



LANDING SMS SERVER HTTP GATEWAY

Versión 1.6

Esta API permite el acceso a todas las características de 360nrs usando Json.

ÚLTIMOS CAMBIOS

Versión 1.2 18/09/2017 Añadidas las funcionalidades para listar, actualizar y eliminar envíos programados

Corrección de errores

Versión 1.3 07/11/2017 Añadidos parámetros *campaignName* y *tags*

Versión 1.4 16/04/2018 Añadidos ejemplos Python, Java y C#

Versión 1.5 04/10/2018 Añadido parámetro para especificar variables de sustitución.

Versión 1.6 04/01/2019 Añadido parámetro *templated*.

ÍNDICE

INTRODUCCIÓN	Pág. 4
PLATAFORMA TÉCNICA	Pág. 4
Petición de envío de SMS	Pág. 5
Parámetros	Pág. 5
Ejemplo de petición básica	Pág. 7
Ejemplo de petición CURL	Pág. 7
Ejemplo de petición PHP	Pág. 7
Ejemplo de petición PYTHON	Pág. 8
Ejemplo de petición JAVA	Pág. 9
Ejemplo de petición C#	Pág. 11
Ejemplos de respuesta	Pág. 12
Gestión de mensajes programados	Pág. 13
Listado	Pág. 13
Parámetros	Pág. 13
Ejemplos de petición	Pág. 13
Ejemplos de respuesta	Pág. 13
Actualización	Pág. 14
Parámetros	Pág. 14
Ejemplos de petición	Pág. 14
Ejemplos de respuesta	Pág. 14
Borrado	Pág. 15
Parámetros	Pág. 15
Ejemplos de petición	Pág. 15
Ejemplos de respuesta	Pág. 15
Ejemplos de respuesta (errores)	Pág. 16
Anexo A – Acuse de recibo	Pág. 17
Anexo B – Conjunto de caracteres	Pág. 18

INTRODUCCIÓN

La plataforma http server permite al usuario enviar mensajes a través de esta plataforma. Para poder acceder a sus estadísticas y datos de facturación puede acceder a la web <https://www.360nrs.com> con sus datos de usuario.

La comunicación entre el cliente no se realizará a través de ningún API proporcionado por la Empresa, sino que simplemente se realizará una comunicación HTTP, con algunos parámetros a la URL indicada.

Este proceso se detalla a continuación.

PLATAFORMA TÉCNICA

Cada petición que se realice tendrá que incluir en la cabecera de la petición http la autenticación del cliente. Para ello se utiliza la autenticación de acceso básica de HTTP.

La cabecera de autorización se construye combinando la cadena “usuario: contraseña” y codificándola en base64. A esta cadena se antepone la cadena “Authorization: Basic”

Por ejemplo, para el usuario “miuser” y el password “mipass” la cabecera resultante sería:
Authorization: Basic bWl1c2VyOm1pcGFzcmw==

A continuación se detallará las opciones de envío disponibles, la URL a la que se debe llamar, y los parámetros que admite.

PETICIÓN ENVÍO DE LANDING SMS

URL: <https://dashboard.360nrs.com/api/rest/landingsms>

MÉTODO: **POST**

PARÁMETROS

Parámetro	Tipo	Obligatorio	Descripción
message	string	Sí	Texto del mensaje. Como máximo puede tener 160 caracteres si no se especifica que el mensaje sea multiparte (ver parámetro 'parts'). El texto tiene que estar codificado en UTF-8. Es necesario incluir el patrón "{LAND_URL}" en el cuerpo del mensaje. Este patrón será sustituido por el enlace a la landing page.
to	array	Sí	Número de teléfono móvil destinatario del mensaje. Debe incluir el prefijo (Ej: En España 34666666666). Este campo permite indicar multiples destinatarios.
from	string	Sí	Texto del Remitente, esta etiqueta se compondrá de 15 números o 11 caracteres alfanuméricos.
templateBody	string	No	Contenido de la landing page en formato HTML y codificación UTF-8. Requerido sin <i>templateId</i> .
templateId	integer	No	ID de plantilla para enviar como contenido de la landing page. Requerido sin <i>templateBody</i> .
encoding	string	No	Los posibles valores son "gsm", "gsm-pt" y "utf-16". El valor "gsm" para envíos normales con codificación GSM7 y 160 caracteres por mensaje y el valor "utf-16 para codificación UCS2 (UTF16) y 70 caracteres por mensaje. En caso de no especificarse, el valor por defecto es "gsm".

scheduleDate	string	No	Fecha de envío del mensaje en formato UTC. Si se necesita enviar mensajes programados se puede especificar la fecha de envío indicando la fecha en formato YYYYmmddHHiiSS (Ej: 20130215142000 sería el 15 de febrero de 2013 a las 14:20:00). En caso de envío inmediato no se tiene que especificar este parámetro.
parts	integer	No	Indica el número máximo de partes en que se dividirá el mensaje para su envío. Esta variable tiene valor 1 por defecto, por lo que si no se especifica y se envía un mensaje de más de 160 caracteres para codificación gsm, el mensaje fallará. Hay que tener en cuenta que los mensajes concatenados solo pueden tener 153 caracteres por parte en gsm y 67 caracteres por parte en utf-16 y que cada parte se tarifica como un envío. El servidor solo utilizará el mínimo de partes necesaria para realizar el envío del texto aunque el número de partes especificado sea superior al necesario. En caso de que el número de partes sea inferior al necesario para el envío del texto, el envío fallará con el error 105. El número máximo de partes permitido es de 8.
notificationUrl	string	No	URL en la que recibir las notificaciones de entrega.
trans	integer	No	Los valores posibles son 1 y 0. Con el valor 0 el servidor no modifica ningún carácter del mensaje, este es el valor por defecto. Con el valor 1 el servidor se encarga de modificar los caracteres comunes no válidos en GSM7 a caracteres válidos con la siguiente tabla de traducción: 'á' => 'a', 'í' => 'i', 'ó' => 'o', 'ú' => 'u', 'ç' => 'C', 'Á' => 'A', 'Í' => 'I', 'Ó' => 'O', 'Ú' => 'U', 'À' => 'A', 'È' => 'E', 'Ì' => 'I', 'Ò' => 'O', 'Û' => 'U', 'ø' => " ", 'æ' => " ", 'Ö' => 'O', 'ö' =>

			'o', 'á' => 'a', 'ê' => 'e', 'í' => 'i', 'ó' => 'o', 'ù' => 'u', 'À' => 'A', 'Ê' => 'E', 'Î' => 'I', 'Ô' => 'O', 'Ù' => 'U', 'ä' => 'a', 'Ä' => 'A'
campaignName	String	No	Nombre de campaña. Si se especifica se creará una campaña con el nombre indicado en el dashboard que contendrá las estadísticas del envío. Si una campaña con este nombre ya existe, las estadísticas de envío se añadirán a la campaña existente.
tags	array	No	<i>campaignName</i> es requerido si se especifica este parámetro. Listado de tags a añadir a la campaña. Los tags pueden ser utilizados para filtrar las estadísticas en el dashboard.
sub	array	No	array con variables de sustitución que se aplicarán al mensaje.

EJEMPLO DE PETICIÓN BÁSICA

```
{
  "to":["34666555444"],
  "message": "TEST API LANDING {LAND_URL}",
  "from": "LANDINGAPI",
  "templateBody":"<html><head><title>LANDING TEST</title></head><body><h1>HELLO LANDING</h1></body></html>"
}
```

Nota: Para generar una URL de baja hay que añadir: {UNSUB_URL} dentro del parámetro “message”.

VARIABLES DE SUSTITUCIÓN

Es posible indicar variables personalizadas en el cuerpo del mensaje y en el código HTML de la plantilla que serán sustituidas por las variables personalizadas del contacto o por las variables indicadas en el parámetro “sub”.

Al utilizar el parámetro sub, el array ha de contener tantos elementos como destinatarios del envío, utilizando el siguiente formato:

```
{
  "from": "TEST",
  "to": ["34666555444", "34666555333"],
  "message": "Hello {name}",
  "templateBody":"<html><head><title>LANDING TEST</title></head><body><h1>HELLO {name}</h1></body></html>",
}
```

```
"sub" : [  
    {"name": "first contact name"},// variables primer destinatario  
    {"name": "second contact name"}//variables segundo destinatario  
  ]  
}
```

EJEMPLO DE PETICIÓN CURL

```
curl -X POST \  
https://dashboard.360nrs.com/api/rest/landingsms \  
-H 'authorization: Basic bWl1c2VyOm1pcGFzcw==' \  
-H 'cache-control: no-cache' \  
-H 'content-type: application/json' \  
-d '{  
    "to": ["34666555444"],  
    "message": "TEST API LANDING {LAND_URL}",  
    "from": "LANDINGAPI",  
    "templateBody": "<html><head><title>LANDING TEST</title></head><body><h1>HELLO  
LANDING</h1></body></html>"  
}'
```


EJEMPLO DE PETICIÓN PHP

```
<?php

$post["to"] = array("34666555444");
$post["message"] = "Test Api Landing {LAND_URL}";
$post["from"] = "LANDINGAPI";
$post["templateBody"] = "<html><head><title>LANDING
TEST</title></head><body><h1>HOLA LANDING</h1></body></html>";
$user = "miuser";
$password = "mipass";

try {
    $ch = curl_init();

    curl_setopt($ch, CURLOPT_URL, "https://dashboard.360nrs.com/api/rest/landingsms");
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
    curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($post));
    curl_setopt($ch, CURLOPT_HTTPHEADER, array(
        "Accept: application/json",
        "Content-type: application/json",
        "Authorization: Basic " . base64_encode($user . ":" .
$password));
    $result = curl_exec($ch);
    var_dump($result);
} catch (Exception $exc) {
    echo $exc->getTraceAsString();
}
```

EJEMPLO DE PETICIÓN PYTHON

```
import base64
import json
import pycurl

if __name__ == "__main__":
    url = "https://dashboard.360nrs.com/api/rest/landingsms"
    usrPass = "miuser:mipass"

    data = json.dumps({
        "to":["34666555444"],
        "message":"Test Api Landing {LAND_URL}",
        "from":"landingAPI",
        "templateBody":"<html><head><title>LANDING
TEST</title></head><body><h1>HELLO LANDING</h1></body></html>",
        "campaignName":"Nombre Campaña",
```

```

        })

    b64Val = base64.b64encode(usrPass)
    headers = ["Accept:Application/json", "Authorization:Basic %s" % b64Val]

    c = pycurl.Curl()
    c.setopt(pycurl.URL, url)
    c.setopt(pycurl.HTTPHEADER, headers)
    c.setopt(pycurl.POST, 1)
    c.setopt(pycurl.POSTFIELDS, data)

    c.setopt(pycurl.SSL_VERIFYHOST, 0)
    c.setopt(pycurl.SSL_VERIFYPEER, 0)
    c.perform()

    http_code = c.getinfo(pycurl.HTTP_CODE)
    print(http_code)

```

EJEMPLO DE PETICIÓN JAVA

```

import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.URL;
import javax.net.ssl.HttpsURLConnection;
import org.json.simple.JSONArray;
import org.json.simple.JSONObject;

public class ApiLandingSms {

    public static void main(String args[]) throws IOException {

        String url = "https://dashboard.360nrs.com/api/rest/landingsms";
        URL obj = new URL(url);
        HttpsURLConnection con = (HttpsURLConnection) obj.openConnection();

        con.setRequestMethod("POST");
        String userpass = "miuser" + ":" + "mipass";
        String basicAuth = "Basic " +
javax.xml.bind.DatatypeConverter.printBase64Binary(userpass.getBytes("UTF-8"));
        con.setRequestProperty("Authorization", basicAuth);
        con.setRequestProperty("Accept", "application/json");

        JSONObject data = new JSONObject();
        JSONArray to = new JSONArray();

        to.add("34666555444");

```

```

        data.put("to", to);
        data.put("message", "Test Api Landing SMS {LAND_URL}");
        data.put("from", "landingAPI");
        data.put("templateBody", "<html><head><title>LANDING
TEST</title></head><body><h1>HELLO LANDING</h1></body></html>");
        data.put("campaignName", "Nombre Campaña");
        String jsonText = data.toJSONString();

        con.setDoOutput(true);
        try (DataOutputStream wr = new DataOutputStream(con.getOutputStream())) {
            wr.writeBytes(jsonText);
            wr.flush();
            wr.close();
        }

        BufferedReader in = new BufferedReader(new
InputStreamReader(con.getInputStream()));
        String inputLine;
        StringBuffer response = new StringBuffer();

        while ((inputLine = in.readLine()) != null) {
            response.append(inputLine);
        }
        in.close();

        System.out.println(response.toString());
    }
}

```

EXAMPLE OF C# REQUEST

```
using Newtonsoft.Json;
using System;
using System.IO;
using System.Net;

namespace nrs_api
{
    class Program
    {
        static void Main(string[] args)
        {

            var httpWebRequest =
(HttpWebRequest)WebRequest.Create("https://dashboard.360nrs.com/api/rest/landingsms");

            httpWebRequest.Method = "POST";
            httpWebRequest.Accept = "application/json";
            String username = "miuser";
            String password = "mipass";
            String encoded =
System.Convert.ToBase64String(System.Text.Encoding.GetEncoding("ISO-8859-1").GetBytes(username + ":" + password));
            httpWebRequest.Headers.Add("Authorization", "Basic " + encoded);

            using (var streamWriter = new
StreamWriter(httpWebRequest.GetRequestStream()))
            {
                string[] to = new string[1];
                to[0] = "34666555444";
                var message = "Test Api Landing {LAND_URL}";
                var from = "landingAPI";
                var templateBody = "<html><head><title>LANDING
TEST</title></head><body><h1>HELLO LANDING</h1></body></html>";
                var campaignName = "Nombre Campaña";

                var data = new
                {
                    to = to,
                    message = message,
                    from = from,
                    templateBody = templateBody,
                    campaignName = campaignName
                };
                string json = JsonConvert.SerializeObject(data);
```

```
        streamWriter.Write(json);
        streamWriter.Flush();
        streamWriter.Close();
    }

    var httpResponse = (HttpWebResponse)httpWebRequest.GetResponse();
    using (var streamReader = new
StreamReader(httpResponse.GetResponseStream()))
    {
        var result = streamReader.ReadToEnd();
        Console.WriteLine(result);
        Console.ReadKey();
    }
}
}
```

La clave de acceso (password) y el código del cliente (username) serán proporcionados por la empresa. Hay que comentar que con objeto de aumentar la seguridad del sistema, el cliente deberá indicar la IP desde donde se va a conectar, solo se permitirán envíos de la IP indicada por el cliente.

EJEMPLOS DE RESPUESTA

CÓDIGO ESTADO 202 (ACCEPTED):

```
[{"accepted":true,"to":"34666555444","id":"102648819"}]
```

CÓDIGO ESTADO 207 (MULTI-STATUS):

```
[{"accepted":true,"to":"34626690739","id":"102648820"},{"accepted":false,
"to":"34","error":{"code":102,"description":"No valid recipients"}}]
```

GESTIÓN DE MENSAJES PROGRAMADOS

LISTADO

Lista todos los envíos programados. Puede filtrarse por tipo (SMS o MAILING) y puede especificarse un GUID, varios GUID o ningún GUID (para listar todos).

El contenido del mensaje no se muestra en la lista, a no ser que se especifique un único GUID.

URL: <https://dashboard.360nrs.com/api/rest/scheduled>

MÉTODO: **GET**

PARÁMETROS

Parámetro	Tipo	Obligatorio	Descripción
guid	integer, array	No	Identificador del mensaje o mensajes. Se le puede pasar un identificador, un array de identificadores o ninguno para mostrarlos todos.
type	string	No	SMS o MAILING

EJEMPLOS DE PETICIÓN

<https://dashboard.360nrs.com/api/rest/scheduled>

<https://dashboard.360nrs.com/api/rest/scheduled?guid=100>

<https://dashboard.360nrs.com/api/rest/scheduled?guid=100,101,102>

<https://dashboard.360nrs.com/api/rest/scheduled?type=SMS>

<https://dashboard.360nrs.com/api/rest/scheduled?guid=100&type=SMS>

EJEMPLOS DE RESPUESTA

Con varios GUID o sin GUID:

```
{"result":[  
{"guid":"100","type":"SMS","created_at":"2017-09-18 10:49:12","updated_at":"2017-09-18 10:49:12","scheduled_at":"2017-11-11 10:10:10"},  
{"guid":"101","type":"SMS","created_at":"2017-09-18 10:49:12","updated_at":"2017-09-18 10:49:12","scheduled_at":"2017-11-11 10:10:10"},  
{"guid":"102","type":"SMS","created_at":"2017-09-18 10:49:12","updated_at":"2017-09-18 10:49:12","scheduled_at":"2017-11-11 10:10:10"}  
],"total":3}
```

Especificando un único GUID:

```
{"result":{"guid":"100","type":"SMS","content":"Contenido del SMS","created_at":"2017-09-18 10:49:12","updated_at":"2017-09-18 10:49:12","scheduled_at":"2017-11-11 10:10:10"},"total":1}
```

ACTUALIZACIÓN

Actualiza la fecha de programación un envío, varios envíos o todos los envíos programados.

Puede filtrarse por GUID y/o tipo.

CUIDADO: Si no se especifica ningún GUID se actualizarán todos los mensajes programados.

URL: <https://dashboard.360nrs.com/api/rest/scheduled>

MÉTODO: **PUT**

PARÁMETROS

Parámetro	Tipo	Obligatorio	Descripción
guid	integer, array	No	Identificador del mensaje o mensajes. Se le puede pasar un identificador, un array de identificadores o ninguno para mostrarlos todos.
type	string	No	SMS o MAILING
scheduleDate	string	Sí	Fecha de envío del mensaje en formato UTC. Si se necesita enviar mensajes programados se puede especificar la fecha de envío indicando la fecha en formato YYYYmmddHHiiss (Ej: 20130215142000 sería el 15 de febrero de 2013 a las 14:20:00). En caso de envío inmediato no se tiene que especificar este parámetro.

EJEMPLOS DE PETICIÓN

```
{"guid": 100, "scheduleDate": "20171011093000"}  
{"guid": [100,101], "scheduleDate": "20171011093000"}  
{"type": "SMS", "scheduleDate": "20171011093000"}  
{"scheduleDate": "20171011093000"}
```

EJEMPLOS DE RESPUESTA

```
{"result":true,"updated":1}  
{"result":true,"updated":2}
```

BORRADO

Elimina un envío, varios envíos o todos los envíos programados.

Puede filtrarse por GUID y/o tipo.

CUIDADO: Si no se especifica ningún GUID se borrarán todos los mensajes programados.

URL: <https://dashboard.360nrs.com/api/rest/scheduled>

MÉTODO: **DELETE**

PARÁMETROS

Parámetro	Tipo	Obligatorio	Descripción
guid	integer, array	No	Identificador del mensaje o mensajes. Se le puede pasar un identificador, un array de identificadores o ninguno para mostrarlos todos.
type	string	No	SMS o MAILING

EJEMPLOS DE PETICIÓN

```
{"guid": 100}
```

```
{"guid": [100,101]}
```

```
{"type": "SMS"}
```

EJEMPLOS DE RESPUESTA

```
{"result":true,"deleted":1}
```

```
{"result":true,"deleted":2}
```


EJEMPLOS DE RESPUESTA

CÓDIGO ESTADO 400 (BAD REQUEST):

```
{"error":{"code":102,"description":"No valid recipients"}}
{"error":{"code":104,"description":"Text message missing"}}
{"error":{"code":105,"description":"Text message too long"}}
{"error":{"code":106,"description":"Sender missing"}}
{"error":{"code":107,"description":"Sender too long"}}
{"error":{"code":108,"description":"No valid Datetime for send"}}
{"error":{"code":109,"description":"Notification URL incorrect"}}
{"error":{"code":110,"description":"Exceeded maximum parts allowed or incorrect number of parts"}}
{"error":{"code":113,"description":"Invalid coding"}}
{"error":{"code":120,"description":"Invalid GUID"}}
{"error":{"code":121,"description":"Invalid scheduled date"}}
```

CÓDIGO ESTADO 401 (UNAUTHORIZED):

```
{"error":{"code":103,"description":"Username or password unknown"}}
{"error":{"code":111,"description":"Not enough credits"}}
```

CÓDIGO ESTADO 402 (PAYMENT REQUIRED):

```
{"error":{"code":111,"description":"Not enough credits"}}
```

CÓDIGO ESTADO 500 (INTERNAL SERVER ERROR):

```
{"error":{"code":122,"description":"Update error"}}
{"error":{"code":123,"description":"Delete error"}}
```

ANEXO A: ACUSES DE RECIBO

Si se desean recibir los acuses de recibo en tiempo real se deberá especificar la variable **"notificationUrl"** con la URL del cliente donde quiere que se notifique es estado del envío.

El funcionamiento consiste en especificar en cada petición http la URL donde se desea que realice una petición de nuestro servidor cuando se reciba una notificación por parte de la operadora. Para ello el cliente debe disponer de un servidor http capaz de recibir esas notificaciones.

Nuestro servidor enviará las variables por el método GET tal como el cliente quiera, para ello en la URL que nos envía tiene que poner el nombre de la variable seguido de un carácter de escape que contendrá el valor, los caracteres de escape tienen la forma del carácter "%" seguido de una letra. Este sería un ejemplo de URL:

<http://mi.server.com/notifica.php?remitente=%p&tel=%P&estado=%d>

Estos son los caracteres de escape definidos:

%i Identificador de NRS que se entregó cuando se hizo el envío
%d valor del acuse de recibo
%p el remitente del SMS
%P el número de teléfono del receptor del mensaje SMS
%t fecha del envío del mensaje con formato "YYYY-MM-DD HH:MM", e.j., "2015-09-21 14:18"

El valor %d es el que nos devolverá el estado final del envío, los valores posibles son:

1: El mensaje ha sido entregado al destinatario.
2: El mensaje no se ha podido entregar al destinatario.
4: El mensaje ha sido entregado al SMSC, es una notificación intermedia, no un resultado final
16: No se ha podido entregar a la operadora final

Para explicar mejor el proceso, a continuación se da un ejemplo de cómo sería el envío de un sms y la recepción de su acuse de recibo.

En primer lugar enviamos el sms con la variable notificationUrl donde indicaremos la URL donde queremos recibir la notificación de entrega, añadiremos a esta URL nuestro identificador de envío para poder identificar inequívocamente cuando lo recibamos. La url final para la notificación sería:

<http://mi.server.com/notifica.php?idenvio=7584remitente=%p&tel=%P&estado=%d>

Por tanto la llamada final que deberíamos hacer para enviar el SMS sería:

```
curl -X POST -H "Content-Type: application/json" -H "Accept: application/json" -H "Authorization: Basic bWl1c2VyOm1pcGFzcw=="  
-d "{ \"to\": \"34666555444\", \"message\": \"mensaje de  
texto\", \"from\": \"TEST\", \"notificationUrl\": \"http://mi.server.com/notifica.php?idenvio=7584remitente=%p&tel=%P&estado=  
%d\" }" https://dashboard.360nrs.com/api/rest/sms
```

Suponiendo que todos los mensajes puedan ser entregados, recibiremos al script notifica.php tres peticiones con el estado=1, remitente=TEST, idenvio=7584 y el número de teléfono correspondiente.

ANEXO B: CONJUNTO DE CARACTERES GSM7

CÓDIGO DE CARACTERES BÁSICO

	0x00	0x10	0x20	0x30	0x40	0x50	0x60	0x70
0x00	@	Δ	SP	0	i	P	ı	p
0x01	£	_	!	1	A	Q	a	q
0x02	\$	Φ	"	2	B	R	b	r
0x03	¥	Γ	#	3	C	S	c	s
0x04	è	Λ	α	4	D	T	d	t
0x05	é	Ω	%	5	E	U	e	u
0x06	ù	Π	&	6	F	V	f	v
0x07	î	Ψ	'	7	G	W	g	w
0x08	ò	Σ	(8	H	X	h	x
0x09	ç	Θ)	9	I	Y	i	y
0x0A	LF	Ξ	*	:	J	Z	j	z
0x0B	Ø	ESC	+	;	K	Ä	k	ä
0x0C	ø	Æ	,	<	L	Ö	l	ö
0x0D	CR	æ	-	=	M	Ñ	m	ñ
0x0E	Å	β	.	>	N	Ü	n	ü
0x0F	å	É	/	?	O	§	o	à

EXTENSIÓN DEL CONJUNTO DE CARACTERES BÁSICO
ESTOS CARACTERES OCUPAN DOS POSICIONES

	0x00	0x10	0x20	0x30	0x40	0x50	0x60	0x70
0x00								
0x01								
0x02								
0x03								
0x04		^						
0x05							€	
0x06								
0x07								
0x08			{					
0x09			}					
0x0A	FF							
0x0B		SS2						
0x0C				[
0x0D	CR2			~				
0x0E]				
0x0F			\					