que EAI

Na EAI somente o Unicode é usado; Rótulos A (Punycode) não são permitidos. Ainda assim, às vezes, os desenvolvedores adaptam softwares de e-mail e serviços para executar endereços de e-mail estilo IDN, em vez de fazer a conversão completa para Unicode.

Como os IDNs podem ser codificados em Punycode, alguns softwares existentes permitem que a parte de IDN dos endereços de e-mail seja representada em ASCII ou Unicode. Por exemplo, alguns softwares tratarão esses

Nem todos os softwares tratarão esses dois e-mails estilo IDN como equivalentes funcionais

usuário@exemplo.みんな = usuário@exemplo.xn--q9jyb4c

dois endereços de “e-mail estilo IDN” de maneira equivalente para todas as finalidades (enviar, receber e pesquisar):

No entanto, alguns softwares não tratarão esses endereços como equivalentes com tanta eficiência, embora os dois sejam válidos, porque não há um requisito para que os softwares processem um Rótulo A (ou seja, “xn--q9jyb4c”) em seu Rótulo U equivalente (ou seja, “みんな”) antes de fazer a comparação. Não é possível prever como isso pode afetar a experiência do usuário. Talvez o usuário fique um pouco confuso se alguns softwares converterem Rótulos U em Rótulos A para fins de “compatibilidade”; como as mensagens são respondidas ou encaminhadas, é possível que seja observado um aumento do número de endereços que são visivelmente diferentes para um usuário ou que não executam as funcionalidades de pesquisa e classificação da maneira esperada.

No exemplo abaixo, alguns softwares talvez tentem converter até mesmo a parte local do endereço de e-mail usando Punycode, criando algo como um Rótulo A na parte local do endereço. Isso não é permitido de acordo com as RFCs existentes, e é muito provável que resulte em erros ao receber e-mails de determinados sistemas e problemas nos processos de pesquisa e classificação, conforme explicado acima.

Nunca converta a parte local de um endereço de e-mail usando Punycode

- 用□@exemplo.みんな
- xn--youq53b@exemplo.xn--q9jyb4c

Softwares e serviços eficientes habilitados para a UA podem conseguir executar e tratar esses formatos de maneira idêntica, mesmo os que não estiverem em conformidade com a RFC. Ainda assim, os softwares habilitados para a UA não devem gerar apenas endereços de e-mail EAI verdadeiros.

Os desafios da linkificação

Softwares modernos às vezes permitem ao usuário criar um hyperlink automaticamente apenas digitando uma cadeia de caracteres semelhante a um endereço da Web, endereço de e-mail ou caminho de rede. Por exemplo, se você digitar “www.icann.org” em uma mensagem de e-mail, é possível que seja criado automaticamente um link clicável para `http://www.icann.org` se o aplicativo tratar o “www.” como um prefixo especial ou o“.org” como um sufixo especial.

A “linkificação” (em tradução livre do inglês “linkification”) deve funcionar de maneira consistente para todos os endereços da Web, nomes de e-mail ou caminhos de rede que têm o formato correto.

“Linkificação” é quando um aplicativo aceita uma cadeia de caracteres e determina dinamicamente se deve criar um hyperlink para um URL (local na Internet) ou um endereço de e-mail (mailto:)

A linkificação usa algoritmos e regras criados por desenvolvedores de software para determinar se uma cadeia de caracteres deve ser considerada um link ou não. Outra questão relacionada é como as pessoas identificam uma cadeia de caracteres como um nome de domínio. Embora os navegadores, clientes de e-mail e editores de texto sejam lugares óbvios, existem muitos outros aplicativos que tomam essas decisões.

Práticas recomendadas

1. Tente criar um link com base nos prefixos de protocolo explícitos (por exemplo, “http://”, ftp://”, “mailto:”), mas só complete a ação se o resta da cadeia de caracteres estiver correto.

Exemplo de cadeia de caracteres	**Comportamento esperado/resultado**
exemplo.com	Não ocorre a linkificação porque o protocolo está ausente e não é inferido.
http://exemplo.com	Um hyperlink é criado porque o protocolo está explícito.
http:exemplo.com	Não ocorre a linkificação porque a sintaxe está errada (faltam os “//”).
http://exemplo.a	Não ocorre a linkificação porque as políticas da ICANN exigem que todo TLD tenha pelo menos dois caracteres. Obs.: é possível que esta sintaxe seja aceita em uma rede interna.
http://exemplo..ab	Não ocorre a linkificação porque a sintaxe está errada (pontos consecutivos).
http:// 普遍接受-□□.世界	Um hyperlink é criado porque o protocolo está explícito.

2. Tente criar um link com base em prefixos de protocolo implícitos (por exemplo, “www” infere “http://www”)

Exemplo de cadeia de caracteres	**Comportamento esperado/resultado**
www.exemplo.com	Um hyperlink é criado porque o protocolo está implícito[16].
rótulo@exemplo.com	Um hyperlink mailto: rótulo@exemplo.com é criado porque o protocolo está implícito.

3. Associe o ponto final ideográfico “。” (U+3002) e o caractere de ponto final árabe “-” (U+06D4) ao ponto final “.” (U+002E) (por exemplo, http://田中。com => http://田中.com), se as demais partes da cadeia de caracteres estiverem corretas.

4. Se os TLDs forem usados como um “sufixo especial” para determinar a possibilidade de criar links automaticamente (“linkificação”), então, todos os TLDs deverão ser incluídos. Uma lista de TLDs válidos deverá ser atualizada dinamicamente com frequência.

[16] Observação: talvez o site exija que os usuários finais digitem https:// em vez de http://. Se esse for o caso, é possível que o hyperlink não seja resolvido ou que seja exibida uma página de erro.

Parte 3: Tópicos avançados

Escritas complexas

É possível que os detalhes sobre as escritas complexas não interessem a desenvolvedores que não criam suas próprias bibliotecas de análise de cadeias de caracteres. Mesmo assim, este documento inclui um resumo para garantir que todos os leitores tenham conhecimento suficiente para reconhecer erros de códigos relacionados a essas escritas, se eles forem exibidos para os usuários.

Conformidade com Unicode e idiomas da direita para a esquerda

A maioria das escritas exibe os caracteres da esquerda para a direita quando o texto é apresentado em linhas horizontais. No entanto, também existem várias escritas, como o árabe e o hebraico, que exibem o texto na horizontal da direita para a esquerda. O texto também pode ser bidirecional (da esquerda para a direita – da direita para a esquerda) quando uma escrita da direita para a esquerda usa dígitos que geralmente são escritos da esquerda para a direita ou quando ela usa palavras incorporadas do inglês ou de outras escritas.

É possível que ocorram problemas ou ambiguidades quando a direção horizontal do texto não é uniforme. Para solucionar esse problema, existe um algoritmo que determina a direção em textos Unicode bidirecionais.

Existe um conjunto de regras que deve ser usado pelo aplicativo para produzir a ordem correta no momento da exibição; essas regras são descritas pelo **Algoritmo Unicode Bidirecional**. Ele é geralmente chamado de "**algoritmo Bidi**".

O algoritmo Bidi

O algoritmo Bidi descreve como os softwares devem processar textos que contêm sequências de caracteres LTR (left-to-right, esquerda para a direita) e RTL (right-to-left, direita para a esquerda). A **direção básica**[17] atribuída à frase determinará a ordem em que o texto será exibido.

Para saber se uma sequência é da esquerda para a direita ou da direita para a esquerda, cada caractere em Unicode tem uma propriedade direcional associada. A maioria das letras é predefinida (**caracteres fortes**) como LTR (esquerda para a direita). As letras de escritas da direita para a esquerda são predefinidas como RTL (direita para a esquerda). Uma sequência de caracteres RTL predefinidos será exibida da direta para a esquerda. Isso não depende da direção básica adjacente. Por exemplo:

(LTR) exemplo - مثال (RTL).

Textos com diferentes direções podem ser combinados em uma linha. Nesses casos, o algoritmo Bidi gera uma **execução direcional** separada de cada sequência de caracteres contínuos com a mesma direcionalidade.

Os espaços e as pontuações não são predefinidos como LTR ou RTL em Unicode porque são usados nesses dois tipos de escrita. Sendo assim, eles são classificados como **caracteres neutros** ou **fracos**. Os caracteres fracos são aqueles com direções vagas. Alguns exemplos desse tipo de caractere são:

- Dígitos europeus
- Dígitos indo-arábicos orientais
- Símbolos aritméticos e símbolos de moedas

[17] Em HTML, a direção básica é herdada da direção padrão do documento, que é da esquerda para a direita, ou definida explicitamente pelo elemento pai mais próximo que usar o atributo `dir`.

- Símbolos de pontuações comuns para muitas escritas, como os dois pontos, a vírgula, o ponto final e o espaço fixo.

Não é possível determinar a direção dos **caracteres neutros** sem um contexto. Alguns exemplos:

- Tabulações
- Separadores de parágrafos
- Outros caracteres whitespace

Quando um caractere neutro estiver entre dois caracteres predefinidos com o mesmo tipo direcional, ele também assumirá essa direção. Por exemplo, um caractere neutro entre dois caracteres RTL será tratado como um caractere RTL e terá o efeito de estender a execução direcional:

- نطاق.مثال

Mesmo se houver vários caracteres neutros entre os dois caracteres predefinidos, todos eles serão tratados da mesma forma.

Quando um espaço ou pontuação estiver entre dois caracteres predefinidos com diferentes direções, o caractere (ou caracteres) neutro será tratado como se tivesse a direção básica predominante. Por exemplo:

- exemplo. مثال

A menos que uma modificação direcional esteja presente, os **números** são sempre codificados (e inseridos) com a extremidade maior primeiro (big-endian[18]), e os numerais são considerados LTR. A direção mais fraca se aplica somente ao posicionamento do número por inteiro.

Para saber mais detalhes sobre o algoritmo Bidi, acesse: http://unicode.org/reports/tr9/tr9-11.html

A regra Bidi para nomes de domínio

Um **nome de domínio Bidi** contém pelo menos um rótulo RTL. Existe uma regra que determina as condições que devem ser atendidas para os rótulos em nomes de domínio Bidi. Essa regra pode ser encontrada na Seção 2 da RFC 5893: https://tools.ietf.org/html/rfc5893

Caracteres de ligação

Alguns idiomas usam escritas alfabéticas em que fonemas únicos são escritos usando dois caracteres; isso é chamado de **dígrafo**. Em outras palavras, um dígrafo é um grupo de duas letras sucessivas que representam um único som (ou **fonema**).

Exemplos de dígrafos em inglês

ch (como em *church* [igreja]) *th* (como em *then* [então]) *sh* (como em *shoe* [sapato])
ph (como em *phone* [telefone]) *th* (como em *think* [pensar])

[18] "Big-endian e little-endian são termos que descrevem a ordem em que uma sequência de bytes é armazenada na memória de um computador. Big-endian é uma ordem em que a extremidade maior, ou "big end", (o valor mais importante na sequência) é armazenado primeiro (no endereço de armazenamento mais inferior). Little-endian é uma ordem em que a extremidade menor, ou "little end", (o valor menos importante na sequência) é armazenado primeiro."
Fonte: http://searchnetworking.techtarget.com/definition/big-endian-and-little-endian

Alguns dígrafos são conectados plenamente como **ligaduras**. Na escrita e tipografia, uma ligadura ocorre quando dois ou mais grafemas ou letras são ligados como um único glifo. Um exemplo disso é o caractere do E comercial (&), que é uma evolução das letras latinas *e* e *t* ("et" significa "e") ligadas.

Se as ligaduras e os dígrafos tiverem a mesma interpretação em todos os idiomas que usarem uma determinada escrita, a normalização de Unicode geralmente resolve essas diferenças e faz uma correspondência. Quando eles tiverem interpretações diferentes, será necessário usar métodos alternativos para fazer a correspondência, provavelmente escolhidos no registro. Caso contrário, os usuários deverão ser informados que a correspondência não será realizada. Um exemplo de interpretação diferente pode ser encontrado na Seção 4.3 da RFC 5894: https://tools.ietf.org/html/rfc5894

O Unicode Consortium relaciona duas estratégias principais para determinar o comportamento de ligação de um determinado caractere após a aplicação do algoritmo Bidi:

- "Ao executar a moldagem, a implementação pode consultar o armazenamento original para confirmar se havia a presença de caracteres ZWNJ ou ZWJ[19] adjacentes.
- Além disso, a implementação também pode substituir ZWJ e ZWNJ por uma propriedade de caractere fora de banda ("out-of-band") associada aos caracteres adjacentes. Desse modo, as informações não interferirão no algoritmo Bidi e serão preservadas quando esses caracteres forem ordenados novamente. Depois que o algoritmo Bidi for aplicado, as informações fora de banda poderão ser usadas para executar a moldagem correta."[20]

Se os registros não derem atenção para a maneira que as cadeias de caracteres com possíveis interpretações diferentes de acordo com o IDNA de 2003 e a especificação atual são processadas, essas diferenças poderão ser usadas como um componente para ataques de correspondência de nomes e confusão de nomes (falsas similaridades). Sendo assim, esse cuidado é fundamental.

Para saber mais sobre ligações/ligadores, consulte a Seção 4.3 da RFC 5894: https://tools.ietf.org/html/rfc5894

Homoglifos e caracteres semelhantes

Os **homoglifos** são caracteres que, devido a semelhanças de tamanho e forma, podem parecer inicialmente idênticos.

Exemplos de homoglifos

Caractere cirílico **а**	=	Número Unicode 0430
Caractere latino **a**	=	Número Unicode 0061

[19] Para saber mais sobre ZWNJ/ZWJ, acesse: http://www.unicode.org/L2/L2005/05307-zwj-zwnj.pdf
[20] Fonte: Mark Davis, Aharon Lanin, Andrew Glass. 2015. *Unicode*. http://unicode.org/reports/tr9

Para evitar que nomes de domínio idênticos sejam registrados, os registros podem usar o procedimento de "agrupamento de homoglifos".[21]

O **agrupamento de homoglifos** é um processo em que, ao registrar um IDN, o sistema de registro automaticamente agrupa todos os homoglifos desse nome (se houver). Isso significa que vários nomes de domínio são agrupados ao mesmo tempo, e nenhum dos outros nomes de domínio desse grupo poderá ser registrado.

O agrupamento de homoglifos é uma prática recomendada para os registros evitarem possíveis práticas de phishing que têm a intenção de enganar os usuários com caracteres idênticos.

Para saber mais sobre os mecanismos de segurança de Unicode para a detecção de caracteres idênticos, acesse:

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

Para ver uma lista de homoglifos, acesse:

- http://homoglyphs.net

Para saber mais sobre caracteres idênticos e práticas recomendadas, consulte:

- M3AAWG Unicode Abuse Overview and Tutorial
  https://www.m3aawg.org/sites/default/files/m3aawg-unicode-tutorial-2016-02.pdf
- M3AAWG Best Practices for Unicode Abuse Prevention
  https://www.m3aawg.org/sites/default/files/m3aawg-unicode-best-practices-2016-02.pdf

Normalização e case folding

Normalização

A Normalização de Unicode ajuda a determinar se duas cadeias de caracteres Unicode quaisquer são equivalentes. Alguns caracteres podem ser representados em Unicode em várias sequências de códigos. Isso é chamado de **equivalência de Unicode**. O Unicode oferece dois tipos de equivalências:

- Canônica (NFD)
- Compatibilidade (NFK)

As sequências que representam o mesmo caractere são chamadas de **equivalente canônico.** Essas sequências têm a mesma aparência e significado quando impressas ou exibidas. Por exemplo:

Exemplos de caracteres que são equivalentes canônicos

U+006E ("n", latino minúsculo) seguido de U+0303 ("◌̃", til combinado)	=	**ñ**
U+00F1 (letra "ñ" minúscula do alfabeto espanhol)	=	**ñ**

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

Equivalentes compatíveis são sequências que têm aparências diferentes, mas em alguns contextos têm o mesmo significado. Esse é um tipo mais fraco de equivalência entre os caracteres e as sequências de caracteres.

Exemplos de caracteres que são equivalentes compatíveis

U+FB00 (a ligadura tipográfica “ff”)	=	**ff**
U+0066 U+0066 (duas letras “f” latinas)	=	**ff**

No exemplo acima, o ponto de código U+FB00 é definido como compatível, mas não é um equivalente canônico da sequência U+0066 U+0066. As sequências que são equivalentes canônicos também são compatíveis, mas o oposto não é necessariamente verdadeiro.

Para evitar problemas de interoperabilidade resultantes do uso de equivalentes canônicos, que, ainda assim, são sequências de caracteres diferentes, o W3C recomenda usar a Forma de Normalização C[22] para todo o conteúdo.

Para ver uma lista de todos os caracteres que podem sofrer alterações em qualquer Forma de Normalização, acesse: http://www.unicode.org/charts/normalization

Outras observações importantes:

- Apenas cadeias de caracteres NÃO transformadas por NFKC[23] são válidas.
- Quando dois aplicativos compartilham dados Unicode, mas os normalizam de maneiras diferentes, é possível que ocorram erros e a perda de dados.
- As Formas de Normalização devem permanecer estáveis com o tempo. Em outras palavras, uma cadeia de caracteres deve permanecer normalizada em todas as versões futuras de Unicode (compatibilidade reversa).

Dica para desenvolvedores de software

Não faça a normalização usando a conversão para maiúsculas, e não ignore os caracteres de espaço fixo, porque isso também pode atrapalhar os processos de classificação, cópia de dados, importação e exportação de dados, bem como a recuperação de dados por aplicativos clientes, o que talvez resulte na perda ou na corrupção dos dados.

Para saber mais sobre Formas de Normalização, acesse: http://www.unicode.org/reports/tr15

Case folding

Case folding é o processo de transformar dois textos, que diferem quanto à capitalização, mas que são “iguais”, em textos idênticos. O mapeamento de [a-z] para [A-Z] funciona na maioria dos

[22] NFC: decomposição canônica, seguida pela composição canônica.

[23] NFKC: decomposição por compatibilidade, seguida pela composição canônica.

documentos de texto simples somente em ASCII. No entanto, começa a ficar mais complicado em idiomas que também usam outros caracteres.

O Unicode define o mapeamento de capitalização padrão para cada ponto de código Unicode. Existem os **mapeamentos de capitalização comuns** e **completos**:

- Os **mapeamentos de capitalização comuns** têm um mapeamento simples e direto para um único ponto de código (geralmente em letra minúscula) correspondente.
- Os **mapeamentos de capitalização completos** normalmente exigem mais de um caractere Unicode.

Uma consideração importante, de acordo com o W3C,[24] é se os valores são restritos ao subconjunto ASCII de Unicode ou se o vocabulário permite o uso de caracteres (como acentos em letras latinas ou vários caracteres Unicode que incluem escritas não latinas) que possivelmente têm requisitos de capitalização mais complexos.[25]

Dica para desenvolvedores de software

▯	Considere usar a Normalização de Unicode além do processo de case folding.

Para saber mais sobre a Normalização de Unicode, consulte:

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

Para ver recomendações sobre o case folding, acesse:

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

[24] W3C: O World Wide Web Consortium (W3C) é uma comunidade internacional em que as organizações integrantes, uma equipe dedicada e o público trabalham juntos para desenvolver normas para a Web. Consulte: https://www.w3.org

[25] Fonte: A Phillips. 2015. *Character Model for the World Wide Web: String Matching and Searching*. https://www.w3.org/TR/charmod-norm

Parte 4: Glossário e outros recursos

Glossário

Rótulo A	A representação em ACE (ASCII Compatible Encoding, Codificação Compatível com ASCII) de um nome de domínio internacionalizado, ou seja, como ele é transmitido internamente no protocolo DNS. Os Rótulos A sempre começam com o prefixo "xn--". Compare com o Rótulo U.
Prefixo ACE	Prefixo de codificação compatível com ASCII.
Caracteres ASCII	American Standard Code for Information Interchange (Código Padrão Americano para o Intercâmbio de Informação). São os caracteres do alfabeto latino básico juntamente com os dígitos europeus/árabes. Eles também estão incluídos no conjunto mais amplo de "caracteres Unicode" que fornece a base para os IDNs.
API	A API (Application Programming Interface, Interface de Programação de Aplicativos) é um conjunto de rotinas, protocolos e ferramentas para a criação de softwares e aplicativos. Uma API pode ser destinada a um sistema baseado na Web, sistema operacional ou sistema de banco de dados. Ela oferece recursos para desenvolver aplicativos para esse sistema usando uma determinada linguagem de programação.
Espaço de código	Intervalo que define os limites de letras maiúsculas e minúsculas em uma codificação.
Pontos de código	Um ponto de código ou posição de código refere-se a qualquer valor numérico que forme o espaço de código. Eles são usados para distinguir a) o número de uma codificação como uma sequência de bits e b) o caractere abstrato de uma determinada representação gráfica (glifo).
Zona raiz do DNS	A zona raiz é o diretório central para o DNS, que é um componente chave na conversão de nomes de host legíveis em endereços de IP numéricos.
EAI	A EAI (Email Address Internationalization, Internacionalização de Endereços de E-mail) requer o uso do Unicode em todas as partes do endereço de e-mail.
IANA	Internet Assigned Numbers Authority (Autoridade para Atribuição de Números na Internet). As funções da IANA abrangem: • A manutenção do registro de parâmetros técnicos de protocolos da Internet • A administração de certas responsabilidades associadas à zona raiz do DNS da Internet • A alocação de recursos numéricos da Internet
ICANN	A ICANN (Internet Corporation for Assigned Names and Numbers, Corporação da Internet para Atribuição de Nomes e Números) é uma corporação sem fins lucrativos organizada internacionalmente, responsável pelas funções de alocação de espaço para endereços IP, pela atribuição de identificadores de protocolo, gerenciamento do sistema de nomes de domínio de primeiro nível com códigos

	de países (ccTLDs) e genéricos (gTLDs) e gerenciamento do sistema de servidores raiz.
IDN	Internationalized Domain Names (Nomes de Domínio Internacionalizados). Os IDNs são nomes de domínio que incluem caracteres usados na representação local de idiomas que não são escritos com as vinte e seis letras do alfabeto latino básico "a-z", os números 0-9 e o hífen "-".
IDNA	Internationalized Domain Names in Applications (Nomes de Domínio Internacionalizados em Aplicativos).
IDN ccTLD	Domínio de primeiro nível com código de país que inclui caracteres usados na representação local de idiomas que não são escritos com as vinte e seis letras do alfabeto latino básico "a-z". Exemplos: • .рф (Rússia) • .صر (Egito) • .السعودية (Arábia Saudita)
IETF	A IETF (Internet Engineering Task Force, Força-tarefa de Engenharia da Internet) é uma grande comunidade internacional aberta de designers de redes, operadores, fornecedores e pesquisadores preocupados com a evolução da arquitetura da Internet e a operação contínua da Internet. Ela é aberta a todos os interessados. A IETF desenvolve padrões para a Internet e, em particular, padrões relacionados com o Internet Protocol Suite (TCP/IP).
Idioma	O método de comunicação humana, falada ou escrita, que consiste no uso de palavras de maneira estruturada e convencional.
Punycode	É um algoritmo usado para representar o Unicode com o subconjunto de caracteres limitados do ASCII aceito pelo Sistema de Nomes de Domínio. O Punycode é destinado à codificação de rótulos na estrutura de IDNA (Internationalized Domain Names in Applications, Nomes de Domínio Internacionalizados em Aplicativos).
Registrador	Uma organização em que os nomes de domínio são registrados pelos usuários. O registrador mantém o registro das informações de contato e envia as informações técnicas a um diretório central conhecido como o "registro".
Registro	O banco de dados mestre e oficial de todos os nomes de domínio registrados em cada domínio de primeiro nível.
RFC	Uma RFC (Request for Comments, Solicitação de Comentários) é um documento formal da IETF (Internet Engineering Task Force, Força-tarefa de Engenharia da Internet) que foi escrito pelo comitê e posteriormente revisado por partes interessadas.
Escrita	O conjunto de letras ou caracteres usados para escrever e que representam os sons de um idioma.
Nome de domínio de segundo nível	Na hierarquia do DNS (Domain Name System, Sistema de Nomes de Domínio), o SLD (Second-Level Domain, Domínio de Segundo Nível), ou 2LD, é um domínio que

	está localizado diretamente abaixo de um TLD (Top-Level Domain, Domínio de Primeiro Nível). Por exemplo, em `exemplo.com`, `exemplo` é o domínio de segundo nível de `.com` TLD.
Rótulo U	Um "Rótulo U" é uma cadeia de caracteres IDNA válida formada por caracteres Unicode que inclui pelo menos um caractere ASCII. As conversões entre Rótulos U e Rótulos A são feitas de acordo com a especificação de Punycode [RFC3492].
Software habilitado para a UA ou pronto para a UA	Software habilitado para a Aceitação Universal. Refere-se a um software que tem a capacidade de Aceitar, Armazenar, Processar, Validar e Exibir todos os Domínios de Primeiro Nível igualmente e todos os IDNs, hyperlinks e endereços de e-mail igualmente.
Unicode	Um padrão universal para a codificação de caracteres. Ele define a maneira que os caracteres individuais são representados em arquivos de texto, páginas da Web e outros tipos de documentos. O Unicode foi criado para aceitar os caracteres de todos os idiomas do mundo. Ele aceita aproximadamente 1.000.000 de caracteres e até 4 bytes para cada caractere. Consulte: http://unicode.org
UTF	Unicode Transformation Format (Formato de Transformação de Unicode). Refere-se a uma forma de transformar os pontos de código Unicode em uma sequência de bytes. O UTF-8 é o formato mais indicado para processar endereços IDN e EAI. O UTF-8 converte Unicode em bytes de 8 bits.
M3AAWG	O Messaging, Malware and Mobile Anti-Abuse Working Group (M^3AAWG) é um grupo de participantes do setor que se reúne para combater botnets, malware, spam, vírus, ataques DoS e outros tipos de ameaças virtuais. Consulte: https://www.m3aawg.org/
W3C	O World Wide Web Consortium (W3C) é uma comunidade internacional em que as organizações integrantes, uma equipe dedicada e o público trabalham juntos para desenvolver normas para a Web. Consulte: https://www.w3.org/
ZWJ	O ZWJ (Zero-Width Joiner, Ligador de Largura Zero) é um caractere não impresso usado na definição tipográfica computadorizada de algumas escritas complexas, como a escrita árabe ou qualquer escrita índica. Ao ser colocado entre dois caracteres que não são conectados, o ZWJ faz com que sejam impressos nas suas formas conectadas.
ZWNJ	O ZWNJ (Zero-Width Non-Joiner, Não Ligador de Largura Zero) é um caractere não impresso usado em sistemas de escrita computadorizada que usam ligaduras. Ao ser colocado entre dois caracteres que não são conectados em uma ligadura, o ZWNJ faz com que sejam impressos nas suas formas final e inicial, respectivamente. Ele também tem o efeito de um caractere de espaço, mas o ZWNJ é usado quando for necessário manter as palavras juntas ou para conectar uma palavra ao seu morfema.

Para ver o glossário completo da ICANN, acesse: https://www.icann.org/resources/pages/glossary-2014-02-03-en

RFCs

RFCs DE PUNYCODE	
RFC 3492	**Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)** A RFC 3492 descreve o Punycode como: *"uma sintaxe de codificação de transferência simples e eficiente criada para ser usada com IDNA (Internationalized Domain Names in Applications, Nomes de Domínio Internacionalizados em Aplicativos)"* O Punycode transforma uma cadeia de caracteres Unicode de maneira única e reversível em uma cadeia de caracteres ASCII. Essa RFC define um algoritmo geral chamado **Bootstring**. Esse algoritmo permite que uma cadeia de caracteres de pontos de código básicos represente exclusivamente qualquer cadeia de caracteres de pontos de código retirada de um conjunto maior. https://tools.ietf.org/html/rfc3492
IDN RFCs	
RFC 5890	**Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework** Esta RFC descreve o contexto de uso e o protocolo para uma revisão de IDNA (Internationalized Domain Names for Applications, Nomes de Domínio Internacionalizados em Aplicativos). https://tools.ietf.org/html/rfc5890
RFC 5891	**Internationalized Domain Names in Applications (IDNA) Protocol** Esta RFC especifica o mecanismo de protocolo, chamada de IDNA (Internationalized Domain Names in Applications, Nomes de Domínio Internacionalizados em Aplicativos), para registrar e pesquisar IDNs de maneira que não seja necessário alterar o próprio DNS. https://tools.ietf.org/html/rfc5891
RFC 5892	**The Unicode Points and Internationalized Domain Names for Applications (IDNA)** A RFC 5892 especifica regras para decidir se um ponto de código, considerado de maneira isolada ou em um contexto, é um candidato para ser incluído em um IDN (Internationalized Domain Name, Nome de Domínio Internacionalizado). https://tools.ietf.org/html/rfc5892
RFC 5893	**Right-to-left scripts for Internationalized Domain Names for Applications (IDNA)** Esta RFC apresenta uma nova regra de Bidi para rótulos IDNA (Internationalized Domain Names in Applications, Nomes de Domínio Internacionalizados em Aplicativos) para ser usada em escritas da direita para a esquerda em IDNs (Internationalized Domain Names, Nomes de Domínio Internacionalizados).

	https://tools.ietf.org/html/rfc5893
RFC 5894	**Internationalized Domain Names for Applications (IDNA): Background, Explanation and Rationale** Este documento informativo apresenta uma visão geral de um sistema revisado criado para lidar com versões mais novas de Unicode e contém explicações dos seus componentes. https://tools.ietf.org/html/rfc5894
RFC 5895	**Mapping Characters for Internationalized Domain Names in Applications (IDNA) 2008** Esta RFC descreve as ações que podem ser tomadas em uma implementação entre o recebimento das entradas de usuários e a transferência dos pontos de código permitidos para o novo protocolo de IDNA (2008). Ela descreve uma operação que deve ser aplicada às entradas de usuários a fim de prepará-las para serem usadas em um protocolo "na rede". O documento também inclui um procedimento de implementação geral de mapeamento. https://tools.ietf.org/html/rfc5895
EAI RFCs	
RFC 6530	**Overview and Framework for Internationalized Email** Este padrão apresenta uma série de especificações que definem mecanismos e extensões de protocolo necessários para a compatibilidade total com os endereços de e-mail internacionalizados. Este documento descreve como os diversos elementos da internacionalização de e-mails estão interligados e as relações entre as especificações principais associadas ao transporte de mensagens, formatos de cabeçalhos e processamento. https://tools.ietf.org/html/rfc6530
RFC 6531	**SMTP Extension for Internationalized Email** O documento define uma extensão SMTP (Simple Mail Transfer Protocol, Protocolo de Transferência de Correio Simples) para que os servidores possam divulgar a capacidade de aceitar e processar endereços de e-mail internacionalizados e cabeçalhos de e-mail internacionalizados. https://tools.ietf.org/html/rfc6531
RFC 6532	**Internationalized Email Headers** Este documento especifica um aprimoramento no IMF (Internet Message Format, Formato de Mensagens da Internet) e no MIME que permite o uso de Unicode em endereços de e-mail e na maior parte do conteúdo de campos no cabeçalho. Este documento especifica um aprimoramento no IMF (Internet Message Format, Formato de Mensagens da Internet) (RFC 5322) e no MIME que permite o uso direto de UTF-8, em vez de apenas ASCII, em valores de campos de cabeçalhos, inclusive endereços de e-mail. Um novo tipo de mídia, mensagem/global, é definido para mensagens que usam esse formato estendido. Esta especificação também suspende a restrição do MIME de ter codificações de transferência de conteúdo sem identidade em qualquer subtipo do tipo de primeiro nível da mensagem para

	que as partes da mensagem/global sejam transmitidas com segurança usando as infraestruturas de e-mail existentes. https://tools.ietf.org/html/rfc6532
RFC 6533	**Internationalized Delivery Status and Disposition Notifications** Esta especificação acrescenta um novo tipo de endereço para endereços de e-mail internacionais, de modo que o endereço original do destinatário com caracteres não ASCII possa ser preservado corretamente mesmo após o downgrade. Ela também apresenta uma atualização dos tipos de mídias de retorno de conteúdo para notificações de status de entrega e notificações de disposição de mensagens para fins de compatibilidade com o uso do novo tipo de endereço. https://tools.ietf.org/html/rfc6533

Principais padrões

ISO 10646 (Unicode)	A fim de proporcionar uma base técnica comum para o processamento de informações eletrônicas em diversos idiomas, a ISO (International Organization for Standardization, Organização Internacional de Normalização) desenvolveu um padrão de codificação internacional chamado ISO 10646. O ISO 10646 corresponde a um padrão unificado para a codificação de caracteres em todos os idiomas mais usados no mundo, inclusive caracteres em chinês tradicional e simplificado. Esse enorme conjunto de caracteres é chamado de UCS (Universal Character Set, Conjunto Universal de Caracteres). O mesmo conjunto de caracteres é definido pelo padrão Unicode, que também define outras propriedades de caracteres e outros detalhes de aplicativos que devem interessar aos implementadores. O Unicode é um sistema de codificação de caracteres criado pelo Unicode Consortium para apoiar o intercâmbio, o processamento e a exibição de textos escritos dos idiomas mais usados no mundo. O ISO 10646 e o Unicode definem várias formas de codificação para seu repertório em comum: UTF-8, UCS-2, UTF-16, UCS-4 e UTF-32. http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=63182
GB18030 (China)	O GB 18030-2000 é um padrão do governo chinês que especifica uma página de códigos estendida para ser usada no mercado chinês, além do UTF-8. O código de processamento interno do repertório de caracteres pode e deve ser Unicode. No entanto, o padrão estabelece que os provedores de software devem garantir uma conversão completa do GB18030 e do código de processamento interno. Atualmente todos os desenvolvedores de produtos vendidos ou que serão vendidos na China devem planejar a migração da página de códigos para fornecer suporte ao GB18030, sem exceções. O GB18030 é um “padrão obrigatório” e o governo chinês controla o processo de certificação para reforçar a implantação do GB18030. http://icu-project.org/docs/papers/unicode-gb18030-faq.html
Unicode Technical Standard #46:	Esta especificação define um mapeamento consistente com os requisitos normativos do protocolo de IDNA de 2008 e compatível o máximo possível com o IDNA de 2003. No caso de softwares cliente, este documento apresenta o comportamento mais

Unicode IDNA Compatibility Processing	consistente com as expectativas dos usuários no que diz respeito ao processamento de nomes de domínio com os dados existentes. http://unicode.org/reports/tr46/

Recursos on-line

APIs	APIs para Windows https://www.msdn.microsoft.com/enus/library/windows/desktop/ff818516%28v=vs.85%29.aspx APIs para SharePoint https://msdn.microsoft.com/en-us/library/office/jj860569.aspx Lista de sufixos públicos https://publicsuffix.org/list/public_suffix_list.dat Lista oficial de TLDs da ICANN http://data.iana.org/TLD/tlds-alpha-by-domain.txt APIs para Android http://developer.android.com/guide/index.html APIs para MAC IOS https://developer.apple.com/library/mac/navigation .Net Framework https://msdn.microsoft.com/en-us/library/system.text.encoding(v=vs.110).aspx
Segurança do Unicode	Considerações sobre a segurança do Unicode http://www.unicode.org/reports/tr36 Mecanismos de segurança do Unicode http://www.unicode.org/reports/tr39
Agrupamentos de caracteres Unicode	Planos de código Unicode http://en.wikipedia.org/wiki/Mapping_of_Unicode_character_planes Visão geral do GB18030 http://en.wikipedia.org/wiki/GB_18030 Tabela oficial de mapeamento entre BG18030-2000 e Unicode http://source.icu-project.org/repos/icu/data/trunk/charset/data/xml/gb-18030-2000.xml Normalização de Unicode https://en.wikipedia.org/wiki/Unicode_equivalence
Vulnerabilidades do Unicode	Seção 3.1, “UTF-8 Exploits” (“Vulnerabilidades do Unicode”) no Relatório Técnico do Unicode #36 (em inglês) http://unicode.org/reports/tr36/#UTF-8_Exploit M3AAWG Best Practices for Unicode Abuse Prevention

	https://www.m3aawg.org/sites/default/files/m3aawg-unicode-best-practices-2016-02.pdf M3AAWG Unicode Abuse Overview and Tutorial https://www.m3aawg.org/sites/default/files/m3aawg-unicode-tutorial-2016-02.pdf Consulte também: http://www.unicode.org
Disposições gerais	URIs http://tools.ietf.org/html/rfc3986 The Domain Name System: A Non-Technical Explanation – Why Universal Resolvability Is Important http://www.internic.net/faqs/authoritative-dns.html Glossário da ICANN https://www.icann.org/resources/pages/glossary-2014-02-03-en

Agradecimentos

Os autores agradecem as pessoas a seguir pela valiosa contribuição e colaboração na elaboração deste 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

Alterações de versão

Da versão 8 para a versão 9

- Correção nas transformações de ponto sugeridas dos pontos Unicode.
- Remoção de um link irrelevante nas fontes oficiais.