

Examen Técnico - MercadoLibre S.R.L

Como todo lo “retro” últimamente está de moda, en MELI nos propusimos utilizar el telégrafo para comunicaciones entre distintos equipos utilizando código MORSE. En este contexto nuestro equipo se propone diseñar una solución que permita identificar los mensajes de cada uno de los emisores dentro de la empresa.

Un problema implícito para la interpretación de los mensajes en MORSE, es la velocidad de transmisión de los mismos. Naturalmente la velocidad de transmisión del código varía ligeramente a lo largo del mensaje ya que, es enviada por un operador humano.

Veamos un ejemplo, el texto “HOLA MELI” en MORSE sería el siguiente:

.... --- .-.. .- -- . .-.. ..

Una transmisión en bits podría representarse de la siguiente forma (considerando el bit 1 como un pulso y el 0 como pausa):

```
0000000011011011001110000011111100011111100111111000000011101111111101110
1110000000110001111110000011111100111111000000011000011011111111011101110
000001101111000000000000
```

Como se puede ver, esta transmisión es generalmente precisa de acuerdo con el estándar, pero algunos puntos, trazos y pausas son un poco más cortos o un poco más largos que los demás en el mismo mensaje.

Tener en cuenta que no necesariamente la frecuencia de envío es constante (dado que es un operador humano) pero sí se considera que la representación es consistente durante todo el mensaje. Es decir, si un operador es lento o rápido, seguirá siendo lento o rápido a lo largo del envío de ese mensaje. Ejemplificando, si los puntos para cierto operador son de largo menor o igual a 5 bits, no pasará durante el envío que desee representar un trazo de largo 4 o 5. El mismo concepto aplica a las pausas para la interpretación de la separación entre letras, palabras y finalización del mensaje.

Se entiende aquí que una pausa prolongada o el ingreso de un “full stop” (.-.-.) indica el fin del mensaje. No es requisito soportar ambos métodos de finalización.

Se pide implementar en cualquier lenguaje de programación :

1. Una función `decodeBits2Morse` que dada una secuencia de bits, retorne un string con el resultado en MORSE.
2. Una función `translate2Human` que tome el string en MORSE y retorne un string legible por un humano. Se puede utilizar la tabla debajo como referencia.

Nota:

- Si desea utilizar un repositorio, de preferencia que sea privado (ej: bitbucket, gitlab)
- Se deja a libre elección, el modo de transformar un input físico a bits.

Bonus:

1. Diseñar una API que permita traducir texto de MORSE a lenguaje humano y viceversa.
2. Hostear la API en un cloud público (como app engine o cloud foundry) y enviar la URL para consulta

Ejemplo:

```
$ curl -X POST "http://meli.com/translate/2text" -d "{text: '.... --- .-. .-   -- .-. ..}'"  
{ code:200, response: 'HOLA MELI'}
```

```
$ curl -X POST "http://meli.com/translate/2morse" -d "{text: 'HOLA MELI}'"  
{ code:200, response: '.... --- .-. .-   -- .-. ..}'
```

TABLA DE TRADUCCIÓN

Caracter	Código MORSE	Caracter	Código MORSE
A	.-	B	-...
C	-.-.	D	-..
E	.	F	..-
G	--.	H
I	..	J	.---
K	-.-	L	.-..
M	--	N	-.
O	---	P	.-.-
Q	--.-	R	.-.
S	...	T	-
U	..-	V	...-
W	.-.-	X	-.-
Y	-.--	Z	--..
0	-----	1	.----
2	..----	3	...--
4-	5
6	-....	7	--...
8	---..	9	----.
Fullstop (period)	.-.-.-		