

Trabajo Fin de Grado

Grado en Ingeniería Informática

**Convertor de cifras numéricas a
texto en gallego**

Autor:

Juan Peraza Machín

Tutor:

Francisco Javier Carreras Riudavets

Contenido

Índice de tablas.....	5
Índice de ilustraciones.....	6
Resumen.....	7
Abstract.....	8
1. ESTADO ACTUAL Y OBJETIVOS INICIALES	9
1.1 Conceptos generales sobre la conversión de números a texto en gallego.	9
1.1.1 Números cardinales.....	9
1.1.2 Números Ordinales	11
1.1.3 Números fraccionales.....	14
1.1.4 Números Multiplicativos.....	15
1.1.5 Números decimales.	15
1.2 Objetivos a cubrir en este trabajo.....	17
2. JUSTIFICACIÓN DE LAS COMPETENCIAS ESPECÍFICAS CUBIERTAS.	17
2.1 Trabajo fin de grado	17
2.2 Comunes a la ingeniería informática.....	18
2.3 Tecnologías de la información.....	19
3. APORTACIONES	19
4. DISEÑO.....	20
4.1 ASP.NET.....	21
4.1.1 Respuesta	23
4.1.2 Error	25
4.2 WCF.....	25
4.2.1 Entrada	26
4.2.2 Salidas	27

4.2.3 Panorama general del proceso de conversión.	27
4.2.4 Estructura de la clase Conversión	30
5. DESARROLLO	38
5.1 Expresiones regulares.....	38
5.2 Programación orientada a objetos.	40
5.3 Concurrencia.	40
5.4 División y concatenación	41
5.5 Optimización del rendimiento mediante el empleo de la clase StringBuilder	43
5.6 Internacionalización del servicio.....	43
6. CONCLUSIONES Y TRABAJOS FUTUROS	49
8. FUENTES DE INFORMACIÓN.....	50

Índice de tablas

Tabla 1 Números cardinales unidades.....	9
Tabla 2 Números cardinales decenas	9
Tabla 3 Números cardinales decenas-2.....	10
Tabla 4 Números cardinales centenas	10
Tabla 5 Números cardinales millares.....	10
Tabla 6 Números cardinales superiores de millar	11
Tabla 7 Números ordinales unidades y decenas.....	11
Tabla 8 Números ordinales decenas.....	12
Tabla 9 Números ordinales decenas-2	12
Tabla 10 Números ordinales centenas.....	12
Tabla 11 Números ordinales millar y superiores	13
Tabla 12 Números fraccionarios	14
Tabla 13 Números multiplicativos	15
Tabla 14 Números decimales	16
Tabla 15 Números decimales-2.....	16
Tabla 16 Expresiones regulares - Cardinales con separadores	39
Tabla 17 Expresiones regulares - Formato plano.....	39
Tabla 18 Expresiones regulares - Números fraccionarios.....	39
Tabla 19 Expresiones regulares - Números decimales.....	40
Tabla 20 Expresiones regulares - Tamaño máximo del número	40
Tabla 21 División y concatenación.....	43

Índice de ilustraciones

Ilustración 1 Front - vista global	21
Ilustración 2 Front - Información y Ejemplos.....	22
Ilustración 3 Front - Idiomas.....	22
Ilustración 4 Front - Versión móvil.....	23
Ilustración 5 Front - Tarjetas.....	23
Ilustración 6 Front - Tarjeta desplegada.....	24
Ilustración 7 Front - Pop-up Notas.....	24
Ilustración 8 Front - Pop-up Referencias	25
Ilustración 9 Front - Tarjeta de error.....	25
Ilustración 10 Back - Diagrama de flujo	28
Ilustración 11 Back - Estructura lista conversión-1	35
Ilustración 12 Back - Estructura lista conversión-2	36
Ilustración 13 Back - Estructura lista conversión-3	37
Ilustración 14 WCF- Recursos-Español.....	45
Ilustración 15 WCF- Recursos-Inglés.....	46
Ilustración 16 WCF- Recursos-Gallego	47
Ilustración 17 ASP.NET- Recursos-Español	48
Ilustración 18 ASP.NET - Recursos-Inglés	48
Ilustración 19 ASP.NET - Recursos-Gallego.....	48

Resumen

El propósito de este proyecto de final de grado es la creación de un servicio y una aplicación web destinados a la conversión de cifras numéricas a su representación textual en gallego. La aplicación proporciona una variedad de características, que abarcan desde la conversión de números a letras en diferentes contextos, como números cardinales, ordinales, decimales, multiplicativos, fraccionarios o partitivos, hasta la presentación de información detallada sobre la morfología, ortografía y gramática de cada número convertido a letras. Además, se incorporan ejemplos que facilitan la comprensión y el uso adecuado de estos números en su forma textual.

Abstract

The purpose of this final degree project is the creation of a service and a web application aimed at converting numbers into their Galician textual representation. The application provides a variety of features, ranging from converting numbers to letters in different contexts, such as cardinal numbers, ordinal numbers, decimals, multiplicatives, fractionals, or partitives, to the presentation of detailed information on the morphology, spelling, and grammar of each number converted into letters. In addition, examples are included to facilitate the understanding and proper use of these numbers in their textual form.

1. ESTADO ACTUAL Y OBJETIVOS INICIALES

1.1 Conceptos generales sobre la conversión de números a texto en gallego.

Al igual que sucede en el idioma español, la conversión de números a texto en gallego sigue un proceso sistemático, aunque con algunas excepciones, que puede conducir a resultados alternativos. Por lo tanto, un número puede expresarse en su forma cardinal, ordinal, fraccionaria, entre otras posibilidades. A continuación, exploraremos las formas más comunes de representar un número en gallego.

1.1.1 Números cardinales

Los números cardinales tienen la función de expresar cantidades de manera numérica. Cuando nos referimos a los números del 0 al 19 en gallego, utilizamos una convención específica que dicta su escritura de la siguiente manera:

0	cero	10	dez
1	un	11	once
2	dous	12	doce
3	tres	13	trece
4	catro	14	catorce
5	cinco	15	quince
6	seis	16	dezaseis
7	sete	17	dezasete
8	oito	18	dezaoito
9	nove	19	dezanove

Tabla 1 Números cardinales unidades

Como se puede observar, a partir del número 16, la forma de escribirlo implica usar la palabra "dez", seguida de "a", y luego se mencionan las unidades.

A partir del número 20 en adelante, la forma de escribir las decenas sigue un patrón establecido que se detalla en la siguiente tabla:

20	vinte	60	sesenta
30	trinta	70	setenta
40	corenta	80	oitenta
50	cincuenta	90	noventa

Tabla 2 Números cardinales decenas

Una vez que alcanzamos el setenta, las decenas se expresan de una manera que guarda similitud con la forma en que se escriben las unidades. Sin embargo, se les agrega el sufijo "-enta"

para conformar el número completo. Este patrón continúa más allá del setenta y se aplica consistentemente en los números en gallego.

Si el número se encuentra en el rango de 21 a 99, y el dígito de las unidades no es cero, se escribe comúnmente como dos palabras separadas por la letra "e" (decenas "e" unidades):

21	vinte e un	28	vinte e oito
34	trinta e catro	52	cincuenta e dous
69	sesenta e nove	79	setenta e nove
85	oitenta e cinco	99	noventa e nove

Tabla 3 Números cardinales decenas-2

Las centenas, en su mayoría, siguen un patrón regular en gallego, con una excepción notoria: el número 100 se convierte en "cen," que equivale a "cien" en español (100=cen, 101=cento un). A partir del 200, podemos notar que las centenas se escriben de manera similar a como se expresan las unidades, pero se les añade el sufijo "centos." Esta transición gradual y la preservación de "cen" para 100 nos permite apreciar cómo se estructuran y forman los números en el idioma gallego, como se puede ver en la siguiente tabla:

100	cen	101	cento un
200	douscentos	252	douscentos cincuenta e dous
300	trescentos	378	trescentos setenta e oito
400	catrocentos	466	catrocentos sesenta e sies
...
900	novecientos	999	novecientos noventa e nove

Tabla 4 Números cardinales centenas

Este mismo concepto se extiende de manera similar a los millares, en los cuales se coloca el número de miles antes de la palabra "mil". Es importante destacar que el número 1,000 se puede expresar de dos formas válidas: como "un mil" o simplemente como "mil":

1.000	mil	10.000	dez mil
2.000	dous mil	25.000	vinte e cinco mil
8.000	oito mil	81.000	oitenta e un mil
9.000	nove mil	99.000	noventa e nove mil

Tabla 5 Números cardinales millares

Cuando nos referimos a números más allá del millón en gallego, seguimos la convención de la "escala larga" (también conocida como "escala grande") para expresar cifras elevadas. En este sistema, los nombres de los números experimentan cambios significativos después de cada millón (10⁶). Esta distinción se hace evidente en la tabla que se muestra a continuación:

1.000.000	un millón
1.000.000.000	un mil millóns
1.000.000.000.000	un billón

1.000.000.000.000.000	un mil billóns
-----------------------	----------------

Tabla 6 Números cardinales superiores de millar

Cuando nos referimos a números que superan un billón y siguen la progresión en potencias ascendentes de diez, se estructuran de la siguiente manera: "trillón" para 10^{12} , "mil trillones" para 10^{15} , "cuatrillón" para 10^{24} , y "mil cuatrillones" para 10^{27} , que se representa como un "uno" seguido de 27 ceros.

Con frecuencia, en el idioma gallego, se tiende a usar espacios (también se puede usar puntos) en lugar de comas para delimitar los miles en números grandes (y ocasionalmente se emplean espacios comunes o apóstrofes). Esto se hace con la finalidad de evitar posibles confusiones en países donde se usa la coma como separador decimal. En consecuencia, un millón se representa como 1 000 000. En ciertas áreas, se podría emplear un punto como separador de miles, pero en ese contexto, es fundamental que la coma sea el símbolo utilizado para separar los decimales.

1.1.2 Números ordinales

Los números ordinales tienen relación con la posición que ocupan en una secuencia o serie. Su forma de escritura guarda similitud con la de los números cardinales, pero presenta algunas diferencias notables.

Desde el número 0 hasta el 19, la escritura se guía de acuerdo con la tabla que se presenta a continuación:

0º		10º	décimo
1º	primeiro	11º	undécimo
2º	segundo	12º	duodécimo
3º	terceiro	13º	décimo terceiro
4º	cuarto	14º	décimo cuarto
5º	quinto	15º	décimo quinto
6º	sexto	16º	décimo sexto
7º	sétimo	17º	décimo sétimo
8º	oitavo	18º	décimo oitavo
9º	noveno	19º	décimo noveno

Tabla 7 Números ordinales unidades y decenas

Hasta este punto, hemos observado que, salvo en los casos de 1º, 2º, 3º y 5º, las unidades en números ordinales guardan una semejanza con sus equivalentes en números cardinales. Sin embargo, es importante destacar que en los casos de 11º y 12º, también es válido el uso de "décimo primero" y "décimo segundo" como alternativas para expresar la misma posición ordinal. Esta peculiaridad añade una flexibilidad adicional a la forma en que se pueden expresar los números ordinales en gallego.

Desde el número 20 en adelante, la escritura de las decenas sigue el patrón indicado en la siguiente tabla:

20º	vixésimo	60º	sesaxésimo
30º	trixésimo	70º	septuaxésimo
40º	cuadraxésimo	80º	octoxésimo
50º	quincuaxésimo	90º	nonaxésimo

Tabla 8 Números ordinales decenas

En lo que respecta a las decenas en gallego, podemos notar que se forman utilizando la terminación "-xésimo", que es el equivalente de la terminación "-ésimo" en español.

Los números ordinales, como "24º", "53º" y similares, se generan en gallego al combinar una decena en su forma ordinal con una unidad en su forma ordinal.

21º	vixésimo primeiro	66º	sesaxésimo sexto
29º	vixésimo noveno	77º	septuaxésimo sétimo
35º	trixésimo quinto	81º	octoxésimo primeiro
48º	cuadraxésimo oitavo	99º	nonaxésimo noveno

Tabla 9 Números ordinales decenas-2

Las centenas se representan según lo ilustra la tabla a continuación:

100º	centésimo	600º	sexcentésimo
200º	ducentésimo	700º	septinxentésimo
300º	tricentésimo	800º	octinxentésimo
400º	cuadrinxentésimo	900º	noninxentésimo
500º	quinxentésimo	""	""

Tabla 10 Números ordinales centenas

En lo que respecta a las centenas en gallego, nos encontramos con dos posibles terminaciones para los números ordinales. Por un lado, está la terminación "-centésimo", que se utiliza en casos como 200º, 300º, 600º y también para el 100º, que se traduce literalmente como "centésimo".

Por otro lado, tenemos la terminación "-xentésimo", que se emplea en situaciones como 400º, 500º, 700º, 800º, 900º. Esta terminación agrega una peculiaridad a la forma en que se expresan los números ordinales en gallego, proporcionando un matiz adicional en la construcción de los ordinales en función de la centena a la que se hace referencia.

En cuanto a las unidades de millares y números superiores, su escritura sigue el patrón que se muestra en la siguiente tabla:

1000º	milésimo
10,000º	dezmilésimo
100,000º	cenmilésimo
1.000.000º	millonésimo

...	...
1.000.000.000.000	billonésimo
...	...
1.000.000.000.000.000.000	trillonésimo

Tabla 11 Números ordinales millar y superiores

Los números ordinales en gallego tienen un patrón consistente y efectivo para expresar la posición relativa en una serie o secuencia. Cuando nos referimos a unidades de millar y números superiores, el patrón de formación incluye la terminación "-ésimo", que se agrega a los números cardinales correspondientes. Empezando con "milésimo," este es el ordinal para el número 1,000, donde "mil" es "mil" y "ésimo" indica la posición. Luego, para "dezmilésimo," se agrega "dezmil" a la estructura y, de manera similar, "cenmilésimo" para el número 100,000, donde "cenmil" es "cien mil."

A medida que avanzamos en la escala numérica, el patrón persiste. Por ejemplo, cuando llegamos a "millonésimo," estamos expresando la posición ordinal del número 1,000,000. Aquí, "millón" es "millón" y nuevamente "ésimo" enfatiza la posición. El patrón se repite con números aún más grandes, como "billonésimo" para 1,000,000,000,000, y así sucesivamente con "trillonésimo" para 1,000,000,000,000,000,000. La terminación "-ésimo" es una característica clave para formar números ordinales en gallego que representan unidades de millar y números superiores.

Los números ordinales más altos generalmente no se expresan en palabras, a menos que sean números "redondos" como "milésimo," "millonésimo," "billonésimo," y así sucesivamente. En su lugar, se representan utilizando una combinación de dígitos y letras, y aquí están algunas reglas importantes para tener en cuenta al hacerlo:

- **Uso de Dígitos:** Los números ordinales superiores, como aquellos en la escala de millones, billones, trillones y más allá, se representan principalmente con dígitos numéricos. Por ejemplo, "1.000.000^o" en lugar de "un millonésimo".
- **Uso de Símbolo Ordinal (°):** En la mayoría de los casos, se coloca el símbolo "°" al final del número para indicar su carácter ordinal. Esto facilita la lectura y la comprensión. Por ejemplo, "1.000.000^o" o "1.000^o".
- **Separación con Puntos:** Para números grandes, es común utilizar puntos como separadores de miles, millones, billones, etc. Esto ayuda a mantener la legibilidad. Por ejemplo, "1.000.000^o" en lugar de "1000000^o".

1.1.3 Números fraccionales

Los números fraccionarios en gallego, al igual que en español, constituyen una herramienta para expresar partes fraccionadas de un número o una cantidad. Estas fracciones están compuestas por dos elementos fundamentales: el numerador y el denominador, los cuales se encuentran separados por una línea horizontal, comúnmente representada por el símbolo "/" o también conocido como "barra inclinada":

- **Numerador:** El numerador, situado sobre la línea horizontal, representa la cantidad de partes que se están considerando o contando en la fracción.
- **Denominador:** El denominador, ubicado debajo de la línea horizontal, indica en cuántas partes iguales se divide el número completo o la unidad.

En lo que respecta al numerador, su representación es bastante directa, ya que simplemente se escribe como un número cardinal, sin mayores complicaciones. Este número cardinal indica la cantidad de partes consideradas o contadas en la fracción. En cambio, el denominador sigue un patrón específico que depende de la fracción que estamos representando. Este patrón es bastante predecible y sigue una estructura uniforme, como se muestra en la siguiente tabla:

1/2	un medio	1/11	un onceavo
1/3	un terzo	1/12	un doceavo
1/4	un cuarto	1/29	un vinte e noveavo
1/5	un quinto	1/45	un corenta e cincoavo
1/6	un sexto	1/100	un centésimo
1/7	un sétimo	1/654	un seiscentos cincuenta e catroavo
1/8	un oitavo	1/840	un oitocentos corentavo
1/9	un noveno	1/1000	un milésimo
1/10	un décimo	1/1000000	un millonésimo

Tabla 12 Números fraccionarios

A partir de 1/11 (con excepción de 1/100, 1/1000, 1/1.000.000...), la formación de fracciones partitivas sigue una regla constante: se agrega el sufijo "-avo" al número cardinal correspondiente. De esta manera, podemos expresar las fracciones de la siguiente manera:

- (1/16): un dezaseisavo
- (2/29): dous vinte e noveavos
- (7/345): sete trescentoscorenta e cincoavos

Esta estructura sistemática permite representar de manera clara y consistente las fracciones partitivas en gallego, utilizando el sufijo "-avo" para indicar la división en partes iguales de un número o una cantidad.

1.1.4 Números multiplicativos

Los números multiplicativos son aquellos que expresan la cantidad de repeticiones de una acción o una cantidad específica. En gallego, estos números se pueden representar de la siguiente manera:

2	duplo	21	vinte e un veces
3	triplo	32	trinta e dous veces
4	cuádruplo	45	corenta e cinco veces
5	quíntuplo	59	cincuenta e nove veces
6	séxtuplo	64	sesenta e catro veces
7	séptuplo	78	setenta e oito veces
8	óctuplo	88	oitenta e oito veces
9	nónuplo	99	noventa e nove veces
10	décuplo	18	dezaoito veces
100	céntuplo	123	cento vinte e tres veces

Tabla 13 Números multiplicativos

Es evidente que los números multiplicativos en gallego siguen una formación específica y, a partir del número 11 (con la excepción del 100), se construyen mediante una combinación del número cardinal seguido de la palabra "veces". Esta regla proporciona una manera clara y consistente de expresar la multiplicación de cantidades o repeticiones de una acción.

1.1.5 Números decimales.

Los números decimales son una notación que permite representar números fraccionarios de manera más concisa sin la necesidad de utilizar una fracción con numerador y denominador explícitos. Por ejemplo, la fracción "5/10" puede expresarse como el decimal "0,5", que resulta de dividir 5 entre 10. En la notación decimal, la coma (,) se utiliza como separador entre la parte entera y la parte fraccionaria del número. Si el decimal es menor que 1, se coloca un cero antes de la coma decimal, como en "0,5" en lugar de simplemente ",5".

En el contexto textual, el número decimal "0,5" puede expresarse de dos maneras en gallego, similar a cómo se hace en español. Puede describirse como "cinco décimas" o como "cero con cinco décimas".

Cuando el número decimal es mayor que 1, es posible nombrar tanto la parte entera como la parte decimal utilizando la palabra "con" como conector. Por ejemplo, el número "2,5" puede representarse como "dous con cinco décimas".

Si la parte decimal de un número tiene dos dígitos en gallego, en lugar de usar la palabra "décimas", empleamos "centésimas". Por ejemplo, el número "0,29" se puede expresar como "vinte e nove centésimas" o como "cero con vinte e nove centésimas". Del mismo modo, si el

número decimal es mayor que 1, se puede nombrar la parte entera y la parte decimal unidas mediante la palabra "con".

Cuando la parte decimal de un número en gallego consta de tres dígitos, empleamos el término "milésimas", como se puede observar en los siguientes casos:

0,897	cero con oitocentas noventa e sete milésimas
231,003	douscentos trinta e un con tres milésimas

Tabla 14 Números decimales

En líneas generales, la forma de especificar la parte decimal se determina según la posición que ocupe el último dígito decimal, siguiendo el patrón indicado en la siguiente tabla:

POSICIÓN DE LOS DÍGITOS													
millones	cientos miles	diez miles	miles	centenas	decenas	unidades	con	decimas	centésimas	milésimas	diezmilésimas	cientmilésimas	millonésimas
7	6	5	4	3	2	1	0	1	2	3	4	5	6

Tabla 15 Números decimales-2

Veamos algunos ejemplos:

- 6,4 → ses con catro **décimas** (1 dígito decimal)
- 50,03 → cincuenta con tres **centésimas** (2 dígito decimales)
- 121,564 → cento vinte e un con cincocentas sesenta e catro **milésimas** (3 dígito decimales)
- 5.213,4322 → cinco mil douscentos trece con catro mil trescentas vinte e dúas **diezmilésimas** (4 dígito decimales)
- 82.691,05045 → oitenta e dous mil seiscentos noventa e un con cinco mil corenta e cinco **centmilésimas** (5 dígito decimales)
- 540.654,686142 → cincocentos corenta mil seiscentos cincuenta e catro con seiscentas oitenta e ses mil cento corenta e dúas **millonésimas** (6 dígito decimales)
- 1.250.126,456798 → un millón douscentos cincuenta mil cento vinte e ses con catrocenas cincuenta e ses mil setecenas noventa e oito **millonésimas** (6 dígito decimal)

1.2 Objetivos a cubrir en este trabajo

Como hemos observado, el dominio de la escritura de números en palabras es fundamental para un uso preciso y adecuado del idioma. Es cierto que escribir números pequeños puede parecer bastante sencillo, pero ¿qué sucede cuando debemos enfrentarnos a números mucho más grandes? Aunque esta tarea no es común en la vida cotidiana, puede ser de gran interés desde una perspectiva curiosa o para enriquecer nuestros conocimientos sobre el idioma. Por lo tanto, resulta valioso contar con una herramienta que nos permita generar estas complejas combinaciones de palabras de manera eficaz y precisa.

En consecuencia, la presentación de este servicio web pretende cubrir los siguientes objetivos:

- Proporcionar una herramienta de verificación ortográfica para la escritura de números.
- Informar sobre las distintas alternativas a la hora de escribir un número con palabras.
- Enseñar sobre la forma adecuada de concatenar las palabras que conforman un número, sin importar cuán extenso sea dicho número.
- Brindar soporte y complementar el funcionamiento de otras aplicaciones informáticas.
- Ahorro de Tiempo, pues facilita el proceso de escritura, ya que los usuarios pueden verificar rápidamente la forma adecuada de expresar números en palabras.

Además, otro de los objetivos importantes de este proyecto es abordar la limitación presente en las tecnologías de inteligencia artificial y sistemas de lectura automática por voz, que a menudo no son capaces de leer cifras de manera eficiente, especialmente cuando se trata de cifras grandes. Estos sistemas tienden a leer las cifras una a una, lo que puede resultar en una comunicación poco fluida y natural. Por lo tanto, este proyecto busca proporcionar una solución que permita una representación textual adecuada de cifras, incluso en contextos donde las tecnologías de lectura automática pueden presentar limitaciones.

2. JUSTIFICACIÓN DE LAS COMPETENCIAS ESPECÍFICAS CUBIERTAS.

En la realización de este proyecto, se han abordado y desarrollado las áreas o competencias específicas que se detallan a continuación.

2.1 Trabajo fin de grado

- **TFG01:**” Ejercicio original a realizar individualmente y presentar y defender ante un tribunal universitario, consistente en un proyecto en el ámbito de las tecnologías

específicas de la Ingeniería en Informática de naturaleza profesional en el que se sinteticen e integren las competencias adquiridas en las enseñanzas.”

Este proyecto ha sido elaborado de manera individual por el autor, con la supervisión de su tutor. Durante su realización, se han empleado conceptos y metodologías adquiridos a lo largo de la carrera, con un enfoque particular en la programación orientada a objetos.

2.2 Comunes a la ingeniería informática

- **CI101:**” Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.”

El diseño y desarrollo del proceso de conversión han sido elementos fundamentales en la creación de una aplicación eficiente y sostenible. En este sentido, se han aplicado herramientas clave, como StringBuilder y técnicas de paralelismo, para optimizar el rendimiento y la velocidad de procesamiento.

Además, la inclusión de múltiples idiomas en la página no solo ha ampliado la accesibilidad de la aplicación, sino que también ha mejorado significativamente la experiencia del usuario. Esta característica permite que los usuarios de habla española, inglesa y gallega utilicen la aplicación de manera más cómoda y efectiva, lo que es un avance importante en términos de accesibilidad y usabilidad.

- **CI108:** “Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados”

La ejecución de mi proyecto, que involucra el uso de tecnologías ASP.NET y WFC para crear un conversor de números a texto en gallego mediante el lenguaje de programación C#, resalta mi capacidad para analizar, diseñar, construir y mantener aplicaciones de manera sólida, segura y eficiente. Durante este proceso, he tomado decisiones fundamentales al elegir los paradigmas y lenguajes de programación más apropiados para lograr los objetivos de manera efectiva.

- **CI014:** “Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.”

Dado que se trata de un servicio web con la expectativa de una amplia demanda, se ha requerido la incorporación de programación paralela en parte del código. Esta elección, sin duda, proporciona al servicio tiempos de respuesta significativamente más rápidos en comparación con un enfoque de programación convencional.

2.3 Tecnologías de la información

- **TI06:** “Capacidad de concebir sistemas, aplicaciones y servicios basados en tecnologías de red, incluyendo Internet, web, comercio electrónico, multimedia, servicios interactivos y computación móvil.”

Como se ha mencionado en varias ocasiones a lo largo de este informe, hemos abordado el problema planteado mediante el desarrollo de una solución basada en un servicio web. Esto requiere al menos un conocimiento básico acerca de lo que implica un servicio web y de las oportunidades que brinda a través de la red de Internet.

3. APORTACIONES

Podemos afirmar que a través de este proyecto se realizan contribuciones en las siguientes áreas:

- **Educativo:** En este ámbito específico, se encuentra la contribución principal del trabajo. La utilidad de este servicio no se limita únicamente a los estudiantes del idioma gallego, sino que también se extiende a personas cuya lengua materna es el gallego. Esto se debe a que aborda de manera eficaz y práctica un desafío que a menudo resulta complicado: la conversión de números de gran magnitud en palabras.

Este proceso de conversión se vuelve especialmente incómodo cuando se trabaja con cifras extremadamente grandes, ya que los términos asociados a estas cifras suelen ser desconocidos para la mayoría de las personas que no están familiarizadas con el ámbito de las matemáticas o las ciencias exactas.

Por ejemplo, en contextos académicos, profesionales o incluso en situaciones cotidianas, como la gestión financiera o la contabilidad, es común encontrarse con números de gran envergadura. La capacidad de nuestro servicio para traducir estos números en palabras no solo ahorra tiempo y esfuerzo, sino que también facilita la

comprensión de cantidades bastante grandes, lo que puede ser esencial para la toma de decisiones informadas.

Además, para aquellos que deseen aprender el idioma gallego o mejorar sus habilidades en él, esta herramienta se convierte en una valiosa fuente de práctica y comprensión, ya que permite ver cómo se construyen los números en gallego de manera contextual y real.

- **Jurídico/Administrativo:** En el ámbito de documentos oficiales y textos legales, se emplea una práctica común que consiste en representar cifras y números utilizando tanto su forma numérica, es decir, con dígitos, como su forma escrita en palabras. Esta práctica se lleva a cabo con el objetivo de garantizar la máxima claridad y concisión en la comunicación de información numérica.
- **Social:** Puede emplearse en combinación con otras aplicaciones destinadas a la conversión automática de números en texto a voz. Estas aplicaciones están experimentando un crecimiento significativo y su relevancia está en constante aumento, sobre todo para personas que enfrentan desafíos de accesibilidad o discapacidades.

4. DISEÑO

El proyecto se estructura en dos componentes esenciales que trabajan en conjunto para brindar una solución integral. Por un lado, tenemos el componente frontend, que se desarrolla utilizando ASP.NET. Esta parte del proyecto es la interfaz visible para los clientes y usuarios finales. En ella, se presentan los resultados de las conversiones de números a texto en gallego.

Además de los resultados en sí, el frontend proporciona información adicional, como ejemplos morfológicos, usos textuales y notas relacionadas con los números traducidos. Un aspecto destacado del frontend es su capacidad de adaptación, lo que significa que la interfaz se ajusta de manera óptima a diferentes dispositivos y tamaños de pantalla. Además, se ha diseñado para ser multilingüe, lo que permite a los usuarios acceder a la información en varios idiomas. Este logro se basa en el uso de recursos, lo que facilita la gestión de contenido en diferentes lenguas.

Por otro lado, el componente backend se implementa utilizando WCF (Windows Communication Foundation). Este es el corazón de la lógica detrás de la traducción de números a texto en gallego. La labor del backend es llevar a cabo todo el procesamiento necesario para realizar esta conversión de manera precisa y eficiente. Para lograrlo, se hace uso de expresiones

regulares, que son herramientas poderosas para reconocer y manipular patrones en los números de entrada. Además, se emplean técnicas de programación paralela, lo que permite acelerar el proceso de conversión al distribuir la carga de trabajo en múltiples hilos de ejecución. Esta estrategia se vuelve crucial cuando se trata de números extremadamente grandes o tareas intensivas en cálculos. Asimismo, se lleva a cabo un minucioso tratamiento de las cadenas de caracteres para garantizar la coherencia y la precisión en la traducción.

4.1 ASP.NET

En lo que respecta al diseño del frontend, el objetivo ha sido crear una experiencia de usuario intuitiva y amigable. Para lograrlo, hemos desarrollado una página centralizada que se centra en un formulario interactivo. En este formulario, los usuarios tienen la posibilidad de ingresar el número que desean traducir a gallego.

ULPGC Universidad de Las Palmas de Gran Canaria | Instituto Universitario de Análisis y Aplicaciones Textuales | IA text

Galego English Español

Números Gallegos

Convertor de número a texto en gallego.

Escriba un número con cifras

Se puede convertir números a letras en cardinal, números en letras ordinal, números en letras decimal, números en letras fraccionario o partitivo y multiplicativo. Se ofrece información morfológica, ortográfica y gramatical de cada uno de los números convertidos a letras. Además, se incluyen ejemplos que ayudan a la comprensión y buen uso.

Escriba un número sin coma ni punto para los miles o millones. Los miles o millones se pueden separar con un espacio en blanco. En los decimales use la coma ",". Para escribir fracciones use la barra inclinada "/".

Prueba los ejemplos

13289	53625999567	345676
13 289	53 625 999 567	345 676
3/4	78/125	100.000/250
3,4	78,125	100.000,250
		345 676/13 289
		345 676,13 289

ULPGC Instituto Universitario de Análisis y Aplicaciones Textuales | IA text

Peraza Machín, J. Trabajo de fin de grado - Ingeniería informática - 2023. Convertidor de números a texto en gallego - Números TIP. Carreras-Riudavets, F Tutor de TFG.

Ilustración 1 Front - vista global

Además de este formulario, hemos incluido elementos informativos que enriquecen la experiencia del usuario. Un texto explicativo detalla las instrucciones de uso de la aplicación, proporcionando orientación sobre cómo utilizarla de manera efectiva. Este texto informativo es esencial para que los usuarios comprendan cómo interactuar con la herramienta y obtengan los resultados deseados de manera rápida y sencilla. Adicionalmente, hemos incorporado ejemplos representativos que los usuarios pueden explorar haciendo clic en ellos. Al hacer clic en estos ejemplos, los usuarios pueden ver cómo se realiza la conversión y obtener una idea más clara de cómo utilizar la aplicación para sus propios números.

Se puede convertir números a letras en cardinal, números en letras ordinal, números en letras decimal, números en letras fraccionario o partitivo y multiplicativo. Se ofrece información morfológica, ortográfica y gramatical de cada uno de los números convertidos a letras. Además, se incluyen ejemplos que ayudan a la comprensión y buen uso.

Escriba un número sin coma ni punto para los miles o millones. Los miles o millones se pueden separar con un espacio en blanco. En los decimales use la coma ",". Para escribir fracciones use la barra inclinada "/".

[Prueba los ejemplos](#)

13289	53625999567	345676
13 289	53 625 999 567	345 676
3/4	78/125	100.000/250
3,4	78,125	100.000,250

Ilustración 2 Front - Información y Ejemplos

También, hemos incorporado íconos con las banderas de Galicia, España e Inglaterra. Al hacer clic en cualquiera de estos iconos, la aplicación cambiará automáticamente al idioma correspondiente a la bandera seleccionada. Esto proporciona a los usuarios la conveniencia de acceder a la aplicación en su idioma preferido con un simple clic en la bandera deseada. Como se puede ver en la siguiente foto:

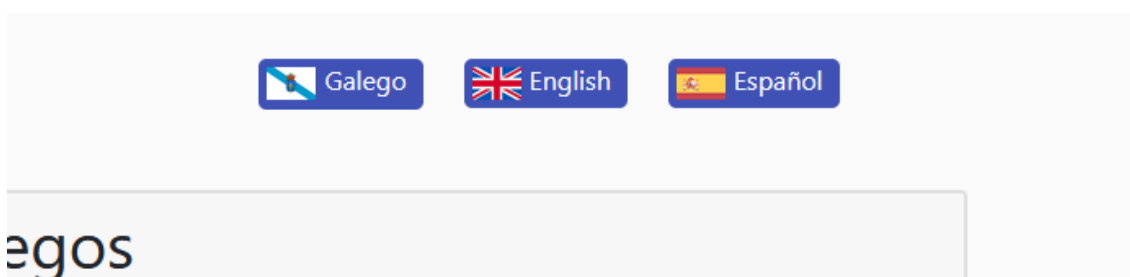


Ilustración 3 Front - Idiomas

Por último, hemos llevado a cabo la adaptación de la página para garantizar su funcionamiento y visualización óptimos en dispositivos móviles, tal como se puede apreciar en la imagen adjunta. Esta adaptación es fundamental, ya que permite a los usuarios acceder a la aplicación de manera efectiva y cómoda desde sus teléfonos:

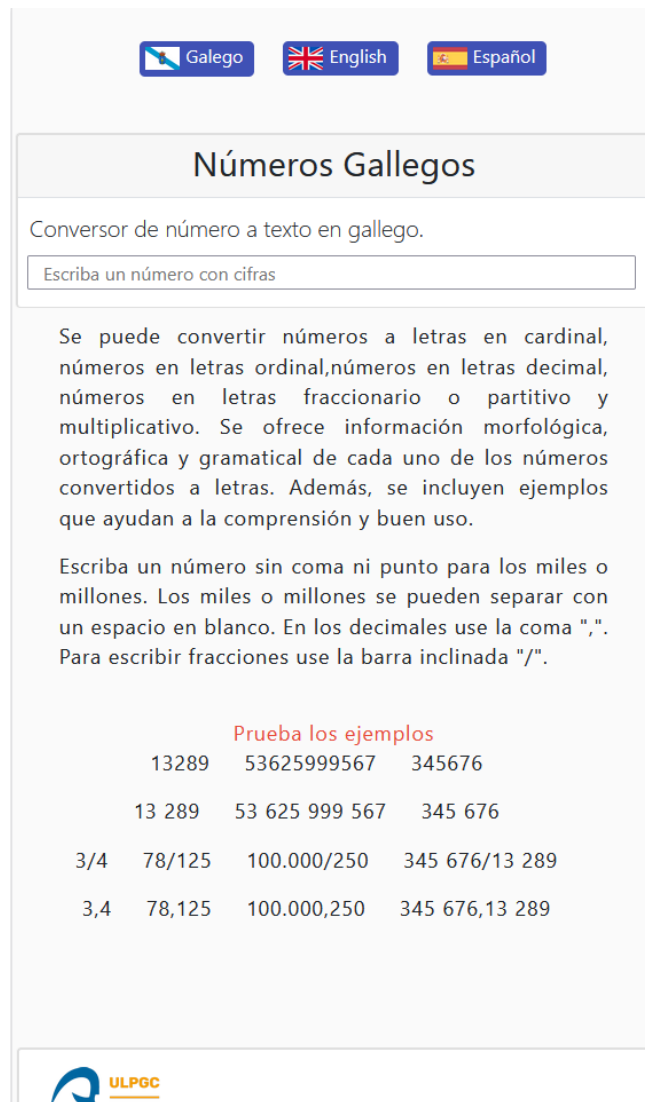


Ilustración 4 Front - Versión móvil

4.1.1 Respuesta

En lo que respecta a la respuesta del servicio, la aplicación mostrará tarjetas específicas según el formato del número de entrada. En la mayoría de los casos, se generarán cuatro tarjetas diferentes, cada una de ellas representando una forma de expresar el número en gallego.



Ilustración 5 Front - Tarjetas

Estas tarjetas incluirán la traducción del número en cuatro categorías principales: cardinal, ordinal, fraccionario y multiplicativo.

En cada tarjeta, encontrarás un menú desplegable que, cuando lo activas, muestra una lista completa de las opciones y funcionalidades asociadas al número, junto con ejemplos ilustrativos. Esta característica proporciona a los usuarios una visión detallada y completa de cómo se puede utilizar el número en diferentes contextos y aplicaciones. Al hacer clic en el menú desplegable, los usuarios pueden explorar todas las posibilidades y ejemplos relacionados con la traducción del número:

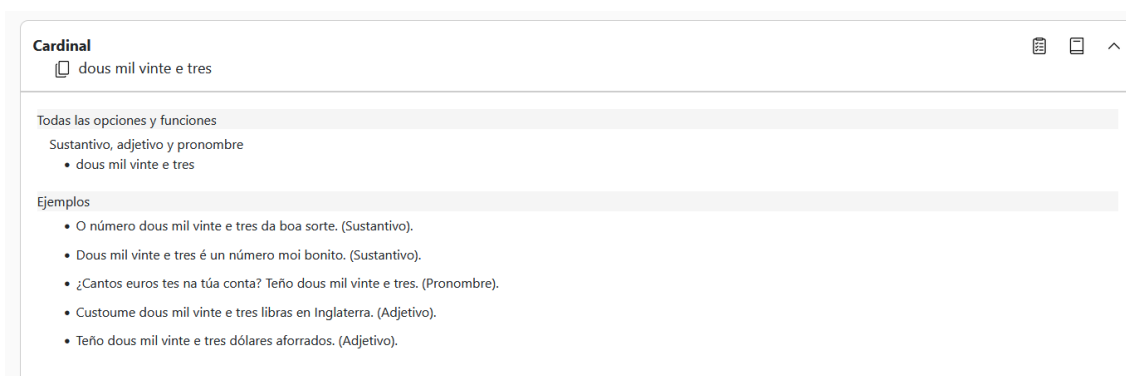


Ilustración 6 Front - Tarjeta desplegada

Además, en la parte derecha de cada tarjeta, hemos colocado botones que al hacer clic en ellos abrirán ventanas emergentes (pop-ups) que ofrecen información detallada sobre esa categoría de número y referencias a diccionarios relacionados. Estos recursos adicionales enriquecen la comprensión del usuario al proporcionar detalles específicos sobre la clase de número que están explorando y fuentes adicionales para obtener más información si es necesario.

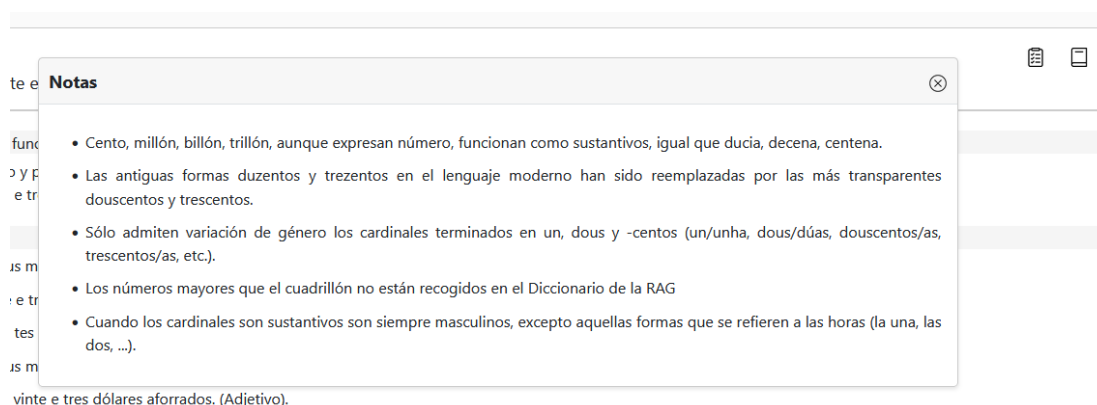


Ilustración 7 Front - Pop-up Notas



Ilustración 8 Front - Pop-up Referencias

4.1.2 Error

Cuando el servicio recibe una entrada, ya sea que esté mal escrita o simplemente el número sea demasiado grande para procesar, se crea una tarjeta de error. En esta tarjeta, se muestran al cliente posibles errores que podrían haber ocurrido, proporcionando una guía útil para identificar y corregir los problemas en la entrada introducida. Como se puede observar en la siguiente imagen:

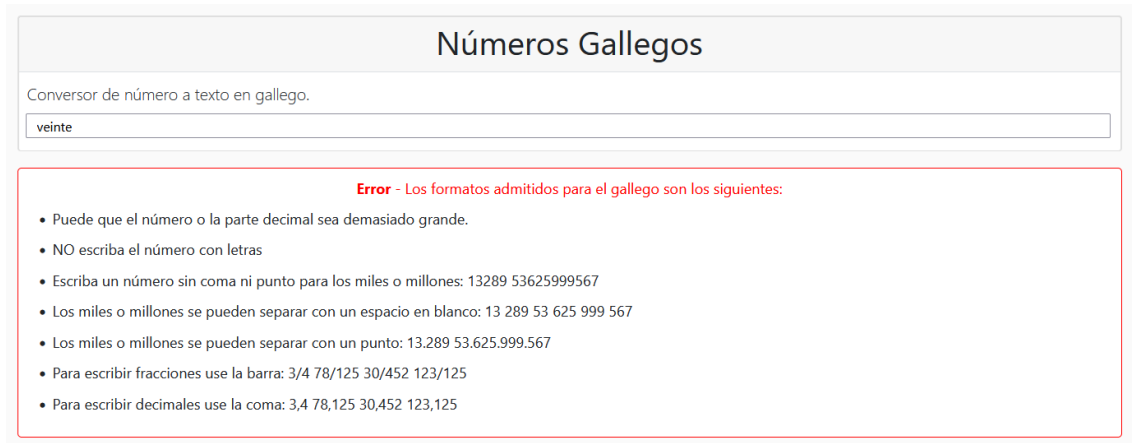


Ilustración 9 Front - Tarjeta de error

4.2 WCF

En lo que respecta al componente backend, hemos empleado Windows Communication Foundation (WCF) para crear un servicio. Este servicio desempeña un papel esencial en el proceso de traducción, ya que se encarga de recibir tanto el valor numérico que se desea traducir como el idioma seleccionado en la página web para entregar el resultado solicitado.

El funcionamiento del backend es sumamente eficiente y se basa en una estrecha colaboración con el frontend de la aplicación. Cuando un usuario ingresa un número y selecciona un idioma en la página web, esta información se envía al servicio WCF en segundo plano. El servicio WCF procesa esta solicitud y realiza la traducción del número al idioma especificado.

4.2.1 Entrada

El número ingresado debe seguir al menos uno de los formatos siguientes, y es importante mantener estrictamente el orden especificado para sus partes:

- **Número entero**

Formato 1: Sin separadores (ej. 1234)

Formato 2: Usando punto como separador (ej. 1.234)

Formato 3: Usando un espacio en blanco como separador (ej. 1 234)

[OBLIGATORIO] de 1 a 30 dígitos numéricos.

- **Número fraccionario**

Formato 1: Sin separadores dentro del numerador o denominador (ej. 1234/1000)

Formato 2: Usando punto como separador dentro del numerador o denominador (ej. 1.234/1.000)

Formato 3 Usando un espacio en blanco como separador dentro del numerador o denominador (ej. 1 234/1 000)

[OBLIGATORIO] un signo / para separar el numerador del denominador.

[OBLIGATORIO] de 1 a 30 dígitos numéricos en el numerador.

[OBLIGATORIO] de 1 a 30 dígitos numéricos en el denominador.

- **Número decimal**

Formato 1: Sin separadores dentro de la parte entera o decimal (ej. 1234,1000)

Formato 2: Usando punto como separador dentro de la parte entera (ej. 1.234,1000)

Formato 3: Usando un espacio en blanco como separador dentro de la parte entera (ej. 1 234,1000)

[OBLIGATORIO] una coma “,” para separar la parte entera de la decimal.

[OBLIGATORIO] de 1 a 30 dígitos numéricos en la parte entera.

[OBLIGATORIO] de 1 a 30 dígitos numéricos en la parte decimal.

4.2.2 Salidas

El resultado que proporciona el servicio es una lista `<Conversion>`, la cual recopila todos los resultados necesarios como parte del proceso de conversión. Esta lista está estructurada utilizando la clase `Conversion` previamente definida. El proceso de conversión se divide en múltiples tareas que se ejecutan de manera simultánea, y luego se combinan todos los resultados para crear el conjunto final de resultados definitivos.

4.2.3 Panorama general del proceso de conversión.

Antes de adentrarnos en una descripción detallada de la estructura de la clase `Conversor`, es fundamental tener una comprensión sólida del enfoque que guía el mecanismo de conversión. Este proceso de conversión ofrece múltiples rutas alternativas para alcanzar los resultados deseados, y estas rutas se seleccionan en función del formato específico del número de entrada. La clave para una traducción precisa y versátil radica en la capacidad de adaptación a diferentes situaciones.

En este contexto, el diagrama de flujo de conversión, que se muestra en la figura a continuación, emerge como una herramienta invaluable. Este diagrama ofrece una representación visual y lógica de las diversas sendas que el proceso de conversión puede seguir. Cada una de estas rutas se ajusta a las características particulares del número de entrada, permitiendo así una traducción precisa y coherente en cada caso.

Este enfoque basado en múltiples caminos garantiza que el proceso de conversión sea flexible y capaz de lidiar con una amplia gama de situaciones. Al comprender estas alternativas y tener una visualización clara de cómo se desarrolla el proceso a través del diagrama de flujo, estamos preparados para explorar con mayor profundidad la estructura interna de la clase `Conversor` y su papel en la ejecución de estas conversiones versátiles.

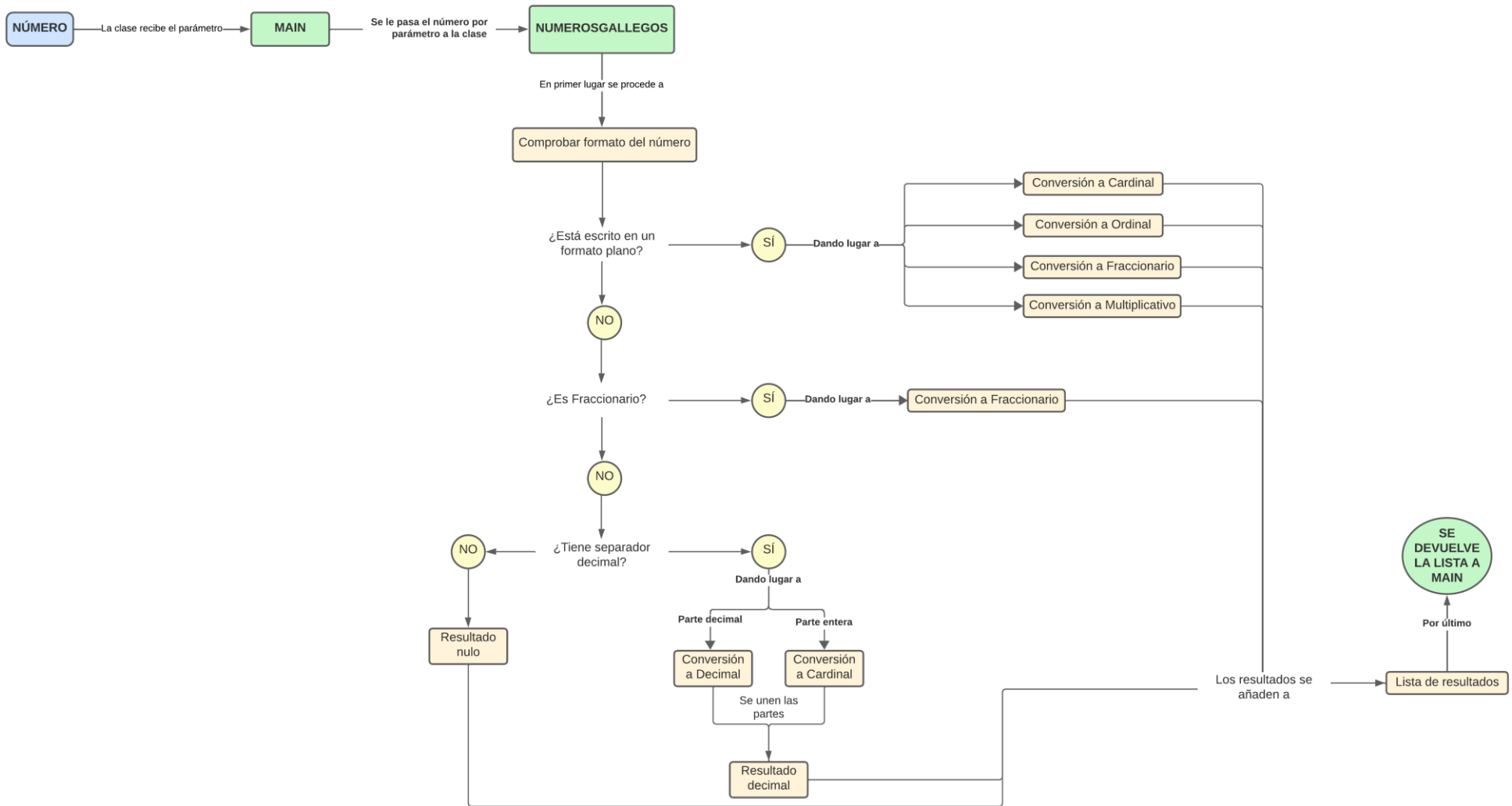


Ilustración 10 Back - Diagrama de flujo

El proceso se puede resumir en los siguientes pasos:

1. Desde una aplicación externa, en nuestro caso, el servicio web ASP.NET mencionado anteriormente, se inicia una instancia del proyecto WCF.
2. Se envía el número a convertir como parámetro a esta instancia del proyecto WCF.

A partir de este punto, se inicia la ejecución del método principal y se llevan a cabo las siguientes acciones:

3. Se pasa el valor a traducir como parámetro a la clase "NumerosGallegos".
4. Se realiza una validación del formato del número. Si el formato no es correcto, se envía una lista nula. En cambio, si el formato es correcto, el número se analiza de la siguiente manera:
 - En caso de detectarse un signo de división ("/"), que se considera un indicador de que se trata de un número fraccionario.
 - Si se identifica una coma, que se considera como un separador decimal, se procede a descomponer el número en su parte entera y parte decimal.
 - Si se identifica un formato plano, lo que significa que el número de entrada no incluye elementos como signos de división o puntos decimales que indiquen una estructura más compleja, entonces el número se mantiene en su forma original sin ninguna alteración. En otras palabras, si el número no presenta indicios de ser fraccionario o tener una parte decimal, se conserva intacto para su posterior procesamiento.
5. Con cada una de las partes validadas (ya sea la parte entera y la parte decimal cuando el número es decimal, o el numerador y el denominador cuando es fraccionario) se somete a tareas de conversión a texto de forma simultánea. Estas tareas de conversión pueden generar varios tipos de resultados, que incluyen texto cardinal, texto ordinal, texto multiplicativo y otros, según corresponda.

6. Luego de permitir que se completen todas las conversiones, los resultados obtenidos se añaden a una lista. Esta lista actúa como un contenedor que almacena todos los resultados generados durante el proceso de conversión. Una vez que se han agregado todos los resultados a esta lista, se procede a devolver esta lista al método principal del programa.
7. Después de que la lista con las traducciones se devuelve al método principal, se inicia el proceso de rellenar una lista `<Conversion>`. Dentro de esta nueva lista, se incorporan gradualmente las diversas opciones de traducción, ejemplos del número traducido, notas sobre la gramática numérica y referencias a diccionarios. En esencia, esta lista `<Conversion>` sirve como un recipiente integral que almacena de manera organizada toda la información relacionada con las traducciones del número. Cada elemento de esta lista representa una conversión específica y contiene detalles que incluyen las diferentes formas de traducción, ejemplos ilustrativos, explicaciones sobre la gramática numérica y enlaces a recursos de diccionarios, si es necesario.

Una vez más, desde el servicio web de ASP.NET:

8. Una vez que se ha obtenido la lista `<Conversion>`, se inicia la fase de construcción de las tarjetas que se utilizarán para presentar los resultados al cliente. Cada tarjeta se crea de manera específica para mostrar una traducción particular y su información asociada de manera clara y organizada.

4.2.4 Estructura de la clase Conversión

La clase `<Conversion>` es una representación de un objeto que almacena información relacionada con la traducción de un número a texto en un contexto específico, como cardinal, ordinal, fraccionario, etc. Esta clase tiene múltiples propiedades y subclases para organizar de manera estructurada los diferentes aspectos de la traducción. A continuación, se explican las principales partes y propiedades de la clase `<Conversion>`:

1. ****Propiedades públicas:****

- `<Tipo>`: Almacena el tipo de traducción, como cardinal, ordinal, etc.
- `<Titulo>`: Representa un título o encabezado relacionado con la traducción.
- `<Respuestas>`: Es una lista de respuestas relacionadas con la traducción.
- `<RespuestasMayuscula>`: Similar a `<Respuestas>`, pero en mayúsculas.

- `TitValorNumerico`: Representa un título relacionado con el valor numérico.
- `ValorNumerico`: Almacena el valor numérico de la traducción.
- `TitOpciones`: Representa un título relacionado con las opciones.
- `MasOpciones`: Es una lista de objetos de tipo `Opcion`, que representan diferentes opciones relacionadas con la traducción.
- `MasOpcionesMayuscula`: Similar a `MasOpciones`, pero en mayúsculas.
- `TitEjemplos`: Representa un título relacionado con ejemplos de uso.
- `Ejemplos`: Es una lista de ejemplos relacionados con la traducción.
- `EjemplosMayuscula`: Similar a `Ejemplos`, pero en mayúsculas.
- `TitNotas`: Representa un título relacionado con notas gramaticales.
- `Notas`: Es una lista de notas gramaticales relacionadas con la traducción.
- `TitReferencias`: Representa un título relacionado con referencias a diccionarios.
- `Referencias`: Es una lista de referencias a diccionarios relacionadas con la traducción.

2. ****Subclase `Opcion`:****

- Esta subclase se utiliza para representar opciones relacionadas con la traducción. Contiene las siguientes propiedades:

- `Titulo`: Almacena un título relacionado con las opciones.
- `Opciones`: Es una lista de opciones específicas.

En resumen, la clase `Conversion` se utiliza para organizar y almacenar de manera estructurada la información relacionada con la traducción de números a texto en diferentes contextos. Cada propiedad y subclase tiene un propósito específico en la representación de los detalles de la traducción, lo que facilita la presentación ordenada de esta información.

A continuación, se presenta la estructura de la lista de objetos de tipo `Conversion` que se obtiene para los números siguientes:

- a. **789.457.175.917** (Ilustración 11 Back - Estructura lista conversión-1)
- b. **841/321** (Ilustración 12 Back - Estructura lista conversión-2)
- c. **213,4554** (Ilustración 13 Back - Estructura lista conversión-3)

return

[0]

Ejemplos

[0]

[1]

[2]

[3]

[4]

EjemplosMayuscula

ErrorRomano

MasOpciones

[0]

Opciones

[0]

Titulo

MasOpcionesMayuscula

Notas

Referencias

[0]

[1]

[2]

Respuestas

[0]

RespuestasMayuscula

Tipo

TitEjemplos

TitNotas

TitOpciones

TitReferencias

TitValorNumerico

Titulo

ValorNumerico

[1]

Ejemplos

[0]

[1]

[2]

[3]

EjemplosMayuscula

ErrorRomano

MasOpciones

[0]

Opciones

[0]

Titulo

[1]

Opciones

[0]

MasOpcionesMayuscula

Notas

[0]

length=4

length=5

"O número setecentos oitenta e nove mil millóns catrocentos cincuenta e sete millo" + "ns cento setenta e cinco mil novecentos dezasete da boa sorte. (Sustantivo)."

"Setecentos oitenta e nove mil millóns catrocentos cincuenta e sete millóns cento " + "setenta e cinco mil novecentos dezasete é un número moi bonito. (Sustantivo)."

"¿Cantos euros tes na túa conta? Teño setecentos oitenta e nove mil millóns catroc" + "entos cincuenta e sete millóns cento setenta e cinco mil novecentos dezasete. (P" + "System.String

"Custome setecentos oitenta e nove mil millóns catrocentos cincuenta e sete millo" + "ns cento setenta e cinco mil novecentos dezasete libras en Inglaterra. (Adjetivo" + "System.String

"Teño setecentos oitenta e nove mil millóns catrocentos cincuenta e sete millóns c" + "ento setenta e cinco mil novecentos dezasete dólares aforados. (Adjetivo)."

length=5

length=1

length=1

"setecentos oitenta e nove mil millóns catrocentos cincuenta e sete millóns cento " + "setenta e cinco mil novecentos dezasete"

"Sustantivo, adxectivo y pronome"

length=0

length=5

length=3

"Normas ortográficas e morfolóxicas do idioma galego | https://publicacions.acade" + "mia.gal/index.php/rag/catalog/book/252"

"Real academia galega | https://academia.gal"

"Portal das palabras | https://portaldaspalabras.gal"

length=1

"setecentos oitenta e nove mil millóns catrocentos cincuenta e sete millóns cento " + "setenta e cinco mil novecentos dezasete"

length=1

"Cardinal"

"Ejemplos"

"Notas"

"Todas las opciones y funciones"

"Referencias"

""

""

""

""

""

""

""

length=4

"A setecentos oitenta nove mil millóns catrocentos cincuenta sete millo" + "setenta cinco mil nonixentésimo décimo sétima persoa que remate os seus estudo" + "System.String

"¿Que pasaxeiro recibirá o premio? o setecentos oitenta nove mil millóns catrocen" + "tos cincuenta sete millóns cento setenta cinco mil nonixentésimo décimo sétim" + "o System.String

"No maratón quedei de setecentos oitenta nove mil millóns catrocentos cincuenta s" + "ete millóns cento setenta cinco mil nonixentésimo décimo sétimo lugar. (Adjet" + "ivi System.String

"Esta é a setecentos oitenta nove mil millóns catrocentos cincuenta sete millo" + "cento setenta cinco mil nonixentésimo décimo sétimo vez que leo este libro. (" + "Adjet System.String

length=4

False

length=2

length=1

"setecentos oitenta nove mil millóns catrocentos cincuenta sete millo" + "setenta cinco mil nonixentésimo décimo sétimo"

"Adjetivo y pronome"

length=1

"setecentos oitenta nove mil millóns catrocentos cincuenta sete millo" + "setenta cinco mil nonixentésimo décimo sétimo"

"Forma feminina"

length=0

length=3

"Todas estas formas admiten morfemas de género y número, que en las formas compues" + "tas sólo se adjuntan al último elemento (décimo primeira)."

ServicioNumerosGalegoWCF.Conversion[]

ServicioNumerosGalegoWCF.Conversion

System.String[]

System.String

System.String

System.String

System.String

System.String

System.String

System.Boolean

ServicioNumerosGalegoWCF.Opcion[]

ServicioNumerosGalegoWCF.Opcion

System.String[]

System.String

ServicioNumerosGalegoWCF.Opcion[]

System.String[]

System.String

System.String

System.String

System.String

System.String

System.String

System.String

System.String

System.String

System.String

System.String

System.String

System.String

ServicioNumerosGalegoWCF.Conversion

System.String[]

System.String

System.String

System.String

System.String

System.Boolean

ServicioNumerosGalegoWCF.Opcion[]

ServicioNumerosGalegoWCF.Opcion

System.String[]

System.String

System.String

ServicioNumerosGalegoWCF.Opcion

System.String[]

System.String

System.String

ServicioNumerosGalegoWCF.Opcion[]

System.String

System.String

[1]	"Los ordinales 11 y 12 admiten dos variantes undécimo y décimo primero, duodécimo " + "y décimo segundo."	System.String
[2]	"Generalmente son adjetivos, y pueden ir antepuestos o pospuestos al sustantivo, s" + "tendo más frecuente su aparición por delante."	System.String
Referencias	length=3	System.String[]
[0]	"Nommas ortográficas e morfolóxicas do idioma galego https://publicacions.acade " + "mia.gal/index.php/rag/catalog/book/252"	System.String
[1]	"Real academia galega https://academia.gal "	System.String
[2]	"Portal das palabras https://portaldaspalabras.gal "	System.String
Respuestas	length=1	System.String[]
[0]	"setecentos oitenta nove mil millóns catrocentos cincuenta sete millóns cento se" + "tenta cinco mil noninxentésimo décimo sétimo"	System.String
RespuestasMayuscula	length=1	System.String[]
Tipo	"Ordinal"	System.String
TitEjemplos	"Ejemplos"	System.String
TitNotas	"Notas"	System.String
TitOpciones	"Todas las opciones y funciones"	System.String
TitReferencias	"Referencias"	System.String
TitValorNumerico	""	System.String
Titulo	""	System.String
ValorNumerico	""	System.String
[2]	length=5	ServicioNumerosGallegoWCF.Conversio
Ejemplos	length=5	System.String[]
[0]	"Un setecentos oitenta e nove mil millóns catrocentos cincuenta e sete millóns cen" + "to setenta e cinco mil novecentos dezaseteavos das estrelas está lonxe. (Sustanti" + "vi	System.String
[1]	"Deixa de lado dous setecentos oitenta e nove mil millóns catrocentos cincuenta e " + "sete millóns cento setenta e cinco mil novecentos dezaseteavos do total de moeda" +	System.String
[2]	"A calvicie sófrea setecentos oitenta e nove mil millóns catrocentos cincuenta e s" + "ete millóns cento setenta e cinco mil novecentos dezaseteava parte da poboación " + "i	System.String
[3]	"Dúas setecentos oitenta e nove mil millóns catrocentos cincuenta e sete millóns c" + "erto setenta e cinco mil novecentos dezaseteavas partes de auga son demasiado po" +	System.String
[4]	"Pedinche un terzo do teu tempo e non me deches nin un setecentos oitenta e nove m" + "il millóns catrocentos cincuenta e sete millóns cento setenta e cinco mil novece" +	System.String
EjemplosMayuscula	length=5	System.String[]
ErrorRomano	False	System.Boolean
MasOpciones	length=1	ServicioNumerosGallegoWCF.Opcion[]
[0]	length=1	ServicioNumerosGallegoWCF.Opcion
Opciones	length=1	System.String[]
Titulo	"Adjetivo y pronombre femenino"	System.String
MasOpcionesMayuscula	length=0	ServicioNumerosGallegoWCF.Opcion[]
Notas	length=3	System.String[]
[0]	"A partir de 1/11 (y exceptuados 1/100, 1/1000,1/1.000.000...) el partitivo se hac" + "e añadiendo el sufijo -avo al cardinal: (1/16) un dezaseisavo, (2/29) dous vinte" + " e n	System.String
[1]	"Todas estas formas admiten morfemas de género y número, que en las formas compues" + "tas sólo se adjuntan al último elemento (otavo, -a)."	System.String
[2]	"Generalmente, los fraccionarios femeninos son adjetivos y los masculinos sustanti" + "vos."	System.String
Referencias	length=3	System.String[]
[0]	"Nommas ortográficas e morfolóxicas do idioma galego https://publicacions.acade " + "mia.gal/index.php/rag/catalog/book/252"	System.String
[1]	"Real academia galega https://academia.gal "	System.String
[2]	"Portal das palabras https://portaldaspalabras.gal "	System.String
Respuestas	length=1	System.String[]
[0]	"setecentos oitenta e nove mil millóns catrocentos cincuenta e sete millóns cento " + "setenta e cinco mil novecentos dezaseteavo"	System.String
RespuestasMayuscula	length=1	System.String[]
Tipo	"Fraccionario"	System.String
TitEjemplos	"Ejemplos"	System.String
TitNotas	"Notas"	System.String
TitOpciones	"Todas las opciones y funciones"	System.String
TitReferencias	"Referencias"	System.String
TitValorNumerico	""	System.String
Titulo	""	System.String
ValorNumerico	""	System.String
[3]	length=4	ServicioNumerosGallegoWCF.Conversio
Ejemplos	length=4	System.String[]

[0]	"Tes setecentos oitenta e nove mil millóns catrocentos cincuenta e sete millóns ce" +	"nto setenta e cinco mil novecentos dezasete veces máis diñeiro que o ano pasado." +	System.String
[1]	"Este número é setecentos oitenta e nove mil millóns catrocentos cincuenta e sete" +	"millóns cento setenta e cinco mil novecentos dezasete veces maior que este outro" +	System.String
[2]	"Debes de multiplicar por setecentos oitenta e nove mil millóns catrocentos cincue" +	"nta e sete millóns cento setenta e cinco mil novecentos dezasete as túas gananci" +	System.String
[3]	"Detido por multiplicar por setecentos oitenta e nove mil millóns catrocentos cinc" +	"uenta e sete millóns cento setenta e cinco mil novecentos dezasete a taxa de alc" +	System.String
▶ EjemplosMayuscula	length=4		System.String[]
ErrorRomano	False		System.Boolean
▲ MasOpciones	length=2		ServicioNumerosGallegoWCF.Opcion[]
▲ [0]			ServicioNumerosGallegoWCF.Opcion
▶ Opciones	length=1		System.String[]
Titulo	"Expresión con adjetivo"		System.String
▲ [1]			ServicioNumerosGallegoWCF.Opcion
▲ Opciones	length=1		System.String[]
[0]	"multiplicar por setecentos oitenta e nove mil millóns catrocentos cincuenta e set" +	"e millóns cento setenta e cinco mil novecentos dezasete"	System.String
Titulo	"Expresión con verbo"		System.String
MasOpcionesMayuscula	length=0		ServicioNumerosGallegoWCF.Opcion[]
▶ Notas	length=3		System.String[]
[0]	"A partir de 11 (y excepto el 100) e forman mediante el cardinal correspondiente "	"seguido de la palabra veces."	System.String
[1]	"Como sustantivos son siempre masculinos."		System.String
[2]	"Los numeros del 2 al 10, y el 100 tiene forma femenina."		System.String
▲ Referencias	length=3		System.String[]
[0]	"Normas ortográficas e morfolóxicas do idioma galego https://publicacions.acade " +	"mia.gal/index.php/rag/catalog/book/252"	System.String
[1]	"Real academia galega https://academia.gal "		System.String
[2]	"Portal das palabras https://portaldaspalabras.gal "		System.String
▲ Respuestas	length=1		System.String[]
[0]	"setecentos oitenta e nove mil millóns catrocentos cincuenta e sete millóns cento" +	"setenta e cinco mil novecentos dezasete veces"	System.String
▶ RespuestasMayuscula	length=1		System.String[]
Tipo	"Multiplicativo"		System.String
TitEjemplos	"Ejemplos"		System.String
TitNotas	"Notas"		System.String
TitOpciones	"Todas las opciones y funciones"		System.String
TitReferencias	"Referencias"		System.String
TitValorNumerico	""		System.String
Titulo	""		System.String
ValorNumerico	""		System.String

Ilustración 11 Back - Estructura lista conversión-1

[0]			
Ejemplos			ServicioNumerosGallegoWCF.Conversion
[0]	length=4	"Quero oitocentos corenta e un trescentos vinte e unavo de empanada. (Sustantivo)." + ""	System.String[]
[1]		"Deixa de lado oitocentos corenta e un trescentos vinte e unavos do total de moeda" + "s. (Sustantivo)."	System.String
[2]		"A calvicie sófrea oitocentos corenta e un trescentos vinte e unava parte da poboaa" + "ción mundial, segundo a OMS. (Adjetivo)."	System.String
[3]		"Oitocentos corenta e un trescentos vinte e unavas partes de auga son demasiado po" + "ucas para a fórmula química. (Adjetivo)."	System.String
EjemplosMayuscula			System.String[]
ErrorRomano			System.Boolean
MasOpciones			ServicioNumerosGallegoWCF.Opcion[]
[0]			ServicioNumerosGallegoWCF.Opcion
MasOpcionesMayuscula			ServicioNumerosGallegoWCF.Opcion[]
Notas			System.String[]
[0]	length=0	"A partir de 1/11 (y exceptuados 1/100, 1/1000,1/1.000.000...) el partitivo se hac" + "e añadiendo el sufijo -avo al cardinal: (1/16) un dezaseisavo, (2/29) dous vinte" + " e nri	System.String
[1]	length=3	"Todas estas formas admiten morfemas de género y número, que en las formas compues" + "tas sólo se adjuntan al último elemento (oitavo, -a)."	System.String
[2]		"Generalmente, los fraccionarios femeninos son adjetivos y los masculinos sustanti" + "vos."	System.String
Referencias	length=3		System.String[]
[0]		"Nommas ortográficas e morfolóxicas do idioma galego https://publicacions.acade" + "mia.gal/index.php/rag/catalog/book/252"	System.String
[1]		"Real academia galega https://academia.gal"	System.String
[2]		"Portal das palabras https://portaldaspalabras.gal"	System.String
Respuestas	length=1		System.String[]
[0]	length=1	"oitocentos corenta e un trescentos vinte e unavo"	System.String
RespuestasMayuscula	length=1		System.String[]
Tipo		"Fraccionario"	System.String
TitEjemplos		"Ejemplos"	System.String
TitNotas		"Notas"	System.String
TitOpciones		"Todas las opciones y funciones"	System.String
TitReferencias		"Referencias"	System.String
TitValorNumerico		""	System.String
Titulo		""	System.String
ValorNumerico		""	System.String

Ilustración 12 Back - Estructura lista conversión-2

[0]			
└ Ejemplos			
[0]			
[1]			
└ EjemplosMayuscula			
ErrorRomano			
└ MasOpciones			
└ [0]			
└ Opciones			
[0]			
Titulo			
MasOpcionesMayuscula			
└ Notas			
[0]			
[1]			
[2]			
└ Referencias			
[0]			
[1]			
[2]			
└ Respuestas			
[0]			
└ RespuestasMayuscula			
Tipo			
TitEjemplos			
TitNotas			
TitOpciones			
TitReferencias			
TitValorNumerico			
Titulo			
ValorNumerico			
	length=2	"O número douscentos trece con catro mil cincocentas cincuenta e catro dezmilésim" + "as da boa sorte. (Sintagma nominal)."	ServicioNumerosGallegoWCF.Conversion
		"O douscentos trece con catro mil cincocentas cincuenta e catro dezmilésimas é un" + " número moi bonito. (Sintagma nominal)."	System.String[]
			System.String
	length=2		System.String
	False		System.Boolean
	length=1		ServicioNumerosGallegoWCF.Opcion[]
			ServicioNumerosGallegoWCF.Opcion
			System.String[]
	length=1	"douscentos trece con catro mil cincocentas cincuenta e catro dezmilésimas"	System.String
		"Expresión"	System.String
	length=0		ServicioNumerosGallegoWCF.Opcion[]
	length=3		System.String[]
		"Las antiguas formas duzentos y trezentos en el lenguaje moderno han sido reemplaz" + "adas por las más transparentes douscentos y trescentos."	System.String
		"Sólo admiten variación de género los cardinales terminados en un, dous y -centos " + "(un/unha, dous/dúas, douscentos/as, trescentos/as, etc.)."	System.String
		"Los números mayores que el cuadrillón no están recogidos en el Diccionario de la " + "RAG"	System.String
	length=3		System.String[]
		"Nommas ortogrãficas e morfolóxicas do idioma galego https://publicacions.acade" + "mia.gal/index.php/rag/catalog/book/252"	System.String
		"Real academia galega https://academia.gal"	System.String
		"Portal das palabras https://portaldaspalabras.gal"	System.String
	length=1		System.String[]
		"douscentos trece con catro mil cincocentas cincuenta e catro dezmilésimas"	System.String
	length=1		System.String[]
		"Decimal"	System.String
		"Ejemplos"	System.String
		"Notas"	System.String
		"Todas las opciones y funciones"	System.String
		"Referencias"	System.String
		""	System.String
		""	System.String
		""	System.String

Ilustración 13 Back - Estructura lista conversión-3

5. DESARROLLO

La implementación del servicio se ha realizado utilizando el lenguaje C#, y se ha estructurado en dos proyectos: uno basado en ASP.NET y otro en WCF (Windows Communication Foundation). El código desarrollado incluye las siguientes características fundamentales:

- **Uso de Expresiones Regulares:** Aplicación de patrones de caracteres para verificar la validez del formato del número ingresado por el usuario.
- **Programación orientada a objetos.**
- **División y concatenación:** El proceso de dividir el número objetivo en grupos de tres dígitos para obtener resultados parciales y, posteriormente, combinar estos fragmentos.
- **Concurrencia:** realización de múltiples conversiones en simultáneo, lo que aumenta la velocidad de obtención de resultados.
- **Optimización del rendimiento mediante el empleo de la clase StringBuilder:** Utilización de instancias de esta clase en sustitución de objetos de tipo String en situaciones en las que se efectúan múltiples y repetidas modificaciones en cadenas de texto.
- **Sólida internacionalización del servicio,** utilizando la clase CultureInfo para adaptarse a diferentes idiomas y regiones. También sigue buenas prácticas al utilizar recursos globales para gestionar textos en distintos idiomas, lo que hace que el servicio sea multilingüe y apto para usuarios de todo el mundo.

5.1 Expresiones regulares.

Antes de iniciar las conversiones numéricas, es crucial validar la precisión del formato del número. Una manera eficiente de lograrlo, sin extender el código innecesariamente, es empleando expresiones regulares.

Una expresión regular es un patrón de búsqueda compuesto por una secuencia de caracteres que se utiliza principalmente para buscar y realizar operaciones de sustitución en cadenas de texto. Esta secuencia puede incluir literales de caracteres, operadores y estructuras, y se utiliza para identificar patrones específicos en los textos.

La lista de caracteres que se pueden emplear en una expresión regular es amplia y no es posible enumerarla completamente en este documento. Sin embargo, podemos analizar y explicar las expresiones regulares específicas que se utilizan para validar el formato del número ingresado por el usuario.

La clase "Comprobar" utiliza cinco patrones distintos para llevar a cabo esta validación, cada uno destinado a verificar un tipo específico de número. Estos patrones incluyen cardinales con separadores (puntos o espacios en blanco como separadores), formato plano de números, números fraccionarios, números decimales y la comprobación del tamaño máximo del número. A continuación, se describen estos patrones en detalle:

- Cardinales con separadores:

$\backslash d\{1,3\}\backslash d\{1,3\}(\backslash .\backslash d\{3}\backslash \backslash s\backslash d\{3})^*$

Instrucciones	Opcionales	Descripción
$\backslash d\{1,3\}$	No	Coincidencia de 1 a 3 dígitos.
$\backslash d\{1,3\}$	No	Coincidencia de 1 a 3 dígitos.
$(\backslash .\backslash d\{3}\backslash \backslash s\backslash d\{3})^*$	Sí	Opción 1: `.` seguido de 3 dígitos.
		Opción 2: Espacio en blanco seguido de 3 dígitos.

Tabla 16 Expresiones regulares - Cardinales con separadores

- Formato plano:

$^{\wedge}[0-9]^*\$$

Caracteres	Opcionales	Descripción
$^{\wedge}$	No	Coincide con el inicio de la cadena.
$[0-9]^*$	No	Coincide con uno o más dígitos (0 al 9).
$\$$	No	Coincide con el final de la cadena.

Tabla 17 Expresiones regulares - Formato plano

- Números fraccionarios:

$^{\wedge}d+\backslash d+\$$

Caracteres	Opcionales	Descripción
$^{\wedge}$	No	Coincide con el inicio de la cadena.
$\backslash d+$	No	Coincide con uno o más dígitos numéricos (del 0 al 9).
\backslash	No	Coincide con el carácter de barra inclinada ("/").
$\backslash d+$	No	Coincide con uno o más dígitos numéricos (del 0 al 9).
$\$$	No	Coincide con el final de la cadena.

Tabla 18 Expresiones regulares - Números fraccionarios

- Números decimales:

$^{\wedge}d+,\backslash d+\$$

Caracteres	Opcionales	Descripción
$^{\wedge}$	No	Coincide con el inicio de la cadena.
$\backslash d+$	No	Coincide con uno o más dígitos numéricos (del 0 al 9).
$,$	No	Coincide con el carácter de coma (",").

\d+	No	Coincide con uno o más dígitos numéricos (del 0 al 9).
\$	No	Coincide con el final de la cadena.

Tabla 19 Expresiones regulares - Números decimales

- Tamaño máximo del número:

$^{[0-9]\{1,30\}}$

Caracteres	Opcionales	Descripción
^	No	Coincide con el inicio de la cadena.
[0-9]	No	Coincide con un dígito numérico (del 0 al 9).
{1,30}	No	Indica que el dígito numérico anterior (el rango [0-9]) debe repetirse de 1 a 30 veces.
\$	No	Coincide con el final de la cadena.

Tabla 20 Expresiones regulares - Tamaño máximo del número

Si el número satisface cualquiera de los patrones mencionados anteriormente, junto con la última verificación que evalúa su tamaño, se considera válido. En este punto, se procede a descomponer el número en su parte entera y parte decimal (si es un número decimal) o en numerador y denominador (si es un número fraccionario).

5.2 Programación orientada a objetos.

El diseño de la clase Conversion refleja claramente la aplicación de los principios de la programación orientada a objetos (POO). En este contexto, se implementan métodos y propiedades que encapsulan la funcionalidad y los datos relacionados con la traducción de números a texto en diferentes contextos, como cardinales, ordinales, fraccionarios, entre otros.

Además de los métodos y propiedades, se utiliza una técnica de anidamiento de clases, donde la clase principal Conversion contiene una subclase denominada Opcion. Esta estructura jerárquica permite organizar de manera estructurada los diferentes aspectos de la traducción y facilita la gestión de las opciones relacionadas con la misma.

5.3 Concurrencia.

Dentro del contexto de nuestro servicio web, es importante destacar que puede haber momentos en los que experimentemos una alta demanda de clientes que requieren la conversión de números a texto. Ante esta situación, la eficiencia en la ejecución de estas conversiones se convierte en un factor crítico.

Para abordar este desafío, hemos adoptado una estrategia que hace un uso óptimo de los recursos disponibles y garantiza un tiempo de respuesta rápido. En este sentido, hemos implementado el uso de Parallel.Invoke, una funcionalidad que forma parte de la programación concurrente. A continuación, explicaremos por qué esta elección es óptima:

1. **Ejecución Concurrente:** El método `Parallel.Invoke` nos permite ejecutar varias tareas de conversión simultáneamente. En lugar de procesar las solicitudes secuencialmente, esta función divide el trabajo en operaciones independientes que pueden ejecutarse al mismo tiempo. Esto significa que, si un cliente solicita la conversión de un número, podemos atender su solicitud al mismo tiempo que otras solicitudes, en paralelo. Como resultado, se reduce drásticamente el tiempo de respuesta global, lo que lleva a una experiencia más rápida y eficiente para los usuarios.
2. **Aprovechamiento de Recursos:** El uso de `Parallel.Invoke` permite aprovechar al máximo los recursos del sistema. Dado que las tareas se ejecutan de manera simultánea, se distribuye equitativamente la carga de trabajo en los núcleos de la CPU disponibles. Esto es esencial para garantizar que el sistema funcione de manera eficiente, especialmente en situaciones de alta demanda.
3. **Escalabilidad:** La implementación de `Parallel.Invoke` asegura que nuestro servicio sea escalable. A medida que la demanda de clientes aumenta, podemos agregar más recursos, como servidores adicionales, para manejar un mayor número de tareas concurrentes. Esta escalabilidad es fundamental para garantizar que el servicio siga siendo eficiente a medida que crece.
4. **Experiencia del Usuario:** Finalmente, el beneficio más evidente es la mejora en la experiencia del usuario. Gracias a la ejecución concurrente, los clientes no tienen que esperar largos periodos de tiempo para que sus solicitudes se completen. Esto se traduce en una experiencia satisfactoria y ágil, lo que es esencial para mantener a los usuarios comprometidos y satisfechos con nuestro servicio.

5.4 División y concatenación

La "División y concatenación" es una estrategia fundamental en la conversión de números a texto, especialmente cuando se trata de números largos. Esta técnica se basa en descomponer el número en partes más pequeñas, procesar cada parte por separado y luego combinar los resultados para obtener la conversión completa del número.

Imagina que tienes un número largo, por ejemplo, un número con muchos dígitos. La tarea de convertir este número a su representación en palabras puede volverse complicada y propensa a errores si se aborda de manera directa. Es aquí donde entra en juego la estrategia de "División y concatenación". La "División" implica descomponer el número en fragmentos más manejables. Usualmente, se divide el número en grupos de tres dígitos cada uno. Esto facilita el procesamiento, ya que cada grupo más pequeño se convierte por separado en su forma textual. Por ejemplo, el número 123456789 se dividiría en tres grupos: "123", "456", y "789". Una vez que cada grupo se ha convertido en su representación de texto, se procede a la "concatenación". Esto significa unir los resultados de cada grupo para obtener la conversión completa del número. En nuestro ejemplo, los resultados parciales se combinarían para formar la representación textual completa del número.

El procedimiento genérico que se sigue en estas tareas se puede resumir en los siguientes pasos:

1. **Alineación a Múltiplo de 3:** Para facilitar la división en grupos de tres dígitos, el código agrega ceros a la izquierda a la entrada si es necesario para que su longitud sea un múltiplo de 3.
2. **División en Grupos de Tres Dígitos:** Luego, divide la cadena de entrada en bloques de tres dígitos cada uno. Esto se hace recorriendo la cadena y tomando subcadenas de tres en tres.
3. **Traducción de Bloques:** A continuación, comienza a traducir cada bloque de tres dígitos por separado. Este es el proceso de "División". Cada bloque se procesa utilizando el método "TraducirBloque", que se encarga de convertir los dígitos en su representación textual.
4. **Composición de Resultados Parciales:** Después de traducir cada bloque, los resultados parciales se combinan para formar la traducción completa del número. Esto es la "concatenación". Se agrega la descripción de los grupos de tres dígitos (miles, millones, etc.) según corresponda.
5. **Retorno de Resultado:** Finalmente, el resultado se devuelve como una cadena que representa la forma cardinal (u ordinal, multiplicativa, etc.) del número ingresado.

Para una comprensión más clara de estos pasos, consideremos un ejemplo, Utilicemos uno de los números que vimos en el apartado sobre la clase conversión

789.457.175.917

- Paso 1: Añadir ceros a la izquierda, si es necesario, para que la longitud sea un múltiplo de 3.
- Paso 2: Se divide el número en grupos de tres dígitos, y cada grupo se agrega a una lista llamada "bloques" y se procede a la traducción del bloque.
- Paso 3: Los resultados parciales obtenidos de la traducción de cada bloque de tres dígitos se combinan para formar la traducción completa del número. Es importante destacar que los resultados parciales se combinan en el orden correcto, teniendo en cuenta la posición y magnitud de cada bloque dentro del número original. Cada iteración se representa de la siguiente manera:

Iteración	Grupo	Resultado
1	917	• novecientos dezasete
2	175	• cento setenta e cinco mil novecientos dezasete
3	457	• catrocientos cincuenta e sete millóns cento setenta e cinco mil novecientos dezasete
4	789	• setecientos oitenta e nove mil millón catrocientos cincuenta e sete millón cento setenta e cinco mil novecientos dezasete

Tabla 21 División y concatenación

5.5 Optimización del rendimiento mediante el empleo de la clase StringBuilder

Los objetos de tipo String son invariables, lo que significa que cada vez que se realiza una operación en una cadena, se crea un nuevo objeto de cadena en la memoria, lo que requiere asignar un nuevo espacio de memoria para ese objeto. Esta situación puede llevar a una disminución en el rendimiento en situaciones en las que sea necesario realizar múltiples modificaciones en una cadena debido a la sobrecarga asociada a la creación de nuevos objetos String.

Por su parte, la clase StringBuilder ofrece la capacidad de modificar una cadena sin necesidad de crear un nuevo objeto, lo que puede llevar a mejoras significativas en el rendimiento, especialmente cuando se requiere la concatenación de múltiples cadenas dentro de un bucle u otras operaciones similares. Por lo tanto, en nuestro caso, donde estamos realizando concatenaciones repetidas, es altamente recomendable emplear objetos de la clase StringBuilder para garantizar un rendimiento óptimo.

En nuestro escenario de servicio web, donde la eficiencia y el rendimiento son críticos, la elección de objetos StringBuilder para realizar concatenaciones repetitivas es una práctica recomendable que contribuirá a garantizar un funcionamiento óptimo de nuestra aplicación.

5.6 Internacionalización del servicio

La internacionalización es un aspecto fundamental en el desarrollo de cualquier servicio en línea, y su relevancia se hace aún más evidente en el contexto de un proyecto que involucra varios

idiomas, como español, inglés y gallego, como es el caso de nuestro servicio de traducción de números a texto en gallego. A continuación, se detallan las razones clave por las cuales la internacionalización desempeña un papel crucial en este proyecto:

1. **Audiencia Multilingüe:**

Nuestro servicio está diseñado para ser accesible y útil para usuarios que hablan diferentes idiomas, incluyendo español, inglés y gallego. Al ofrecer soporte para múltiples idiomas, podemos llegar a una audiencia más amplia y diversa.

2. **Respeto a la Preferencia del Usuario:**

La internacionalización permite que los usuarios elijan su idioma preferido al interactuar con nuestro servicio. Esto demuestra respeto por la diversidad lingüística y cultural de nuestros usuarios, mejorando así su experiencia.

3. **Cumplimiento Legal y Normativo:**

En algunas regiones, existen regulaciones que exigen que los servicios en línea estén disponibles en varios idiomas. La internacionalización garantiza que cumplamos con estas regulaciones y evitemos problemas legales.

4. **Imagen Profesional:**

La internacionalización refleja un alto nivel de profesionalismo y compromiso con la calidad. Esto contribuye a que nuestro servicio sea percibido como confiable y serio.

La internacionalización se logra utilizando recursos globales para proporcionar traducciones de las salidas del servicio, tales como los tipos de números en gallego ("Cardinal", "Ordinal", etc.), ejemplos, notas, referencias y otros elementos.

Los archivos de recursos globales se crean generalmente con una extensión ".resx" y contienen pares clave-valor. Cada par clave-valor representa una cadena en un idioma específico. Desde el lado de WCF, hemos creado los siguientes recursos globales:

Español

Nombre	Valor
Adjetivo	Adjetivo
Cardinal	Cardinal
Decimal	Decimal
Expresión con adjetivo	Expresión con adjetivo
Expresión con verbo	Expresión con verbo
Fraccionario	Fraccionario
Multiplicativo	Multiplicativo
NotaCardinal1	Cento, millón, billón, trillón, aunque expresan número, funcionan como sustantivos, igual que ducia, decena, centena.
NotaCardinal2	Las antiguas formas duzentos y trezentos en el lenguaje moderno han sido reemplazadas por las más transparentes douscentos y trescentos.
NotaCardinal3	Sólo admiten variación de género los cardinales terminados en un, dous y -centos (un/unha, dous/dúas, douscentos/as, trescentos/as, etc.).
NotaCardinal4	Los números mayores que el cuadrillón no están recogidos en el Diccionario de la RAG
NotaCardinal5	Cuando los cardinales son sustantivos son siempre masculinos, excepto aquellas formas que se refieren a las horas (la una, las dos, ...).
NotaFraccionario1	A partir de 1/11 (y exceptuados 1/100, 1/1000, 1/1.000.000...) el partitivo se hace añadiendo el sufijo -avo al cardinal: (1/16) un dezaseisavo, (2/29)
NotaFraccionario2	Todas estas formas admiten morfemas de género y número, que en las formas compuestas sólo se adjuntan al último elemento (oitavo, -a).
NotaFraccionario3	Generalmente, los fraccionarios femeninos son adjetivos y los masculinos sustantivos.
NotaMultiplicativo1	A partir de 11 (y excepto el 100) e forman mediante el cardinal correspondiente seguido de la palabra veces.
NotaMultiplicativo2	Como sustantivos son siempre masculinos.
NotaMultiplicativo3	Los números del 2 al 10, y el 100 tiene forma femenina.
NotaOrdinal1	Todas estas formas admiten morfemas de género y número, que en las formas compuestas sólo se adjuntan al último elemento (décimo
NotaOrdinal2	Los ordinales 11 y 12 admiten dos variantes undécimo y décimo primero, duodécimo y décimo segundo.
NotaOrdinal3	Generalmente son adjetivos, y pueden ir antepuestos o pospuestos al sustantivo, siendo más frecuente su aparición por delante.
Ordinal	Ordinal
Pronombre	Pronombre
Sintagma nominal	Sintagma nominal
Sustantivo	Sustantivo
TitEjemplos	Ejemplos
TitNotas	Notas
TitOpciones	Todas las opciones y funciones
TitOpcionesCardinal1	Sustantivo, adjetivo y pronombre
TitOpcionesCardinal2	Forma femenina
TitOpcionesDecimal	Expresión
TitOpcionesFraccionario1	Sustantivo masculino
TitOpcionesFraccionario2	Adjetivo y pronombre femenino
TitOpcionesMultiplicativo1	Expresión con adjetivo
TitOpcionesMultiplicativo2	Expresión con verbo
TitOpcionesMultiplicativo3	Sustantivo masculino y adjetivo
TitOpcionesMultiplicativo4	Verbo
TitOpcionesMultiplicativo5	Forma femenina

Ilustración 14 WCF- Recursos-Español

Inglés

Nombre	Valor
Adjetivo	Adjective
Cardinal	Cardinal
Decimal	Decimal
Expresión con adjetivo	Expression with an adjective
Expresión con verbo	Expression with a verb
Fraccionario	Fractional
Multiplicativo	Multiplicative
NotaCardinal1	Hundred, million, billion, trillion. Although they express numbers, they function as nouns, just like dozen, ten,
NotaCardinal2	The ancient forms "duzentos" and "trezentos" have been replaced in modern language by the more transparent
NotaCardinal3	Only cardinals ending in "un," "dous," and "-centos" admit gender variation (un/una, dous/dúas, douscentos/as,
NotaCardinal4	Numbers greater than a quadrillion are not included in the RAG Dictionary.
NotaCardinal5	When cardinals act as nouns, they are always masculine, except for those forms that refer to hours (a unha, as
NotaFraccionario1	Starting from 1/11 (except for 1/100, 1/1000, 1/1,000,000...), the partitive is formed by adding the suffix "-avo" to
NotaFraccionario2	All these forms admit gender and number morphemes, which in compound forms are only attached to the last
NotaFraccionario3	Generally, feminine fractionals are adjectives, and masculine ones are nouns.
NotaMultiplicativo1	Starting from 11 (except for 100), they are formed by using the corresponding cardinal followed by the word
NotaMultiplicativo2	As nouns, they are always masculine.
NotaMultiplicativo3	Numbers from 2 to 10 and number 100 have a feminine form.
NotaOrdinal1	All these forms admit gender and number morphemes, which in compound forms are only attached to the last
NotaOrdinal2	The ordinals 11 and 12 admit two variants: "undécimo" and "décimo primeiro," "duodécimo" and "décimo
NotaOrdinal3	They are generally adjectives and can be placed before or after the noun, the former being more frequent.
Ordinal	Ordinal
Pronombre	Pronoun
Sintagma nominal	Nominal phrase
Sustantivo	Noun
TitEjemplos	Examples
TitNotas	Notes
TitOpciones	All options and functions
TitOpcionesCardinal1	Noun, adjective, and pronoun
TitOpcionesCardinal2	Feminine form
TitOpcionesFraccionario1	Masculine noun
TitOpcionesFraccionario2	Feminine adjective and pronoun
TitOpcionesMultiplicativo1	Expression with an adjective
TitOpcionesMultiplicativo2	Expression with a verb
TitOpcionesMultiplicativo3	Masculine noun and adjective
TitOpcionesMultiplicativo4	Verb
TitOpcionesMultiplicativo5	Feminine form
TitOpcionesOrdinal1	Adjective and pronoun

Ilustración 15 WCF- Recursos-Inglés

Gallego

Nombre	Valor
Adjetivo	Adxectivo
Cardinal	Cardinal
Decimal	Decimal
Expresión con adjetivo	Expresión con adxectivo
Expresión con verbo	Expresión con verbo
Fraccionario	Fraccionario
Multiplicativo	Multiplicativo
NotaCardinal1	Cento, millón, billón, trillón, aínda que expresan número, funcionan como substantivos, igual que ducia, decena,
NotaCardinal2	As formas antigas duzentos e trezentos na lingua moderna foron substituídas polas máis transparentes
NotaCardinal3	Só admiten variación de xénero os cardinais rematados en -un -dous e -centos (un/unha, dous/dúas,
NotaCardinal4	Os números maiores que o cuatrillón non están recollidos no dicionario da RAG.
NotaCardinal5	Cando os cardinais son substantivos son sempre masculinos, agás aquelas formas que se refiren ás horas(a unha,
NotaFraccionario1	A partir de 1/11 (e exceptuados 1/100, 1/1000, 1/1.000.000...) o partitivo faise engadindo ao cardinal o sufixo -avo:
NotaFraccionario2	Todas estas formas admiten morfemas de xénero e número, que nas formas compostas só se unen ao último
NotaFraccionario3	Xeralmente os fraccionarios femininos son adxectivos e os masculinos substantivos.
NotaMultiplicativo1	A partir de 11 (e exceptuado 100) fórmanse mediante o cardinal correspondente seguido da palabra veces.
NotaMultiplicativo2	Como masculino son sempre substantivos.
NotaMultiplicativo3	Os números do 2 ao 10, e o 100 ten forma feminina
NotaOrdinal1	Todas estas formas admiten morfemas de xénero e número, que nas formas compostas só se unen ao último
NotaOrdinal2	Os ordinais 11 e 12 admiten dúas variantes, undécimo e décimo primeiro, duodécimo e décimo segundo.
NotaOrdinal3	Xeralmente son adxectivos, e poden ir antepostos ou pospostos ao substantivo, sendo máis frecuente a súa
Ordinal	Ordinal
Pronombre	Pronome
Sintagma nominal	Sintagma nominal
Sustantivo	Substantivo
TitEjemplos	Exemplos
TitNotas	Notas
TitOpciones	Todas as opcións e funcións
TitOpcionesCardinal1	Substantivo, adxectivo e pronome
TitOpcionesCardinal2	Forma feminina
TitOpcionesFraccionario1	Substantivo masculino
TitOpcionesFraccionario2	Adxectivo e pronome feminino
TitOpcionesMultiplicativo1	Expresión con adxectivo
TitOpcionesMultiplicativo2	Expresión con verbo
TitOpcionesMultiplicativo3	Substantivo masculino e adxectivo
TitOpcionesMultiplicativo4	Verbo
TitOpcionesMultiplicativo5	Forma feminina
TitOpcionesOrdinal1	Adxectivo e Pronome

Ilustración 16 WCF- Recursos-Gallego

Para ASP.NET, hemos utilizado los siguientes recursos globales:

Español

Nombre	Valor
InicioCabecera	Conversor de número a texto en gallego.
listaError1	Puede que el número o la parte decimal sea demasiado grande.
listaError2	NO escriba el número con letras
listaError3	Escriba un número sin coma ni punto para los miles o millones: 13289 53625999567
listaError4	Los miles o millones se pueden separar con un espacio en blanco: 13 289 53 625 999 567
listaError5	Los miles o millones se pueden separar con un punto: 13.289 53.625.999.567
listaError6	Para escribir fracciones use la barra: 3/4 78/125 30/452 123/125
listaError7	Para escribir decimales use la coma: 3,4 78,125 30,452 123,125
placeholder	Escriba un número con cifras
textInicial1	Se puede convertir números a letras en cardinal, números en letras ordinal, números en letras decimal, números en
textInicial2	Escriba un número sin coma ni punto para los miles o millones. Los miles o millones se pueden separar con un
titEjemplos	Prueba los ejemplos
titError	Los formatos admitidos para el gallego son los siguientes:

Ilustración 17 ASP.NET - Recursos-Español

Inglés

Nombre	Valor
InicioCabecera	Number to text converter in Galician.
listaError1	The number or decimal part may be too large.
listaError2	DO NOT spell out numbers.
listaError3	Type in numbers without a comma or period for thousands or millions: 13289 53625999567
listaError4	Thousands or millions can be separated by a space: 13 289 53 625 999 567
listaError5	Thousands or millions can be separated by a period: 13.289 53.625.999.567
listaError6	To write fractions, use the slash symbol: 3/4 78/125 3/4 78/125
listaError7	To write decimals, use the comma: 3,4 78,125 30,452 123,125
placeholder	Type a number
textInicial1	Numbers can be converted into their spelled out cardinal, ordinal, decimal, fractional, partitive and multiplicative
textInicial2	Please write a number without commas or periods for thousands or millions. Thousands or millions can be
titEjemplos	Try the examples
titError	Supported formats for Galician are as follows:

Ilustración 18 ASP.NET - Recursos-Inglés

Gallego

Nombre	Valor
InicioCabecera	Conversor de número a texto en galego
listaError1	Pode ser que o número ou a parte decimal sexan demasiado grandes.
listaError2	NON escriba o número con letras.
listaError3	Escriba un número sen coma nin punto para os miles ou millóns: 13289 53625999567
listaError4	Os miles ou millóns poden separarse cun espazo en branco: 13 289 53 625 999 567
listaError5	Os miles ou millóns poden separarse cun punto: 13.289 53.625.999.567
listaError6	Para escribir fraccións, use a barra: 3/4 78/125 3/4 78/125
listaError7	Para escribir decimais, use a coma: 3,4 78,125 30,452 123,125
placeholder	Escribe un número con cifras
textInicial1	Pódense converter números a letras en cardinal, números a letras en decimal, números en letras ordinais, números
textInicial2	Escriba un número sen coma nin punto para os miles ou millóns. Os miles ou millóns poden separarse cun espazo
titEjemplos	Probe os exemplos
titError	Os formatos admitidos para o galego son os seguintes:

Ilustración 19 ASP.NET - Recursos-Gallego

6. CONCLUSIONES Y TRABAJOS FUTUROS

En vista de todo lo expuesto, se pueden obtener las siguientes conclusiones:

- La representación de números en palabras en gallego sigue un proceso sistemático, muy parecido al español, aunque con excepciones particulares.
- Aunque los términos utilizados para representar números conforman un conjunto finito, permiten referirse a números extremadamente grandes, lo que amplía su aplicabilidad.
- La internacionalización de esta herramienta, lograda mediante el uso de recursos globales y el soporte para varios idiomas, enfatiza su versatilidad y capacidad para satisfacer las necesidades de una audiencia diversa. Esto contribuye significativamente a su utilidad y accesibilidad en un contexto global.

La consideración de expandir esta utilidad de traducción de números a texto a otros idiomas se basa en varios factores clave. En primer lugar, el éxito anticipado de la herramienta en la conversión de números a texto en español subraya su efectividad y demanda. Esto sugiere que existe un mercado o una audiencia interesada en soluciones similares en otros idiomas.

La expansión a otros idiomas presenta varias ventajas. En un mundo cada vez más globalizado, donde las comunicaciones transcurren en múltiples lenguajes, contar con una herramienta versátil que pueda trabajar en diferentes idiomas es esencial. La aplicación de esta utilidad en gallego es solo un primer paso hacia la satisfacción de esa necesidad. Además, al adaptar la herramienta para otros idiomas, se puede llegar a un público más amplio y diverso. Cada idioma tiene sus propias particularidades y matices lingüísticos, por lo que ofrecer traducciones precisas y coherentes en varios idiomas puede aumentar su relevancia y utilidad.

En términos de aplicabilidad, la traducción de números a texto es útil en una variedad de contextos, desde aplicaciones financieras hasta educación y comunicaciones internacionales. Al expandir la herramienta a otros idiomas, se asegura que un público más amplio pueda beneficiarse de ella en diferentes situaciones y entornos culturales.

8. FUENTES DE INFORMACIÓN

- [1] Portal das Palabras. [En línea]. Disponible en: <https://portaldaspalabras.gal/> [Accedido: 10-oct-2023]
- [2] "Gallego/Gramática/Numerales." Wikilibros. [En línea]. Disponible en: <https://es.wikibooks.org/wiki/Gallego/Gram%C3%A1tica/Numerales> [Accedido: 10-oct-2023]
- [3] Academia Galega da Lingua Portuguesa. [En línea]. Disponible en: <https://academia.gal/inicio> [Accedido: 10-oct-2023]
- [4] Regex101: Online regex tester and debugger. [En línea]. Disponible en: <https://regex101.com/> [Accedido: 10-oct-2023]
- [5] "Lenguaje de expresiones regulares - Referencia rápida - .NET." Microsoft Learn. [En línea]. Disponible en: <https://learn.microsoft.com/es-es/dotnet/standard/base-types/regular-expression-language-quick-reference> [Accedido: 10-oct-2023]
- [6] "Utilizar la clase StringBuilder en .NET." Microsoft Learn. [En línea]. Disponible en: <https://learn.microsoft.com/es-es/dotnet/standard/base-types/stringbuilder> [Accedido: 10-oct-2023]
- [7] "Normas ortográficas e morfolóxicas do idioma galego." Academia Galega da Lingua Portuguesa,2012. [En línea]. Disponible en: <https://publicacions.academia.gal/index.php/rag/catalog/book/252> [Accedido: 10-oct-2023]
- [8] "CSS Introduction." w3schools.com. [En línea]. Disponible en: https://www.w3schools.com/css/css_intro.asp [Accedido: 10-oct-2023]
- [9] "Números a Texto." TúLengua. [En línea]. Disponible en: <https://tulengua.es/> [Accedido: 10-oct-2023]
- [10] "Provide resources for localization in ASP.NET Core." Microsoft Docs. [En línea]. Disponible en: <https://learn.microsoft.com/es->

[es/aspnet/core/fundamentals/localization/provide-resources?view=aspnetcore-7.0](https://learn.microsoft.com/es/aspnet/core/fundamentals/localization/provide-resources?view=aspnetcore-7.0)

[Accedido: 10-oct-2023]

- [11] "What is Windows Communication Foundation." Microsoft Docs. [En línea]. Disponible en: <https://learn.microsoft.com/en-us/dotnet/framework/wcf/whats-wcf> [Accedido: 10-oct-2023]
- [12] "ASP.NET Overview." Microsoft Docs. [En línea]. Disponible en: <https://learn.microsoft.com/en-us/aspnet/overview> [Accedido: 10-oct-2023]
- [13] "WCF Web Service Reference Guide." Microsoft Docs. [En línea]. Disponible en: <https://learn.microsoft.com/en-us/dotnet/core/additional-tools/wcf-web-service-reference-guide> [Accedido: 10-oct-2023]
- [14] "Parallel.Invoke Method." Microsoft Docs. [En línea]. Disponible en: <https://learn.microsoft.com/es-es/dotnet/api/system.threading.tasks.parallel.invoke?view=net-7.0> [Accedido: 10-oct-2023]