



ULPGC
Universidad de
Las Palmas de
Gran Canaria

Escuela de
Ingeniería Informática



Conversor de nomenclatura numérica en español a cifra

Titulación: Grado en Ingeniería Informática

Autor: Alexandru Robert Lazar

Tutor: D. Francisco Javier Carreras Riudavets

Fecha: 06/2022

Agradecimientos

Este trabajo va especialmente dedicado a mis familiares más cercanos, por haberme prestado su apoyo y su ayuda durante toda mi carrera académica, así como por el esfuerzo realizado para tener la oportunidad de recibir formación universitaria.

Asimismo, le agradezco enormemente a mi tutor por su apoyo y ayuda durante la realización de este trabajo.

Y, por supuesto, le agradezco también a todo el equipo docente de la Escuela de Ingeniería Informática por haber hecho posible mi formación en esta titulación.

Resumen

Este proyecto consistirá en la realización de un servicio Web que, a partir de un número escrito como texto en español, realice la conversión automáticamente al número escrito con cifras y devuelva el resultado al usuario.

Concretamente, se permitirá convertir los números que contengan hasta 122 cifras. Además, este servicio será fácilmente escalable, de tal forma que sea sencillo aumentar estos límites.

También se realizará una interfaz, que servirá, fundamentalmente, para que los usuarios puedan interactuar con el servicio Web. Por otro lado, en esta interfaz, se creará una función de “autocompletar”, con una navegación similar a la de los editores de código, que ayudará a los usuarios a evitar errores a la hora de escribir el número.

Abstract

This thesis consists of creating a Web service that allows converting automatically from numbers written as text in Spanish to digits, returning the result to the user.

More precisely, this service will allow converting numbers up to 122 digits. Moreover, this service will be designed so that it is easily scalable so increasing its limits is an easy task.

On the other side, an interface will be created, to allow user interaction with the service. This interface will also include an “autocomplete” feature that will help users avoid mistakes while inserting the number, and it will have similar navigation to the IDEs.

This thesis fits within the natural language processing area.

Índice general de contenidos

1. Introducción.....	3
1.1. Planificación del proyecto.....	4
2. Estado actual y objetivos iniciales.....	5
2.1. Análisis de aplicaciones similares.....	5
2.2. Análisis de los números en español.....	6
2.2.1. Números enteros cardinales.....	6
2.2.2. Números fraccionarios.....	8
2.2.3. Números decimales.....	10
2.3. Objetivos iniciales.....	11
3. Competencias específicas y aportaciones.....	12
3.1. Competencias específicas.....	12
3.2. Aportaciones.....	13
4. Desarrollo.....	15
4.1. Requisitos.....	15
4.2. Diseño.....	15
4.3. Implementación.....	23
4.4. Análisis de calidad.....	37
4.4.1. Servicio de conversión.....	37
4.4.2. Cliente web.....	44
4.4.3. Aplicación para realizar pruebas.....	48
5. Normativa y legislación.....	51
6. Conclusión y trabajos futuros.....	52
6.1. Conclusión.....	52
6.2. Trabajos futuros.....	52
7. Fuentes de información.....	53
8. Anexos.....	55
8.1. Anexo I – Contenidos del almacén de números.....	55
8.2. Anexo II – Contenidos del fichero de entrada para realizar las pruebas.....	62

Índice general de ilustraciones

Ilustración 1: Conexiones al servicio WCF	16
Ilustración 2: Página de presentación de la herramienta – formato escritorio.....	17
Ilustración 3: Página de presentación sobre la herramienta - formato móvil.....	18
Ilustración 4: Modal de ayuda, formato escritorio	19
Ilustración 5: Modal de ayuda, formato móvil	20
Ilustración 6: Página de conversión de textos a números, formato escritorio.....	21
Ilustración 7: Página de conversión de textos a números, formato móvil	22
Ilustración 8: Aplicación para realizar pruebas automáticas.....	23
Ilustración 9: Código del fichero "Dictionary.js"	25
Ilustración 10: Código de añadido de números al diccionario.....	27
Ilustración 11: Código que tiene la función de carga de números.	28
Ilustración 12: Código que obtiene los números similares al último.....	29
Ilustración 13: Código con la función de verificación del máximo número introducido.	30
Ilustración 14: Aproximación teórica para aplicar los filtros en los números.	31
Ilustración 15: Función de carga de números.....	33
Ilustración 16: Función para obtener la longitud de la parte decimal de un número.	35
Ilustración 17: Funciones de separación de cifras e inversión de elementos en un array.	36
Ilustración 18: Página inicial de configuración, con los datos ya introducidos	38
Ilustración 19: Segundo paso de la configuración del proyecto	39
Ilustración 20: Generación del token el proyecto.....	40
Ilustración 21: Instrucciones para ejecutar el análisis desde línea de comandos	41
Ilustración 22: Resultados del análisis en el servicio del conversor	42
Ilustración 23: Incidencias menores del servicio	43
Ilustración 24: Bug encontrado en el servicio.....	43
Ilustración 25: Resumen de los resultados del análisis en el cliente web	44
Ilustración 26: Ficheros con código duplicado.....	45
Ilustración 27: Vista parcial de los bugs de la aplicación	46
Ilustración 28: Vista parcial de las incidencias menores del código	47
Ilustración 29: Bugs en el código del fichero "About.aspx"	48
Ilustración 30: Resumen de los resultados del análisis de calidad en la aplicación para realizar pruebas	49
Ilustración 31: Incidencia menor detectada en la aplicación para hacer pruebas	49

Índice general de tablas

Tabla 1: Planificación inicial del proyecto.....	4
Tabla 2: Partes de un número entero.....	7
Tabla 3: Patrones para formar números enteros cardinales.....	7

1. Introducción

Este Trabajo de Fin de Grado se fundamenta en el hecho de que no existen conversores de nomenclatura numérica en español a cifra, al menos de acceso libre al público.

Existen numerosos documentos, sobre todo relacionados con sentencias judiciales, que contienen nomenclatura numérica (véase, como ejemplo, la Sentencia de la Sala Tercera del Tribunal Supremo relativa al recurso de casación número 3233/2011, ubicada en [1]) y que, sin embargo, no contienen su equivalente en cifras. Ello dificulta el procesamiento de este texto (por ejemplo, para fines estadísticos), dado que, como se comentaba antes, aún no hay disponibles este tipo de conversores.

A pesar de que esta herramienta no permitirá procesar números que están en un texto, sí que se podría, en un futuro, aprovechar su código para ampliar su funcionalidad, de tal forma que se puedan obtener los números directamente desde un texto.

Además, también se pretende realizar este proyecto con la finalidad de facilitar el aprendizaje del español, tanto para los niños como para las personas no hispanohablantes, puesto que contará con una función de autocompletado de números.

También se podría aprovechar, en un futuro, este código para crear una herramienta que permita hacer, por ejemplo, pruebas de conocimiento sobre los números, comprobando los resultados directamente utilizando el conversor de números. Ello resultará sencillo de hacer, puesto que el conversor de números estará ubicado en un servicio aparte de la aplicación web.

En definitiva, esta herramienta puede suponer el comienzo de un antes y un después en el tratamiento de los textos que contengan nomenclatura numérica, dadas las posibilidades que ofrecerá a la hora de convertir dichos textos y a su gran escalabilidad.

1.1. Planificación del proyecto

La planificación inicial del proyecto se ve reflejada en la siguiente tabla:

Tabla 1: Planificación inicial del proyecto

Fases	Duración estimada (en horas)	Tareas
Estudio previo / Análisis	30	Tarea 1.1: Documentación y herramientas
		Tarea 1.2: Análisis de requisitos de usuario
		Tarea 1.3: Análisis de requisitos de software
Diseño / Desarrollo / Implementación	240	Tarea 2.1: Diseño e implementación del servicio
		Tarea 2.2: Diseño e implementación del cliente web
Evaluación / Validación / Prueba	20	Tarea 3.1: Implantación
		Tarea 3.2: Pruebas de la aplicación
Documentación / Presentación	10	Tarea 4.1: Documentación y presentación

Esta planificación se ha cumplido, sin sufrir variaciones importantes.

2. Estado actual y objetivos iniciales

En este apartado, se presentan, por una parte, los progresos que ha habido a la hora de crear conversores similares al creado en este proyecto y, por otra, los objetivos que se cumplen con la realización de este.

2.1. Análisis de aplicaciones similares

Hay numerosos conversores que tratan con números, como el realizado por el IATEX (véase en [2]), pero que proporcionan la funcionalidad inversa, es decir, convertir de cifras a nomenclatura numérica. Sin embargo, a fecha de la defensa de este proyecto, no se pueden encontrar conversores cuya funcionalidad sea la vista en este, es decir, convertir de nomenclatura numérica en español a cifras.

Sí se puede encontrar un conversor que realiza esta función en inglés (véase [3]), pero cuyos límites son perceptibles, dado que solo permite convertir números del orden de los trillones, aunque permite emplear números como “dos millones de trillones”.

Además, la aplicación Microsoft Word [4] ofrece la posibilidad de convertir, mediante su función de dictado, de voz a número. Sin embargo, esta herramienta únicamente permite convertir números sencillos. Por ejemplo, al intentar convertir “dos mil con tres milésimas”, el Word da como resultado “2000,3 milésimas”. Además, al igual que en el caso anterior, esta herramienta también tiene unos límites perceptibles, dado que su límite está en el orden de los millones (ello se afirma puesto que, al intentar convertir “un billón ochocientos”, el resultado que proporciona el Word es “un billón 800”, pero, al intentar convertir “dos millones ochocientos mil treintaiséis”, su resultado es “2800036”). Nótese que los resultados obtenidos en esta parte son directamente extraídos de la función de dictado.

Por otra parte, existen aplicaciones que, haciendo uso de la inteligencia artificial, pueden deducir números a partir de ejemplos (véase [5] para un ejemplo). Sin embargo, los sistemas de inteligencia artificial habitualmente no llegan a tener una precisión del 100%, algo que sí se pretende realizar con esta herramienta, de forma algorítmica. Además, dado el gran tamaño de los números que se van a utilizar en esta herramienta, es inviable crear tantos ejemplos como para tener una gran precisión en inteligencia artificial.

Por tanto, si bien es cierto que hay numerosas aplicaciones que intentan acercarse al objetivo de esta herramienta, ninguna de ellas consigue tener un límite lo suficientemente grande de números como para que el usuario no lo pueda percibir.

Si bien es cierto que es imposible emplear todos los números posibles, dado que estos son infinitos (esta afirmación se sustenta en el teorema de Euclides [6], que afirma que existen infinitos números primos, con lo que se deduce que hay infinitos números) y, sin embargo, no existe almacenamiento infinito (por lo que tampoco existen tablas con números infinitos), sí se prevé proporcionar unos límites muy grandes, que llegarían hasta el orden de los vigintillones (10^{120}). Ello se hace así para proporcionar al usuario unos límites prácticamente inalcanzables, con la finalidad de no poder percibirlos.

Además, este límite es fácilmente ampliable, pues únicamente se necesita añadir más números en la lista de números que la aplicación utiliza. El requisito de cualquier nuevo número que se pretenda añadir

será que el resto de dividir la diferencia del orden del nuevo número y el orden del último número añadido de la lista sea cero, y que la resta entre estos órdenes sea seis (como 10^{126} , o 10^{132}).

2.2. Análisis de los números en español

Una vez vistas las aplicaciones que comparten algún símil con la que se va a realizar en este proyecto, es importante conocer cómo son los números en español para poder llevar a cabo este proyecto. Este análisis también será útil para determinar qué números será necesario testear para poder hacerlo sin incluirlos todos (lo cual sería muy complicado, dado el alcance al que se va a llegar con este proyecto), pero que sea de la manera más completa posible.

En este proyecto, se va a trabajar con tres tipos de números: cardinales, fraccionarios y decimales. La razón de no trabajar con números ordinales es que no es muy habitual que se utilicen para formar números complejos o largos.

2.2.1. Números enteros cardinales

Los números cardinales son aquellos que representan cantidades. Estos números sirven para contar cosas (se tienen **dos** manzanas, una operación dura **cuarenta** milisegundos, un día tiene **veinticuatro** horas...), pero también pueden indicar la falta de una cantidad de cosas mediante los números negativos (faltan **dos** horas para el vuelo, se necesitan **treinta y ocho** euros más para comprar ese reloj...). Estos últimos números se representan mediante la palabra “menos” delante de estos.

En el Anexo I, en el que aparecen los contenidos del fichero que sirve de almacén de números, aparece una lista exhaustiva de números cardinales que no se forman a partir de combinaciones con otros números cardinales. Concretamente, todos aquellos números cuyo valor no empiece por “/” son cardinales.

Es importante mencionar que hay algunos números que tienen variantes en femenino (por ejemplo, “doscientas”); sin embargo, estas variantes se convierten a masculino automáticamente por el aplicativo para hacer la conversión, con la finalidad de hacer este fichero más ligero, por lo que no aparecen reflejadas.

Como se comentaba anteriormente, esta lista de números no incluye aquellos que se puedan formar a partir de combinaciones con otros (por ejemplo, “treinta y tres”). Es este proyecto el que hará posible la conversión de un número así a cifras, y no tendría sentido su existencia si existiera una lista con todos los números del $-999 \cdot 10^{123}$ al $999 \cdot 10^{123}$. Además, hacer una lista así sin apoyarse de un proyecto como este tampoco tendría sentido, dado el tiempo que se tardaría en hacer una lista así y el tamaño que tendría. Es importante comentar cómo se combinan estos números para poder implementar su conversión y para conocer la profundidad que deben alcanzar las pruebas que se le pasarán al conversor al final de su desarrollo.

Las partes de un número cardinal son las siguientes (con un número de ejemplo):

Tabla 2: Partes de un número entero

...	Centenas de millones	Decenas de millones	Millones	Centenas de millar	Decenas de millar	Millares	Centena	Decena	Unidad
	1	0	9	8	3	7	2	5	9

De esta manera, un número se combina mediante la suma de sus partes. Así, en el anterior número, las partes serían:

- 100000000
- 9000000
- 800000
- 30000
- 7000
- 200
- 50
- 9

Y, si se hace la suma de todos estos números en los que se ha descompuesto, se obtiene el número original: 109837259.

A continuación, en la tabla 2, se observan algunos patrones que ayudan a formar números en español. Nótese que estos patrones no tratan de números que ya están definidos en una sola palabra (como “treintatrés”), sino que se buscan aquellos que combinen números que se formen con más de una palabra.

Además, estos patrones serán incrementales, es decir, si se quisiera obtener el número “1048”, se tendrían que mirar los tres patrones que se verán en la siguiente tabla:

Tabla 3: Patrones para formar números enteros cardinales

Rango de números	Definición de patrón para combinar
21-99	Se insertan las decenas, seguidas de un espacio, la palabra “y”, otro espacio y las unidades.
101-999	Se insertan las centenas, seguidas por los demás números, que vendrán separados de esta unidad por un espacio.
1001 en adelante	Se inserta un rango de números del 1-999, según los patrones anteriores (si fuera el caso), seguido de una unidad de mil al vigintillón (mil, millón, billón...), seguido de números adicionales, separados de esta unidad por un espacio, hasta completarlos. No se puede introducir una unidad mayor o igual al millón si ya hay una unidad menor introducida (por ejemplo, “un millón un billón” es incorrecto).

Este análisis pretende ser útil también para determinar unos casos de prueba, de tal forma que se prueben un rango de casos lo suficientemente amplio como para dar la aplicación por válida, pero sin tener que usar una gran cantidad de casos.

En el caso de los números cardinales, por tanto, el siguiente conjunto de números sería suficiente para validar su correcto funcionamiento:

- Rango 0-99: se necesitan testear todos estos números, dado que, a partir del 30, los números pueden tener variantes con dos palabras (“treinta y tres”) y con una sola palabra (“treintaitrés”).
- Números 100, 101, 110, 111, 122, 168, 198, 211, 422, 689, 712, 879, 919: Se trata de un conjunto reducido de números del rango 100-999, que prueba, fundamentalmente, que los números se combinan de manera correcta. En los cuatro primeros casos, se trata de números especialmente escogidos para ello, mientras que los demás números son aleatorios, dado que:
 - o El primer número prueba que el “cien” o “ciento” funciona correctamente.
 - o El segundo número, además, prueba la combinación de una centena con una unidad, mientras que el tercero, prueba la combinación de una centena con una decena
 - o El último número prueba la combinación de centenas con decenas y unidades.
- Números 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111, 2389, 4001, 5087, 6938, 7907, 8004, 9019: Se trata de un conjunto reducido del rango 1000-9999, que sirve para probar la combinación correcta de estos con los números siguientes (estando, para ello, los 8 primeros números) y que, además, funcione correctamente la concatenación entre un número entre 1-9 y la unidad de millar, para formar números mayores a 1999 (por ejemplo, “dos mil trescientos ochenta y nueve”).
- Números 10000, 11000, 20939, 30001, 49012, 58700, 69810, 78569, 81293, 94182: Se trata de otro conjunto, bastante más reducido, del rango 10000-99999, que sirve para probar lo mismo que en el caso anterior, para este rango.
- Números del millón al vigintillón: Se escogerán números aleatorios, cada uno con una cifra más, ya que el patrón que se sigue en este caso será siempre el mismo que en el caso de los millares. Por ejemplo, se podría comenzar por “2093129”.

Tras ver que los casos anteriores funcionan correctamente, se puede considerar que el servicio de conversión funciona de forma correcta para los números enteros cardinales.

2.2.2. Números fraccionarios

Los números fraccionarios son aquellos que determinan en cuántas partes se divide una cosa (se suelen utilizar con mayor frecuencia al hablar sobre comidas: una tarta tiene cuatro trozos, una pizza tiene ocho partes...), así como cuántas de esas partes se van a consumir (volviendo al ejemplo de las comidas, se puede decir que una persona, por ejemplo, va a comerse dos trozos de una tarta que tiene seis trozos).

Estos números también son simplificables. En el ejemplo anterior, se ve un caso de número fraccionario simplificable, puesto que es lo mismo decir que una persona puede comerse dos trozos de una tarta que tiene seis y que una persona puede comerse un trozo de una tarta que tiene tres trozos (dando por hecho que el tamaño inicial de la tarta es el mismo).

En este proyecto, no se va a trabajar con la simplificación de números, al desear tratar con los números tal cual están escritos, pero la característica de simplificación de fraccionarios es más que útil para entender cómo funcionan.

Al igual que en el caso de los números cardinales, se van a establecer algunos casos de prueba para poder comprobar que estos números funcionan de forma correcta. Para ello, se debe establecer primero un patrón que determine cómo se escriben estos números.

En cuanto a los numeradores, el patrón que se sigue es exactamente el mismo que en el caso de los números cardinales. En cambio, en cuanto a los denominadores, si bien el patrón es muy parecido, hay algunos cambios.

Los denominadores se podrían clasificar en dos tipos:

- Predefinidos: son aquellos números que ya están establecidos, y que se reflejan en el Anexo I (son aquellos cuyo valor empieza por “/”, además de todos los números a partir de “mil”, excluyendo “millardo”, a los que se añade el sufijo “-ésimo”, “-ésima”, “-ésimos”, “-ésimas”, según el numerador que esté introducido, y a los que se les eliminan las tildes, como, por ejemplo, “millonésimo”).
- Definidos mediante combinación: son aquellos números que se combinan para formar otros, y que, necesariamente, terminan en uno de los sufijos “-avo”, “-ava”, “-avos”, “-avas”, cuyo género y número también se determinan por el numerador (no se podría decir, por ejemplo, “doscientas milavo”).

En cuanto los denominadores predefinidos, se hace sencillo realizar casos de prueba, puesto que solamente se trata de probar cada uno de estos con diferentes numeradores; por tanto, no se necesitan analizar en mayor profundidad.

Sin embargo, en los denominadores definidos mediante combinación, por su naturaleza, se deberá hacer un análisis más específico sobre cómo se combinan, para después determinar algunos casos de prueba.

Realmente, los denominadores se combinan de manera igual que los numeradores; lo único que cambia es que se eliminan las tildes en algunas posiciones, el separador “y” pasa a ser “i”, y no se insertan espacios.

Por tanto, utilizando cinco casos de prueba aleatorios, se puede determinar que la aplicación funciona de forma correcta. Se proponen los siguientes números, dada su variabilidad y complejidad:

- 1087/10298371
- 491029102098/1029381023918203
- 1290381293/2
- 1/102948143181290481281
- 10238131381290318120712/120938190238214204824498

Si estos casos de prueba funcionan correctamente, se puede afirmar que la conversión de los números fraccionarios también funciona de forma correcta. Ello parte del hecho de que los números cardinales

funcionan correctamente, dado que ya se ha comprobado en el apartado anterior, con lo que hay muchos casos de prueba que ya no son necesarios, y únicamente se tiene que emplear un conjunto reducido de casos de prueba, con números grandes, que son más complejos de escribir.

2.2.3. Números decimales

Los números decimales son aquellos que representan cualquier cantidad que contiene partes menores a la unidad. Es habitual que, en ocasiones, no se necesite hacer uso de unidades enteras (por ejemplo, para pagar productos) o que el resultado de una operación no implique el uso único de los números enteros (por ejemplo, en el caso de obtener el resultado de un número fraccionario), por lo que se hace necesario el uso de este tipo de números.

Estos números se dividen en dos partes: la parte entera y la parte decimal. Para separarlas, es necesario hacer uso de una de las siguientes palabras: “con”, “coma” e “y”.

Nótese que esta última palabra puede dar lugar a alguna ambigüedad a la hora de separar números como “treinta y siete”, puesto que surge la duda acerca de si este número realmente es “37” o “30,7”. Este tipo de ambigüedad no se da al emplear la palabra “con” ni la palabra “coma”. Para este tipo de casos, se decide que se tendrá en cuenta únicamente la primera forma de expresar los números, dada su mayor frecuencia.

Además, opcionalmente, la parte decimal puede incluir una palabra, al final de esta, que define cuántas unidades decimales utiliza el número introducido. Por ejemplo, para el número “dos con una milésima”, en cifras se representaría como “2,001” y no como “2,1”.

Esta palabra puede ser una de las siguientes opciones: “décima”, “centésima”, “milésima”, así como cualquier unidad mayor a mil, según los contenidos del Anexo I (salvo “millardo”), siempre que no incluya las tildes y venga con uno de los siguientes sufijos: “-ésima” o “-ésimas”.

Al contrario de lo que ocurría con los números fraccionarios, el número de esta palabra viene determinado por la propia parte decimal (no se podría decir, por ejemplo, “dos con una milésimas” o “uno con dos centésima”). Además, como se puede observar, estos sufijos están necesariamente en género femenino. Ello ocurre porque los números decimales deben estar en femenino.

Al igual que en los dos casos anteriores, es necesario definir unos casos de prueba, y, previo a ello, hacer un breve análisis sobre la forma en la que se combinan estos números para dar lugar a un número válido.

En este caso, no es necesario adentrarse mucho en los patrones, ya que son los mismos que en el caso de los números cardinales, siempre que se tenga en cuenta, además, lo descrito anteriormente.

Por tanto, se proponen los siguientes cinco casos de prueba, completamente aleatorios, que prueban números con diferentes características y complejidades:

- 1,12398
- 12938193,12093131203
- 1,000000000001293483
- 19283,0012334412413

- 20945729084572234908573954,000000100203457280452435773

La elección de estos casos de prueba está fundamentada en el mismo supuesto que en el caso de los números fraccionarios, es decir, que los números enteros cardinales funcionan correctamente. Realmente, la creación de un número decimal consiste en unir dos números enteros mediante una coma (o punto, según el sistema de representación que se utilice) y que, en el caso de la parte decimal, se pueden emplear algunas palabras que no se utilizan en los enteros, como “centésimas”. El uso de este conjunto reducido de casos de prueba, por tanto, permite asegurar que la aplicación cumple con el requisito que obliga a que su porcentaje de fallos sea menor al 2%, al usar, mayoritariamente, números largos y que son más complejos de escribir.

2.3. Objetivos iniciales

En este Trabajo de Fin de Grado, como objetivo principal, se desea realizar una aplicación web que permita convertir números escritos como texto a cifras. Los tipos de números que se admitirán serán los fraccionarios, los cardinales y los decimales.

Ello no se podrá hacer utilizando cálculos matemáticos, dadas las limitaciones de tamaño de los tipos numéricos (el más grande, que sería el “long” sin signo, apenas supera los dieciocho trillones [7], que es mucho menor que el límite que se propone como objetivo en esta herramienta: novecientos noventa y nueve viginitillones). Por tanto, se tratarán directamente cadenas de caracteres.

Se incluirá también una herramienta de autocompletado, con dos finalidades principales. Concretamente, se desea evitar que los usuarios cometan errores escribiendo los números, facilitando su aprendizaje del idioma y, con ello, se evita que el conversor tenga errores a la hora de convertir. Para esta herramienta, se utilizará un objeto de JavaScript, que simulará un diccionario y cuya complejidad será de $O(1)$.

Además, para facilitar la visualización del número, el conjunto de cifras generadas tendrá separadores. Concretamente, cada 3 cifras, como es habitual, se introducirá un espacio, salvo al tener exactamente 4 cifras, donde, según la RAE, es incorrecto introducir un espacio [8]. Téngase en cuenta que la separación de números en las partes decimales de los números va de izquierda a derecha (en las cifras que se encuentren detrás del separador de decimales), mientras que, en las partes enteras de los números y en los denominadores de los fraccionarios, la separación de números se hará de derecha a izquierda.

Por último, cabe mencionar que esta aplicación se dividirá en tres partes, para hacerla lo más modular posible. La primera parte consiste en un servicio WCF [9] que será el encargado de hacer la conversión propiamente dicha de los números; la segunda, será el cliente web desde el que se podrá utilizar el servicio; y, por último, se creará una aplicación de tipo WinForms para comprobar el funcionamiento correcto del servicio.

Tras completar esta aplicación, sus dos primeras partes se desplegarán en dos servidores remotos, con la finalidad de que los usuarios puedan aprovecharla. De esta forma, aquellas personas que necesiten utilizar únicamente el servicio WCF podrán hacerlo sin necesidad de utilizar también el cliente web.

3. Competencias específicas y aportaciones

En este apartado, se especificarán las competencias del Grado en Ingeniería Informática que están aparejadas a este proyecto, junto a una justificación que explique por qué existe un enlace entre ambos, y, a continuación, se presentarán las aportaciones que este proyecto hará a la sociedad.

3.1. Competencias específicas

A continuación, se enumeran las competencias del Grado en Ingeniería Informática asociadas a este Trabajo de Fin de Título, y se motiva, en cada caso, por qué se considera que se cumplen dichas competencias.

T1 – “Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto, de acuerdo con los conocimientos adquiridos según lo establecido en apartado 5 de la resolución indicada, la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas.”

Esta competencia se cumple, puesto que la realización de este Trabajo de Fin de Grado ha supuesto realizar todas las actividades mencionadas.

T2 – “Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática, de acuerdo con los conocimientos adquiridos según lo establecido en apartado 5 de la resolución indicada.”

Esta competencia se cumple, puesto que la realización de este proyecto ha supuesto la dirección de sus actividades.

T3 – “Capacidad para diseñar, desarrollar, evaluar y asegurar la accesibilidad, ergonomía, usabilidad y seguridad de los sistemas, servicios y aplicaciones informáticas, así como de la información que gestionan.”

Puesto que una parte de este proyecto trata de realizar un sitio web, esta competencia necesariamente se cumple.

T8 – “Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.”

Puesto que este proyecto es algo novedoso y, además, se utilizan lenguajes de programación vistos durante algunas asignaturas, esta competencia se cumple.

T9 – “Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática.”

Al tratarse de un proyecto con un único autor, esta competencia se cumple, puesto que se han tenido que resolver situaciones novedosas de forma autónoma y, además, durante la redacción de la presente memoria, se han tenido que transmitir los conocimientos acerca del proyecto.

T11 – “Capacidad para analizar y valorar el impacto social y medioambiental de las soluciones técnicas, comprendiendo la responsabilidad ética y profesional de la actividad del Ingeniero Técnico en Informática.”

Esta competencia se cumple parcialmente, puesto que se ha tenido que realizar un análisis de impacto social (desde el punto de vista de las aportaciones que puede dar este proyecto) durante la redacción de la presente memoria, pero no se ha realizado un análisis de impacto medioambiental.

3.2. Aportaciones

Las aportaciones que este proyecto dará al conjunto de la sociedad se pueden analizar teniendo en cuenta múltiples entornos, que se van a comentar a continuación:

- Educativo: En este aspecto, es necesario tener en cuenta dos partes: los docentes y los alumnos.
 - o Docentes: Las aportaciones que este proyecto realizará a los docentes consistirán, fundamentalmente, en facilitar la creación de cursos que contengan temario relativo a los números en español. Por ejemplo, se podría crear un curso de aprendizaje del español básico de forma mucho más sencilla, ya que se podrían incluir cuestionarios que contengan preguntas sobre estos números. De esta forma, se podría aprovechar el servicio, haciendo la conversión en tiempo real de los números que los estudiantes introduzcan en estos cuestionarios.
 - o Estudiantes: Mediante el uso de este conversor, todas aquellas personas que deseen aprender los números en español podrán hacerlo de una forma mucho más interactiva que siguiendo un curso convencional. Para ello, podrán hacer uso del cliente web, ya que también tendrá la función de autocompletado.
- Investigativo: La principal aportación que esta herramienta dará a los investigadores será que se les facilitará muchísimo crear herramientas que tengan que ver con el tratamiento de números escritos como texto. Un ejemplo de estas herramientas está descrito en el apartado de trabajos futuros.
- Judicial/Administrativo: En el campo de la justicia o la administración, los números, en muchas ocasiones, no suelen incluir su equivalente en cifras, como se ha visto en la introducción de este proyecto. Este conversor, por tanto, permite el procesamiento de estos números, al devolverlos como cifras.

Este proyecto no solamente aporta a la sociedad en su conjunto, sino también al autor de esta obra. Mediante la realización de este proyecto, se han adquirido algunas competencias adicionales a las mencionadas anteriormente.

Estas competencias son, a grandes rasgos, las siguientes:

- Capacidad de medir la calidad de un software mediante una herramienta automatizada.
- Capacidad de programar en el framework .NET, que no se ha visto durante las asignaturas de la titulación.
- Capacidad de realizar a cabo acciones de investigación, como pueden ser análisis teóricos.

- Capacidad de desplegar aplicaciones en entornos remotos.
- Capacidad para utilizar las distintas funcionalidades de la herramienta Visual Studio Enterprise de forma eficiente.

Una vez vistas estas aportaciones, se va a proceder, a continuación, al desarrollo del proyecto.

4. Desarrollo

En este apartado, se detallará todo el proceso de desarrollo de la aplicación, desde sus requisitos hasta la implementación.

4.1. Requisitos

A continuación, se enumeran los requisitos que debe cumplir esta aplicación web. Primeramente, se detallan los requisitos funcionales y a continuación los requisitos no funcionales.

En cuanto a los requisitos funcionales, destacan tres requisitos fundamentales.

En primer lugar, la tasa de errores en las conversiones no debe superar el 2%, aunque, idealmente, esta tasa se ubicará en un 0%. Se deja esta tasa de errores en un 2%, por si ocurre el caso en el que pudiera existir algún caso extremo que no se haya detectado durante la realización del proyecto. Este requisito quedará cubierto con la realización de las pruebas que se harán más adelante.

En segundo lugar, el tiempo de respuesta de la función de autocompletado, igual que el de la conversión, no debe ser mayor a un segundo, para que el usuario no note mucho retardo.

Y, para finalizar con los requisitos funcionales, el tiempo de respuesta del conversor no debe ser mayor a un segundo. Además, la conversión deberá realizarse sin recargar la página.

A continuación, se detallan los requisitos no funcionales.

En primer lugar, este proyecto deberá desarrollarse utilizando el lenguaje C#, con ASP.NET. Ello se debe a que este proyecto se realiza para el IATEX, por tanto, se necesita que el lenguaje coincida con el de los demás proyectos de dicha entidad.

Para desarrollar la herramienta, es fundamental contar con el programa Visual Studio 2019, dado el conjunto de herramientas que integra, tanto para facilitar el desarrollo como su ejecución, además de permitir, de una forma muy sencilla, realizar el despliegue del cliente web y del servicio. En este caso, se va a utilizar la edición Enterprise de esta [10]. Además, para que se pueda desplegar en Internet, es necesario que se cuente con un servidor con IIS [11], o bien tener una suscripción a Azure.

En segundo lugar, la herramienta deberá ser capaz de convertir números cardinales, fraccionarios y decimales. Además, el límite máximo de número que se puede convertir es de 999 vigintillones.

4.2. Diseño

El diseño de este proyecto se realizará de tal forma que tanto el aplicativo para comprobar el correcto funcionamiento del conversor como el cliente web se conecten a un servicio SOAP [12], de tipo WCF. La conexión se hará, fundamentalmente, de la manera que se podrá observar en la siguiente ilustración:

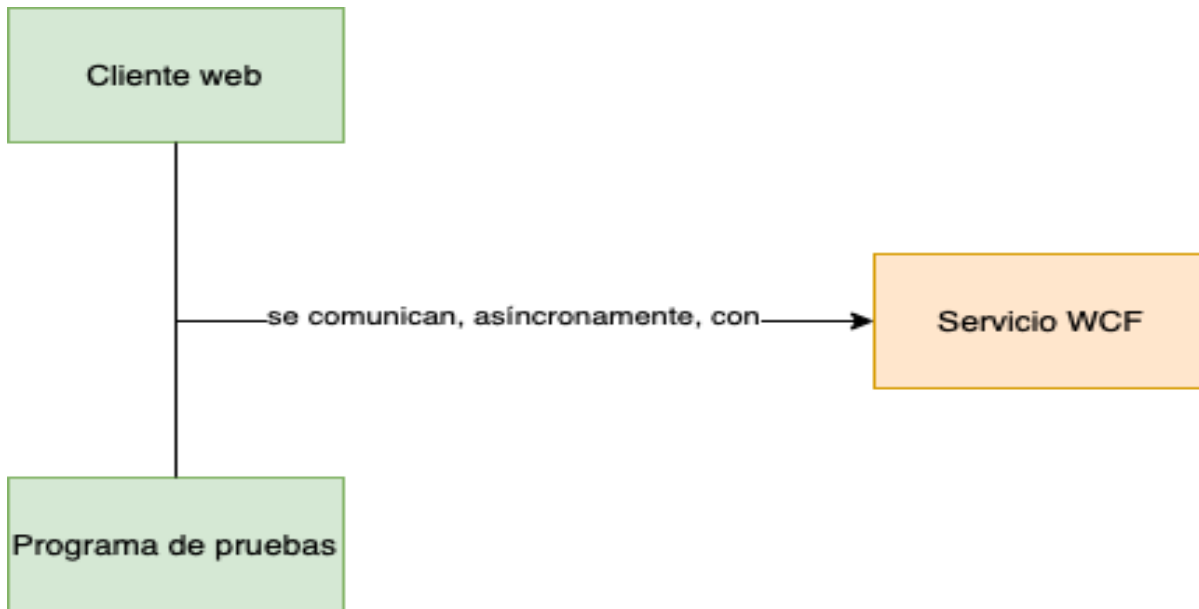


Ilustración 1: Conexiones al servicio WCF

Los clientes, en este caso, lo único que hacen es llamar a las funciones del servicio que se exportan (“GetNumbers()” y “ComputeNumber()”), para obtener o calcular números, respectivamente.

De esta manera, no hay que copiar dos veces el código relativo a la conversión en sí, sino que basta con hacerlo una vez, en el servicio, para que múltiples aplicaciones puedan aprovecharlo. De esta forma, queda demostrada la utilidad de utilizar un servicio aparte.

Ahora, se explicará el diseño y el desarrollo de las distintas partes vistas anteriormente.

En cuanto al cliente web, su diseño será muy sencillo. Téngase en cuenta que esta aplicación está destinada al aprendizaje de los números, con lo cual es muy probable que sea utilizada también por niños; por lo tanto, su diseño no debe ser complejo. Para facilitar, además, que esta aplicación pueda ser utilizada en todos lados, su diseño será responsive.

Este sitio web se dividirá en 3 partes fundamentales, que se comentarán a continuación.

La primera de ellas consiste en la página que contiene el conversor propiamente dicho, y que será una página sencilla que tendrá una caja de texto para insertar el número, una caja que mostrará las opciones del autocompletar (según el texto que se vaya insertando), y un botón de “Enviar”. Además, tras pulsar en este botón, aparecerá una cajita donde aparecerá el resultado. La obtención del resultado se realizará de tal forma que no se recargue la página para mostrarlo. Para ello, se utilizará un UpdatePanel [13].

La segunda parte de esta página será una breve descripción de la herramienta: en esta se comentará qué se pretende conseguir con la herramienta y en qué contexto se ha creado.

Por último, habrá un modal de ayuda que se abrirá al pulsar un botón con un “?”, y que explicará brevemente cómo se utiliza la herramienta. Además, proporcionará algunos ejemplos para poder probarla de forma rápida, puesto que, al hacer clic en alguno de ellos, automáticamente el campo de texto se rellenará con ese número de ejemplo.

A continuación, en las siguientes ilustraciones, se podrá observar el diseño de estas tres partes del cliente web:

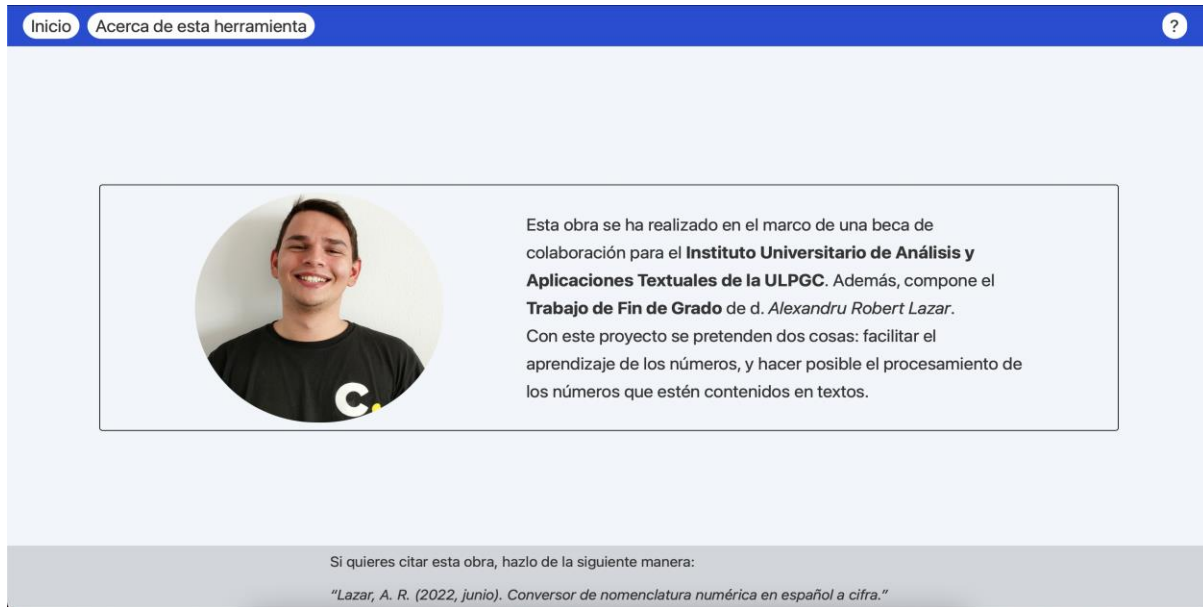


Ilustración 2: Página de presentación de la herramienta – formato escritorio





Esta obra se ha realizado en el marco de una beca de colaboración para el **Instituto Universitario de Análisis y Aplicaciones Textuales de la ULPGC**. Además, compone el **Trabajo de Fin de Grado** de d. *Alexandru Robert Lazar*.

Con este proyecto se pretenden dos cosas: facilitar el aprendizaje de los números, y hacer posible el procesamiento de los números que estén contenidos en textos.

Si quieres citar esta obra, hazlo de la siguiente manera:

"Lazar, A. R. (2022, junio). Conversor de nomenclatura numérica en español a cifra."

Ilustración 3: Página de presentación sobre la herramienta - formato móvil

Inserta tu propio número o utiliza algunos de nuestros ejemplos:

[doscientos mil ochocientos cuatro](#)
[dos dosmilcuatrocientos tresavos](#)
[veinte coma cuatrocientos ocho](#)
[dos con una milésima](#)

Al insertar tu propio número, deja que la función de "Autocompletar" te ayude.
Para seleccionar una opción u otra con el teclado, utiliza las flechas,
y, una vez tengas tu opción elegida, pulsa el tabulador.

Nota: La función de "Autocompletar" solamente funciona con
números enteros y decimales. Los fraccionarios se deberán introducir manualmente.

Ilustración 4: Modal de ayuda, formato escritorio

Cerrar

Inserta tu propio número o utiliza algunos de nuestros ejemplos:

[doscientos mil ochocientos cuatro](#)
[dos dosmilcuatrocientos tresavos](#)
[veinte coma cuatrocientos ocho](#)
[dos con una milésima](#)

Al insertar tu propio número, deja que la función de "Autocompletar" te ayude. Para seleccionar una opción u otra con el teclado, utiliza las flechas, y, una vez tengas tu opción elegida, pulsa el tabulador.

Nota: La función de "Autocompletar" solamente funciona con números enteros y decimales. Los fraccionarios se deberán introducir manualmente.

Ilustración 5: Modal de ayuda, formato móvil

Inicio Acerca de esta herramienta ?

Convertor de texto a número

vein|

veinte
veintiuno
veintiuna

El resultado se mostrará aquí

Convertir

Si quieres citar esta obra, hazlo de la siguiente manera:
"Lazar, A. R. (2022, junio). Convertor de nomenclatura numérica en español a cifra."

Ilustración 6: Página de conversión de textos a números, formato escritorio



Ilustración 7: Página de conversión de textos a números, formato móvil

En cuanto al diseño de la aplicación para realizar las pruebas automáticas, también será muy sencillo, puesto que consistirá en un formulario que únicamente pedirá al usuario elegir la carpeta en la que se encuentra el fichero de entrada, como se puede ver en la próxima ilustración. Además, habrá una barra de progreso que indica al usuario del avance de la ejecución de estas pruebas.

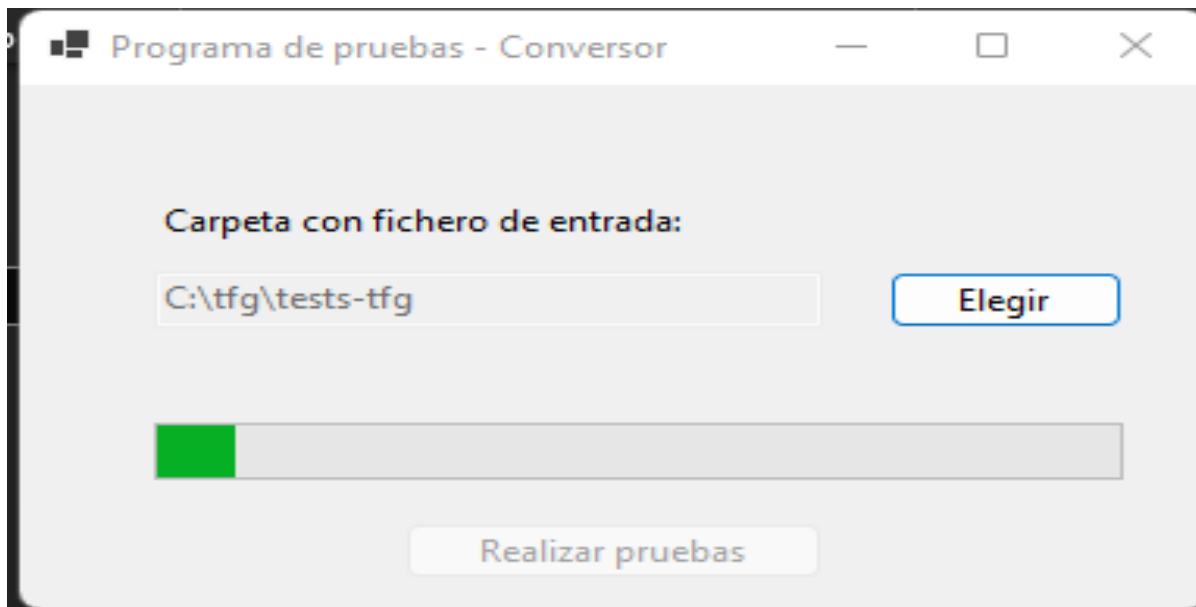


Ilustración 8: Aplicación para realizar pruebas automáticas

Los contenidos del fichero de entrada se podrán ver en el Anexo II.

4.3. Implementación

Para la implementación de este proyecto, al haber sido realizado por una única persona, se ha realizado mediante programación en cascada. Esta metodología define cinco fases principales: análisis, diseño, implementación, verificación y mantenimiento [14]. Téngase en cuenta que, a pesar de tratarse de desarrollo secuencial de software, esta metodología permite dar marcha atrás si fuera necesario.

En cuanto el servicio de conversión, su implementación se hará mediante un servicio “Windows Communication Foundation”, que “consiste en un marco para crear aplicaciones orientadas a servicios”, y es de tipo servicio SOAP [9]. Ello facilitará realizar más aplicaciones en el futuro, que podrán aprovechar este servicio, sin tener que repetir (ni revelar) su código. Para crearlo, se utilizará el lenguaje de programación C#, usando .NET 5.0.

Por otra parte, en cuanto al sitio web, también se utilizará el lenguaje de programación C#, utilizando el marco de trabajo ASP.NET. Además, se utilizará, por supuesto HTML, JavaScript y TailwindCSS [15].

Por último, en cuanto a la herramienta para realizar las pruebas, se ha utilizado un formulario WinForms, también programado en C#. Es importante notar que, en el desarrollo de este proyecto, si bien se ha utilizado TDD, no ha sido hasta la finalización del desarrollo del servicio de conversión cuando se han creado pruebas automáticas. Además, estas pruebas únicamente comprueban el correcto funcionamiento de este servicio, y no el de la herramienta de autocompletado. Las comprobaciones relativas a esta última se han realizado de manera manual en todo momento.

La decisión de utilizar estos lenguajes fundamentalmente se debe a que se desea mantener una cierta uniformidad con el resto de los proyectos realizados por el Instituto Universitario de Análisis y Aplicaciones Textuales.

Entrando ya en detalles de la implementación, lo primero que se va a hacer es crear el fichero de números. Este fichero contendrá un número por cada línea y vendrá dado en el siguiente formato:

`<nomenclatura>;<cifra>`

Por cada cifra, además, puede haber más de una nomenclatura (por ejemplo, para la cifra “1”, puede existir “un”, “uno” o “una”. Para no tener que insertar demasiadas nomenclaturas para un mismo número, esta herramienta tratará automáticamente aquellos números que tengan formas en femenino o en plural (por ejemplo, “una” o “unas”). Por tanto, en el ejemplo dado solamente se tendrá que insertar en el fichero de números las dos primeras formas, es decir, “un” y “uno”.

Además, en las nomenclaturas que hagan referencia a números decimales o fraccionarios, la cifra que se le asignará tendrá una “/” delante, para poder distinguirlos.

Una vez visto esto, se pasará a explicar la implementación del cliente web y, tras ello, se explicará la implementación del servicio del conversor de números. Para concluir, se hará la explicación de la implementación del programa de pruebas automáticas.

En cuanto al despliegue de este, se podrá encontrar en [16]. Este despliegue se encuentra alojado en un servidor Microsoft Azure, para asegurar la máxima disponibilidad de este conversor de números para los usuarios.

El cliente web tendrá dos partes importantes: una parte de frontend, donde se realizarán las funciones del autocompletar, fundamentalmente, y una parte de backend, donde se realizará la obtención de resultados del servicio.

Para el frontend, existirán algunos ficheros fundamentales:

- “Default.aspx” – contendrá el HTML de la página de inicio y el código relacionado con los eventos (como la inserción de texto en el campo de texto o la pulsación de las flechas del teclado, que navegará por las distintas funciones de la función de autocompletado).
- “Site.Master” – contendrá el HTML de la barra de navegación (realmente, habrá dos barras de navegación que se mostrarán según el tamaño de la pantalla), así como de la barra que explicará cómo citar este trabajo y el modal de ayuda, explicado previamente en el apartado “4.2. Diseño”.
- “Scripts/CustomScripts/Dictionary.js” – contiene el código que representará la estructura de datos del diccionario que contendrá los pares clave y valor que enlazarán los textos con las representaciones numéricas.
- “Scripts/CustomScripts/Autocomplete.js” – contiene el código de la función de autocompletado.

Y, en el backend, se utilizarán los siguientes ficheros:

- “Default.aspx.cs” – contendrá las llamadas a los endpoints del servicio de conversión.
- “Web.config” – contiene la configuración del cliente web, así como la información para conectarse al servicio del conversor.

A continuación, se explicará el código más complejo de la aplicación, que está contenido en los dos últimos ficheros. Explicar el código de los otros cuatro ficheros no se hace tan relevante, puesto que es sencillo.

En cuanto al fichero que contiene la definición de la estructura de datos, hallamos las siguientes funciones:

```
class Dictionary {
  contents = {}
  add(key, value) {
    if (!this.contents[key]) {
      this.contents[key] = value
      return true
    }
    return false
  }
  remove(key) {
    if (!this.contents[key]) {
      return false
    }
    return delete this.contents[key];
  }
  get(key) {
    if (!this.contents[key]) {
      return null
    }
    return this.contents[key]
  }
  containsValue(val) {
    if (Object.values(this.contents).indexOf(val) > -1) return true
    return false
  }
  clone(dict) {
    this.contents = {...dict.contents}
  }
}

var dict = new Dictionary()
```

Ilustración 9: Código del fichero "Dictionary.js"

Esta estructura de datos consiste en un array de pares clave-valor, cuyas claves serán las nomenclaturas textuales de los números, y los valores serán sus representaciones numéricas.

Este código contiene básicamente un CRD: “get()”, que obtiene el valor asociado a una clave; “add()”, que añade una nueva clave que no exista en la tabla (y que tiene asociado también un valor), y “remove()”, que elimina una clave con su valor si existen en la tabla. Nótese que las funciones de añadir y de borrar devolverán verdadero o falso, en función de si se ha podido añadir o eliminar esa clave, mientras que la función “get()” devolverá nulo en caso de no encontrarse la clave especificada.

Además, existirán dos funciones adicionales: una función que comprobará si un valor existe en la tabla y una función de clonación que, como su nombre lo indica, realizará una nueva copia profunda [17] de la tabla de contenidos.

Ahora, se explicará el código de la función de autocompletado. Este código es el más complejo de esta parte de la aplicación, por lo que la explicación se realizará función a función.

Además, téngase en cuenta que este código está orientado a una implementación sin estados (es decir, las sugerencias de la función de autocompletado no dependerán del estado anterior de la caja de texto, lo cual es útil, por ejemplo, si se borra una parte del texto seleccionándolo con el ratón), pero su coste añadido es que, al insertar (o eliminar) un solo carácter, se tienen que verificar todos los números anteriores para saber qué sugerencias se dan al usuario. Ello podrá causar que la herramienta parezca lenta, pero evitará los fallos a la hora de modificar la caja de texto de forma diferente que mediante el uso de teclas.

Lo primero que se puede encontrar en este código es una función que procesa los números que se van a añadir al diccionario explicado anteriormente. Ello se hará de tal forma que las formas femeninas y plurales de aquellos números que las tengan se insertarán automáticamente, permitiendo así no insertar dichas formas en el fichero de números del servicio. Concretamente, la función es la siguiente:

```

function treatNumbersToAdd(key, val) {
  var normalizedKey = key.normalize('NFD').replace(/[\u0300-\u036f]/g, '')
  if (key !== 'tercero') dict.add(key, val)
  if (key.includes('illón')) {
    pluralKey = key.replace('illón', 'illones')
    dict.add(pluralKey, val)
  }
  if (key === 'millardo') {
    dict.add(key + 's', val)
  }
  if (val.includes('/') || key === 'uno') {
    if (key.lastIndexOf("o") === key.length - 1) {
      dict.add(key + "s", val)
      if (!key.includes('terci') && !key.includes('medi')) {
        dict.add(key.substring(0, key.length - 1) + "a", val)
        dict.add(key.substring(0, key.length - 1) + "as", val)
      }
    }
  }
  else if (key.includes("uno")) {
    dict.add(key.substring(0, key.length - 1) + "a", val)
  }
  if (key.lastIndexOf("os") === key.length - 2 && val.length >= 3) {
    dict.add(key.substring(0, key.length - 2) + "as", val)
  }
  if (key === 'mitad') {
    dict.add('mitades', val)
  }
  if (key === 'tercera') {
    dict.add('terceras', val)
  }
  if (val.length >= 4 && val.includes('1000') && key !== 'millardo') {
    dict.add(normalizedKey + 'ésima', '/' + val)
    dict.add(normalizedKey + 'ésimo', '/' + val)
    dict.add(normalizedKey + 'ésimas', '/' + val)
    dict.add(normalizedKey + 'ésimos', '/' + val)
    dict.add("diez" + normalizedKey + 'ésima', '/' + val + "0")
    dict.add("diez" + normalizedKey + 'ésimo', '/' + val + "0")
    dict.add("diez" + normalizedKey + 'ésimas', '/' + val + "0")
    dict.add("diez" + normalizedKey + 'ésimos', '/' + val + "0")
    dict.add("cien" + normalizedKey + 'ésima', '/' + val + "00")
    dict.add("cien" + normalizedKey + 'ésimo', '/' + val + "00")
    dict.add("cien" + normalizedKey + 'ésimas', '/' + val + "00")
    dict.add("cien" + normalizedKey + 'ésimos', '/' + val + "00")
  }
}

```

Ilustración 10: Código de añadido de números al diccionario

Una vez visto esto, a continuación, se verá una función que realizará la carga de los números en el diccionario. La diferencia con la función anterior es que, mientras aquella únicamente trataba las formas femeninas y plurales de los números para añadirlas, esta función realiza la carga desde el servicio de todos los números disponibles en el fichero de números (véase la siguiente ilustración).

```
async function reloadAllNumbers() {
  if (Object.keys(dict.contents).length === 0) {
    var response = await $.ajax({
      type: 'GET',
      url: 'Default.aspx/GetNumbers',
      contentType: 'application/json; charset=utf-8',
      dataType: 'json'
    })
    for (let [key, val] of Object.entries(response.d)) {
      treatNumbersToAdd(key, val)
    }
    dict.add("menos", "-")
    dict.add("con", ",")
    dict.add("y", " ")
    dict.add("coma", ",")
  }
}
```

Ilustración 11: Código que tiene la función de carga de números.

Una vez hecha la carga de números, en el momento en el que el usuario modifique los contenidos de la caja de texto, se ejecutará una función que obtendrá los números similares (y válidos, según el contexto) al último número que se encuentre en la misma (véase la siguiente ilustración).

```

function getSimilarNumbers(num) {
  // changed to stateless implementation
  lastInsertedNumber = {
    index: -1
  }
  filterPlurals = false
  filterFeminines = false
  filterSingulars = false
  filterMasculines = false
  excessDecimalSeparators = false
  separatorInserted = false
  showThousands = true
  wasLastNumberSeparator = false
  maxLengthInText = 0
  filterFractions = false
  isFractionEnded = false
  wrongNumber = false

  var num = num.toLowerCase()
  similarNumbers = []

  num.split(" ").forEach((word, index) => {
    word = word.trim().toLowerCase()
    if (index < num.split(" ").length - 1) {
      clickedNumber(word, false)
    }
  })
  num = num.split(" ")[num.split(" ").length - 1]
  for (let [key, val] of Object.entries(dict.contents)) {
    if (num !== '' && key.search('\\b' + num.trim() + '.*') === 0) {
      treatAutocompleteNumbers(key, val)
    }
  }
  document.getElementById('number-options').innerHTML = ''
  similarNumbers.forEach((number) => {
    document.getElementById('number-options').innerHTML += '<li onclick="clickedNumber('${number}')" class="even:bg-gray-100 cursor-pointer p-2 hovers:font-bold w-auto" id="auto-${number}">${number}</li>'
  })
}

```

Ilustración 12: Código que obtiene los números similares al último.

Aquí se pueden encontrar tres cuestiones importantes.

La primera de ellas reinicia los filtros, siguiendo el principio de que no se dependerá del estado anterior de la caja de texto.

Lo siguiente que se realiza es verificar el estado de los números anteriores al último. De esta forma si algún número erróneo insertado, ya no se mostrará ninguna sugerencia al usuario. Además, se adaptarán las sugerencias mostradas, según los números insertados.

Por último, se cargan todos los contenidos del diccionario y se pasan unos filtros (contenidos en la función “treatAutocompleteNumbers()”, explicada más adelante). Todas aquellas claves que cumplan los filtros se introducen en un array llamado “similarNumbers” (lo cual se hace directamente en la función mencionada). Cuando se haya terminado de cargar todas estas claves, se muestran al usuario, insertándolas en el contenedor de las sugerencias.

A continuación, la función que se va a explicar es aquella que verifica el número máximo insertado en el texto. Esta función es muy útil puesto que apoyará al autocompletado para no ofrecer sugerencias como “dos millones tres trillones”. Esta función se podrá ver en la siguiente ilustración:

```

function getMaxUnitsFromText() {
    let text = document.getElementById("MainContent_TextBox1").value
    let indexOfSeparator = text.indexOf('con')
    if (indexOfSeparator === -1) indexOfSeparator = text.indexOf('coma')
    if (indexOfSeparator === -1) {
        var splitWords = text.split(" ")
        for (i = 1; i < splitWords.length; i++) {
            if (!dict.get(splitWords[i - 1]).includes('/') && dict.get(splitWords[i - 1]).length !== 2) {
                indexOfSeparator = i
                break
            }
        }
    }
    if (indexOfSeparator === -1) return
    let splitWordsFromSeparator = text.substr(indexOfSeparator).trim().split(" ")
    splitWordsFromSeparator.shift()
    let firstMaxLength = 0
    let firstMaxLengthIndex = 0
    splitWordsFromSeparator.forEach((word, index) => {
        if (index === splitWordsFromSeparator.length - 1) return
        if ((numericNum = dict.get(word)) !== null) {
            if (!numericNum.includes('/')) {
                if (numericNum.length > 4) {
                    firstMaxLength = numericNum.length
                    firstMaxLengthIndex = index
                    return
                } else {
                    if (numericNum.length > firstMaxLength) {
                        firstMaxLength = numericNum.length
                        firstMaxLengthIndex = index
                    }
                }
            }
        } else {
            if (index < splitWordsFromSeparator.length - 1) firstMaxLength = -1
            return
        }
    })
    if (firstMaxLength === -1) return -1
    let finalLength = firstMaxLength > 4 ? firstMaxLength-1 : firstMaxLength
    // check previous numbers, if length > 1 add it (-1)
    for (i = firstMaxLengthIndex-1; i >= 0; i--) {
        if ((numericNum = dict.get(splitWordsFromSeparator[i])) !== null) {
            if (numericNum.length === 4) finalLength += 4
            else if (i === 0) finalLength += numericNum.length-1
        } else {
            return -1
        }
    }
    return finalLength
}

```

Ilustración 13: Código con la función de verificación del máximo número introducido.

La siguiente función que se va a comentar es la de los filtros. Esta función se sustenta en la siguiente aproximación teórica (véase la siguiente ilustración):

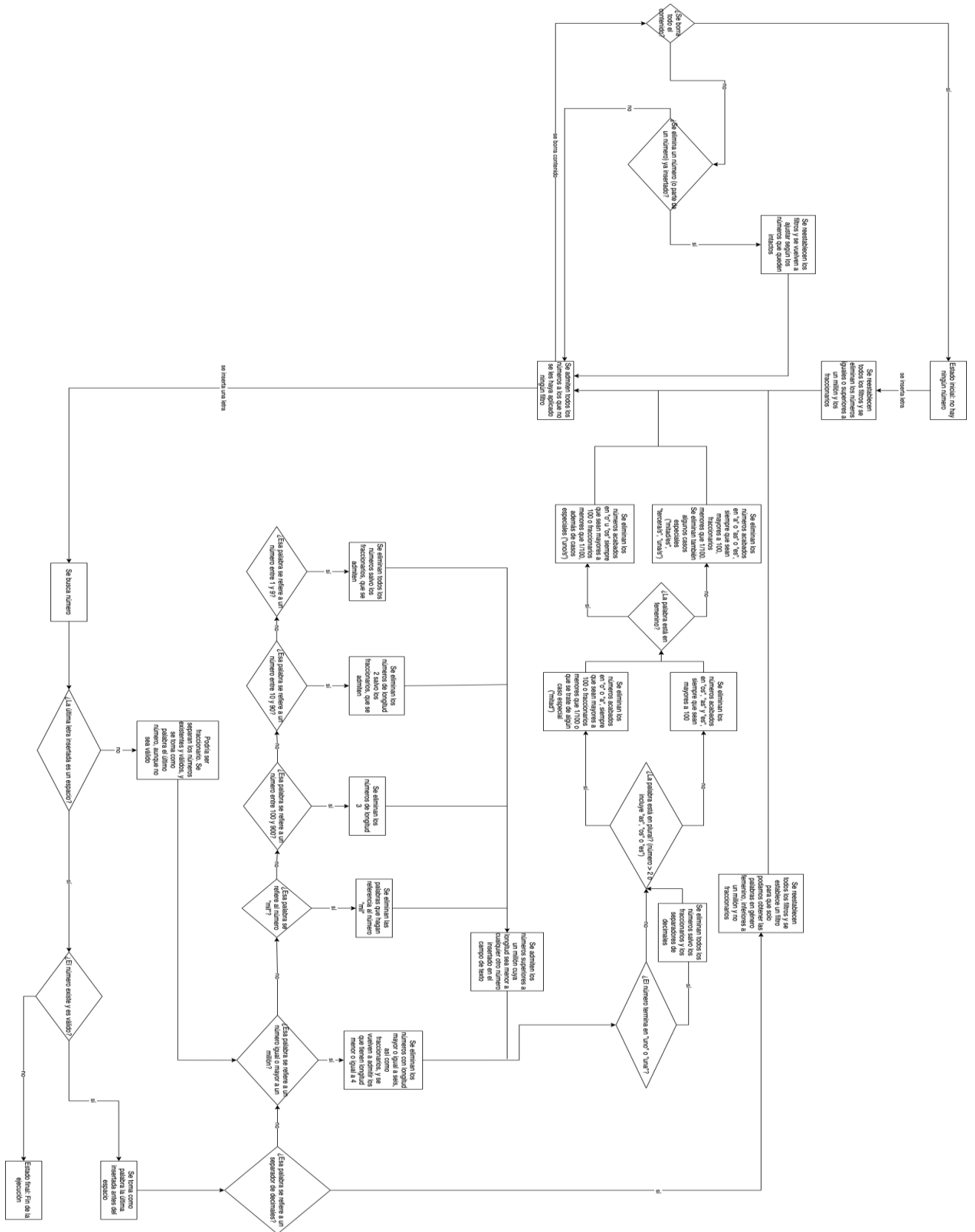


Ilustración 14: Aproximación teórica para aplicar los filtros en los números.

Dada la longitud de esta función, no es factible mostrarla mediante capturas de pantalla en la presente memoria. Sin embargo, se podrá encontrar en el código con el siguiente nombre: `“treatAutocompleteNumbers()”`.

Esta función tiene una particularidad: se puede usar tanto para verificar que un número insertado es correcto (mediante el parámetro `“isTestMode”`, puesto como `“verdadero”`), como para insertar nuevas sugerencias (si han pasado los filtros) en el ya mencionado array `“similarNumbers”`. Este modo de prueba es especialmente útil, al haber realizado una implementación que no tiene en cuenta el estado anterior de la caja de texto, para no mostrar sugerencias al usuario si se ha introducido algún número erróneo con anterioridad (por ejemplo, no se mostrarán sugerencias si se ha introducido `“millón dos”`).

Para concluir con esta parte de la aplicación, se explicará la función que verifica el estado de un número. No debe confundirse con la función anterior, puesto que esta función se encarga de ajustar los filtros y de insertar el número en la caja de texto. Además, esta función no modifica en ningún momento las sugerencias de la función de autocompletado. Dado su tamaño, tampoco es factible mostrarla en esta memoria; sin embargo, se podrá encontrar en el código con el nombre de `“clickedNumber()”`. Sin embargo, para realizar el ajuste de los filtros, esta función también aprovecha la aproximación teórica vista en la ilustración anterior.

Con esto comentado, concluye la explicación del código relativo al cliente web. Ahora, se pasará a explicar la implementación del servicio que realizará la conversión de los números.

El servicio se encontrará desplegado en Microsoft Azure, en [18]. El motivo por el que este servicio se ha desplegado en Azure es igual que para el cliente web: asegurar la máxima disponibilidad del servicio.

En el fichero que contiene el código del servicio, se pueden encontrar múltiples funciones que se comentarán a continuación, individualmente, dada su complejidad.

Lo primero que se puede encontrar en este servicio es una función que únicamente carga los números del fichero (véase la siguiente ilustración):

```

private bool LoadNumbers()
{
    string numberFilePath = @"D:\home\site\storage\tft-data\numbers.txt";
    if (File.Exists(numberFilePath))
    {
        using (StreamReader sr = new StreamReader(numberFilePath))
        {
            while (sr.Peek() >= 0)
            {
                string[] vs = sr.ReadLine().Split(';');
                numberSet.Add(vs[0], vs[1]);
            }
            sr.Close();
        }
        return true;
    }
    else
    {
        return false;
    }
}

```

Ilustración 15: Función de carga de números.

A continuación, se explicará la función que realiza el procesamiento propiamente dicho del texto. Al igual que ha ocurrido en algunas funciones del cliente web, esta función es demasiado larga para reflejarlo en el presente informe, por lo que se realizará únicamente su explicación. Sin embargo, se podrá encontrar en el código con el nombre de “ComputeNumber()”.

Lo que hace esta función es tomar la aproximación teórica vista anteriormente y obtener el número en cifras a partir del texto que se le envía por parámetro. Realmente, este código comparte algunas similitudes con el relativo a la obtención de las sugerencias de la función de autocompletado, pero, en vez de darse sugerencias, se realiza el cálculo del número. Además, esta función también puede calcular números fraccionarios complejos (como, por ejemplo, “un diezmiltreintaicuatavo”), a diferencia de la función de autocompletado.

La siguiente función que se va a explicar es “ComputeFractionNumbers()”. Al igual que la función anterior, se hace demasiado larga para insertarla como captura de pantalla, por lo que únicamente se realizará la explicación de esta.

El objetivo de esta función no es más que obtener, a partir de un número fraccionario, aquellos números que lo componen. Por ejemplo, si se tiene “diezmilcuarentaitresavos”, se devolverá “diez mil cuarenta tres”. Ello es especialmente útil para insertar el número a la derecha de la fracción aprovechando completamente la función “ComputeNumber()”.

Si bien esta función aparenta sencilla, lo cierto es que ha sido una de las funciones más complicadas de realizar en todo este proyecto, sobre todo dado que suponía la lectura carácter a carácter de la palabra.

Una de las dificultades era determinar si la palabra debería leerse de izquierda a derecha o de derecha a izquierda. A priori, está claro que la lectura debería ser de izquierda a derecha, pues es la forma natural de leer textos en español; sin embargo, en algunas ocasiones no se calculaba bien el número.

Para explicar el motivo, es necesario adentrarse un poco más en cómo funciona esta función. La separación de los números se consigue mediante la obtención de substrings de caracteres, que aumentan de tamaño, a partir de la palabra, con la finalidad de detectar si estas conformaban números. Una vez comprobado, si la substring es un número existente en el fichero de números, se introduce en una lista de números y se reinicia la posición inicial de la substring a la siguiente de la actual para empezar a calcular el siguiente número.

Supóngase el número “milmillonavo”. La secuencia de substrings sería la siguiente: “m”, “mi”, “mil”, “m”, “mi”, “mil”, “milm”, “milmi”, “milmil”, “milmill”, “milmillo”, “milmillon” (nótese que el “avo” se elimina al principio de la función, ya que es irrelevante para la separación del número).

De esta manera, resulta trivial ver cuál es el problema: el número “mil” se insertaba dos veces. En este caso, la solución parecía sencilla: leer los números al revés, y, de esta manera, no emplear “parches”. Sin embargo, si bien la mayoría de los números funcionaban correctamente, había un caso que tampoco funciona muy bien: el número “cuatrillonavo”. Al leer este número al revés, se llega únicamente a obtener la substring de “trillon”, por lo que se calculaba mal.

Eso ha hecho que sea necesario el empleo de algunas verificaciones de casos específicos, que se ven claramente en el código.

En cuanto a la siguiente función a explicar (véase la primera ilustración de la siguiente página), lo que se desea hacer es obtener, al tener un número con decimales, cuánta longitud tiene su parte decimal. Ello es especialmente útil por dos motivos: para saber cuántos ceros hay que meter por delante de la parte decimal (si hubiera que meterlos, como en el siguiente ejemplo: “uno con cuarenta milésimas”, que llevaría un cero por delante del “40” – quedándose como “1,040”), y para saber si un número con decimales está bien escrito (por ejemplo, el número “dos con veinte décimas” sería incorrecto).

```

/**
 *
 * <summary>Method to compute the amount of shifts a decimal should go through.</summary>
 * <param name="number">Number ended in '-ésima'.</param>
 * <returns>Amount of shifts a number should go through.</returns>
 *
 */
1 reference | Alexandru Robert Lazar, 5 days ago | 1 author, 3 changes
private int ComputeDecimalShifts(string number)
{
    string originalNumber = number;
    if (numberSet.ContainsKey(number))
    {
        return numberSet[number].Length - 1;
    }
    if (number.Contains("décima"))
    {
        return 1;
    }
    if (number.Contains("ésimas")) number = number.Replace("ésimas", String.Empty);
    else if (number.Contains("ésima")) number = number.Replace("ésima", String.Empty);
    else if (number.Contains("écima")) number = number.Replace("écima", String.Empty);
    else number = number.Replace("écimas", String.Empty);
    if (number.Contains("llon") && !number.Contains("llones"))
    {
        number = number.Replace("llon", "llón");
    }
    if (numberSet.ContainsKey(number))
    {
        return numberSet[number].Length - 1;
    }
    // Special cases like "a hundred thousand millionth"
    string fractionNumber = "";
    if (originalNumber.Contains("ésima"))
    {
        // last number
        fractionNumber = ComputeFractionNumbers(originalNumber);
        string origPrevNumberInserted = prevNumberInserted;
        int origMaxNumberLength = maxNumberLength;
        bool origIsThousandInserted = isThousandInserted;
        prevNumberInserted = "";
        maxNumberLength = 0;
        isThousandInserted = false;
        fractionNumber = ComputeNumber(fractionNumber).Replace(" ", String.Empty);
        prevNumberInserted = origPrevNumberInserted;
        maxNumberLength = origMaxNumberLength;
        isThousandInserted = origIsThousandInserted;
    }
    if (fractionNumber.Contains("Error") || fractionNumber.Contains("inválido"))
    {
        return -1;
    }

    return fractionNumber.Length - 1;
}

```

Ilustración 16: Función para obtener la longitud de la parte decimal de un número.

A continuación, y para concluir con esta parte, se van a comentar las dos funciones de la siguiente ilustración, que no tienen mayor complejidad.

La primera de ellas tiene que ver con dar un formato al número que se devolverá por el conversor. Lo que se hace es que se separan los números de las distintas partes (decimal, entera o fraccionaria) de un número de tal forma que sus cifras estén separadas de tres en tres (salvo de que se trate de un número con cuatro cifras, en cuyo caso no se realiza la separación). Ello hará que sea mucho más fácil de visualizar por el usuario. Además, mientras en la parte que está a la izquierda del separador (en el caso de ser un número con decimales, sería la “,” y, en un fraccionario, será la “/”) la separación se hace de derecha a izquierda, en el caso contrario se hará de izquierda a derecha (ello vendrá definido por el parámetro “orientation”).

La segunda función, como su nombre indica, consiste en invertir el orden de los elementos en un array. Al contrario que en otros lenguajes de programación, C# no tiene una función en el tipo de objeto “Array” para invertir el orden de sus elementos, por lo que será útil para realizarlo.

```
/**
 *
 * <summary>Method for separating strings each three characters from right to left.</summary>
 * <param name="str">The string to be separated.</param>
 * <returns>The separated string.</returns>
 *
 */
private string SeparateNumbers(string str, string orientation)
{
    if (orientation == "left")
    {
        str = ReverseString(str);
    }
    for (int i = 0; i < str.Length; i += 4)
    {
        str = str.Substring(0, i) + " " + str.Substring(i);
    }
    if (orientation == "left")
    {
        return ReverseString(str);
    }
    else
    {
        return str;
    }
}

/**
 * <summary>Method to reverse a string easily.</summary>
 * <param name="str">String to be reversed.</param>
 * <returns>The reversed string</returns>
 *
 */
private string ReverseString(string str)
{
    char[] vs = str.ToCharArray();
    Array.Reverse(vs);
    return new string(vs);
}
```

Ilustración 17: Funciones de separación de cifras e inversión de elementos en un array.

Una vez vista la implementación del servicio de conversión, se pasará a explicar la última parte de este proyecto, que será la herramienta para hacer pruebas automáticas.

En esta parte, lo único que importa es el fichero llamado "Form1.cs", que contiene, fundamentalmente, la gestión de eventos del formulario que se mostrará al usuario (para elegir la carpeta de entrada) y, como parte de ésta, se incluirán algunas funciones para realizar correctamente las pruebas.

El funcionamiento de esta parte es prácticamente trivial, pudiendo deducirse fácilmente echando un vistazo al Anexo II. Fundamentalmente, tras elegir la carpeta donde se encuentra el fichero de entrada, cuyo nombre tendrá que ser "entrada.txt", y cuyos contenidos seguirán el siguiente formato:

```
<representación en cifras>:<representación textual>
```

De esta forma, este fichero contendrá una línea por cada caso de prueba.

Tras pulsar sobre el botón "Realizar pruebas", en la carpeta elegida se creará un fichero llamado "salida.txt", en el que se mostrarán todos los fallos encontrados. De esta forma, lo ideal es que este fichero esté vacío tras la ejecución de las pruebas.

Durante el procesamiento de cada caso de prueba, lo que se hace es obtener el texto que representa al número (que se encuentra a la derecha del separador ":" en cada caso), se llama a la función del servicio para realizar la conversión de este y, por último, se compara el resultado obtenido del servicio con el resultado esperado, que se encuentra a la izquierda del separador ":". Si ambos resultados coinciden, entonces se dará por superado este caso de prueba; en caso contrario, se considerará un fallo y se escribirá la información sobre este fallo en el fichero de salida.

Una vez explicado esto, concluye la realización de este apartado, con lo que, a continuación, se hará un análisis de calidad al código.

4.4. Análisis de calidad

En este apartado, se hará un análisis de calidad al código de las tres partes de este proyecto. Para ello, se empleará el programa SonarQube [19], dada su potencia para encontrar potenciales fallos de calidad en el código. De esta forma, la calidad del proyecto se valida mediante dos vertientes: funcionamiento (mediante el TDD visto anteriormente) y código.

En primer lugar, se hará el análisis de calidad al código del servicio de conversión; a continuación, al código del cliente web; y, por último, al código de la aplicación para realizar las pruebas.

4.4.1. Servicio de conversión

Lo primero que se debe hacer para realizar este análisis de calidad es descargar el software SonarQube en un sistema Windows (dado que se requiere el uso de la herramienta MSBuild [20], disponible solo en esta plataforma). Además, se deberá descargar el escáner específico para .NET [21].

No se necesita instalar ningún software en el ordenador para ejecutar el análisis de calidad, sino que únicamente se deben descomprimir los contenidos de los ficheros, según las instrucciones de [21] y [22].

Tras hacerlo, se debe ejecutar el programa según las instrucciones vistas y se deberá acceder a la siguiente dirección:

<http://localhost:9000>

En la página que se mostrará se pedirá iniciar sesión. Al ejecutar la aplicación por primera vez, el usuario y la contraseña por defecto serán “admin”. Tras iniciar sesión, se exigirá el cambio de la contraseña por defecto. Una vez hecho, ya se podrá acceder a los proyectos que se vayan a escanear. En este caso, será necesario crear un nuevo proyecto pulsado en el botón azul denominado “Create a new project”. Ello abrirá la ventana de la siguiente ilustración, donde se configurará el nombre y la clave del proyecto:

Create a project

All fields marked with * are required

Project display name *
TFG-Service ✓
Up to 255 characters. Some scanners might override the value you provide.

Project key *
TFG-Service ✓
The project key is a unique identifier for your project. It may contain up to 400 characters. Allowed characters are alphanumeric, '-' (dash), '_' (underscore), '.' (period) and ':' (colon), with at least one non-digit.

[Set Up](#)

Embedded database should be used for evaluation purposes only
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

SonarQube™ technology is powered by SonarSource SA
Community Edition - Version 9.4 (build 54424) - LGPL v3 - Community - Documentation - Plugins - Web API

Ilustración 18: Página inicial de configuración, con los datos ya introducidos

A continuación, tras pulsar en el botón “Set Up”, se abrirá la siguiente página, que nos permitirá determinar cómo se realizará el análisis (mediante alguna herramienta remota o en local). En este caso, se deberá escoger la opción “Locally”:

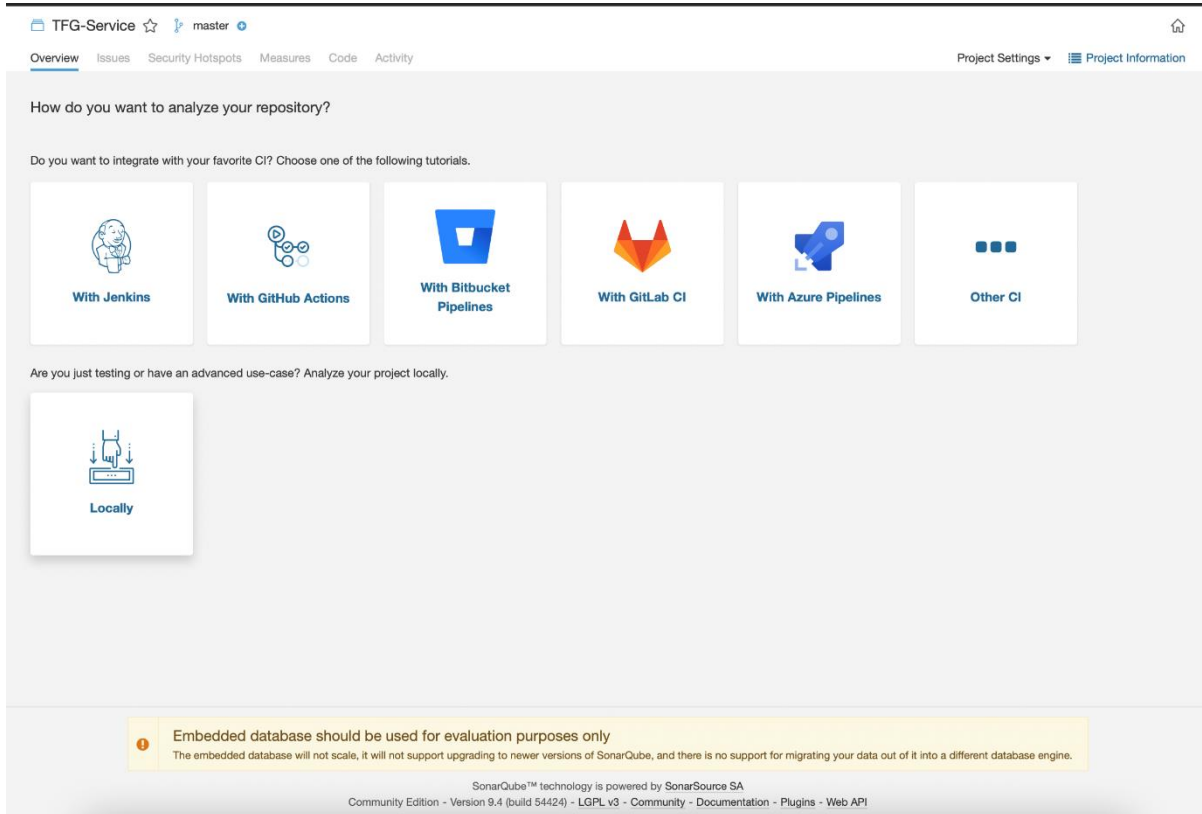


Ilustración 19: Segundo paso de la configuración del proyecto

Como último paso previo a la ejecución del análisis, se abrirá una ventana en la que se deberá generar un token a ser utilizado por el scanner, como se ve en la siguiente ilustración:

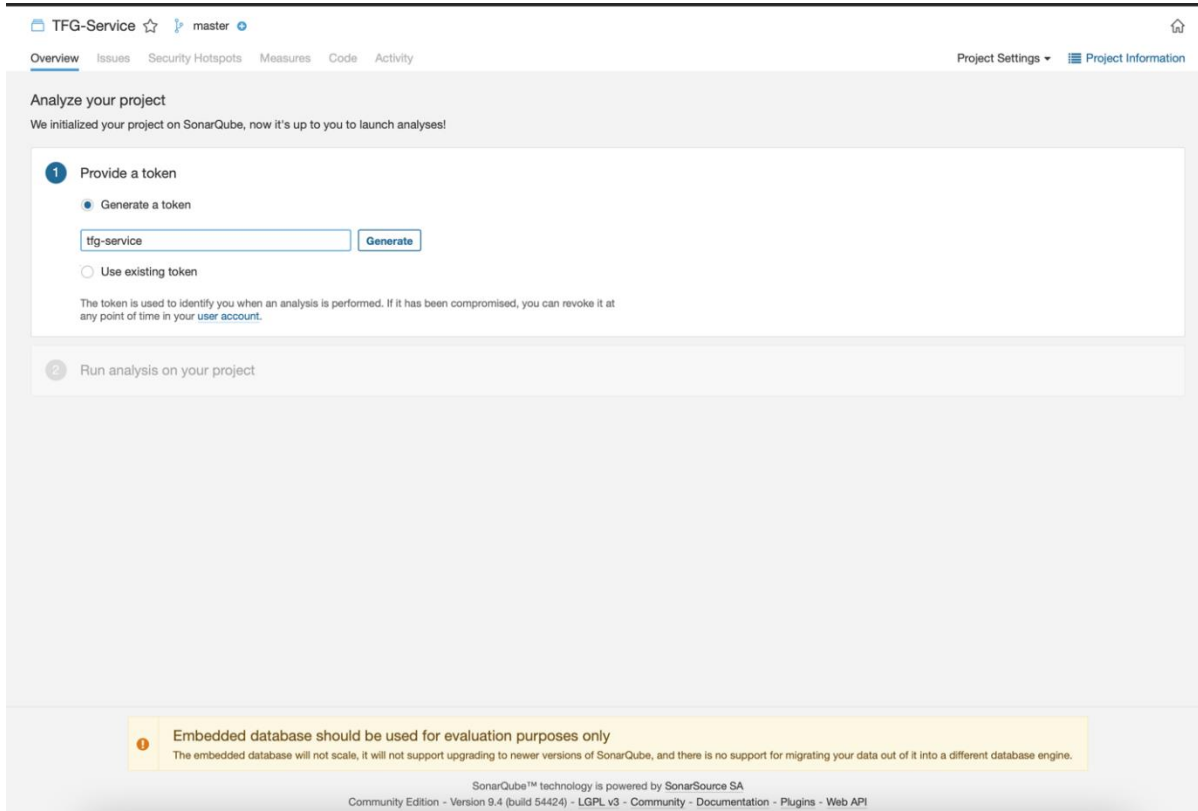


Ilustración 20: Generación del token el proyecto

Una vez se pulse sobre el botón “Generate”, se mostrará el token y, tras pulsar el botón “Continue” que aparecerá, se verán algunas opciones para elegir el lenguaje en el que está realizado el proyecto y, tras seleccionar “.NET”, se darán las instrucciones para ejecutar el análisis, como se ve la siguiente ilustración:

TFG-Service ☆ master

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Analyze your project
We initialized your project on SonarQube, now it's up to you to launch analyses!

1 Provide a token ✔ tfg-service:7ea0e487f89e37b8d77d5de88daa3da7aaafcd3a

2 Run analysis on your project

What option best describes your build?

Choose your build tool

Download and unzip the Scanner for .NET
 Visit the official documentation of the Scanner for .NET to download the latest version, and add the executable's directory to the %PATH% environment variable

Execute the Scanner
 Running a SonarQube analysis is straightforward. You just need to execute the following commands at the root of your solution.

```
SonarScanner.MSBuild.exe begin /k:"TFG-Service" /d:sonar.host.url="http://localhost:9000" /d:sonar.login="7ea0e487f89e37b8d77d5de88daa3da7aaafcd3a" Copy
```

```
MsBuild.exe /t:Rebuild Copy
```

```
SonarScanner.MSBuild.exe end /d:sonar.login="7ea0e487f89e37b8d77d5de88daa3da7aaafcd3a" Copy
```

Please visit the official documentation of the Scanner for .NET for more details.

Is my analysis done? If your analysis is successful, this page will automatically refresh in a few moments.
 You can set up Pull Request Decoration under the project settings. To set up analysis with your favorite CI tool, see the tutorials.
 Check these useful links while you wait: [Branch Analysis](#), [Pull Request Analysis](#).

Ilustración 21: Instrucciones para ejecutar el análisis desde línea de comandos

Antes de ejecutar estos comandos, es importante tener en cuenta que, posiblemente, sea necesario añadir la carpeta que contiene MSBuild en el "path", para que sea detectado (las instrucciones para hacerlo se pueden ver en [23]).

Una vez ejecutados estos comandos, se verán los resultados del análisis, como se ve en la siguiente ilustración:

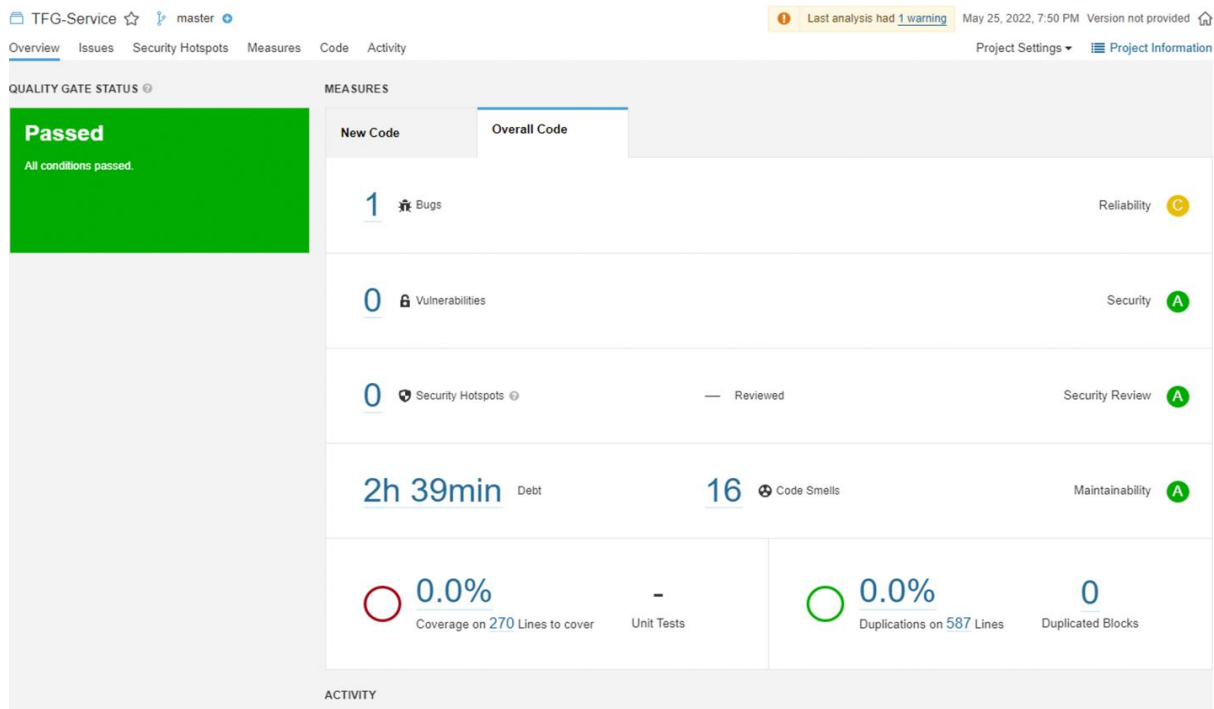


Ilustración 22: Resultados del análisis en el servicio del conversor

Como se puede observar, la calidad del código es bastante buena, ya que solo se ha detectado un posible bug y 16 incidencias menores. Los detalles de algunos de estos fallos se pueden encontrar en las dos siguientes ilustraciones:

TFG-Service ☆ master Last analysis had 1 warning May 25, 2022, 7:50 PM Version not provided

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Service1.svc.cs

- Refactor your code not to use hardcoded absolute paths or URIs. Code Smell
- Change return type to 'void': not a single caller uses the returned value. Code Smell
- Refactor your code not to use hardcoded absolute paths or URIs. Code Smell
- Refactor this method to reduce its Cognitive Complexity from 153 to the 15 allowed. Code Smell +63
- Merge this if statement with the enclosing one. Code Smell +1
- Remove this empty statement. Code Smell
- Use a StringBuilder instead. Code Smell
- Use a StringBuilder instead. Code Smell
- Refactor this method to reduce its Cognitive Complexity from 153 to the 15 allowed. Code Smell +63

TFG-Service /Service1.svc.cs See all issues in this file

```

1 - using System;
2 using System.Collections.Generic;
3 using System.ServiceModel;
4 using System.ServiceModel.Web;
5 using System.Text;
6 using System.IO;
7 using System.ServiceModel.Activation;
8
9 namespace TFTService
10 {
11     [AspNetCompatibilityRequirements(RequirementsMode = AspNetCompatibilityRequirementsMode.Allowed)]
12     [ServiceBehavior(ConcurrencyMode = ConcurrencyMode.Single, InstanceContextMode = InstanceContextMode.PerCall)]
13     public class MainService : IMainService
14     {
15         readonly Dictionary<string, string> numberSet = new Dictionary<string, string>();
16         private string prevNumberInserted = "";
17         private int maxLength = 0;
18         private bool isThousandInserted = false;
19         private string errorLog(string error)
20         {
21             string errorFile = @"D:\home\site\storage\tft-data\error.log";
22
23             FileStream fs;
24             if (!File.Exists(errorFile))
25             {
26                 fs = File.Create(errorFile);
27             }
28             else
29             {
30                 fs = File.Open(errorFile, FileMode.Append, FileAccess.Write);
31             }
32             using (StreamWriter stream = new StreamWriter(fs))
33             {
34                 stream.WriteLine(error);
35             }
36         }
37     }
38 }

```

Refactor your code not to use hardcoded absolute paths or URIs. Why is this an issue? 3 minutes ago L21

Code Smell Minor Open Not assigned 20min effort Comment No tags

Ilustración 23: Incidencias menores del servicio

TFG-Service ☆ master Last analysis had 1 warning May 25, 2022, 7:50 PM Version not provided

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Service1.svc.cs

- Use the return value of method 'Insert'. Bug

1 of 1 shown

```

230
231
232     {
233         var leftFraction = SeparateNumbers(fractionParts[0].Trim(), "left").Trim();
234         if (leftFraction == "") leftFraction = "1";
235         return leftFraction + "/" + fractionParts[1].Trim();
236     }
237     finalResult = SeparateNumbers(parts[0], "left").Trim();
238     if (parts.Length > 1)
239     {
240         finalResult += "," + SeparateNumbers(parts[1], "right").Trim();
241     }
242     if (finalResult[0] == '/')
243     {
244         finalResult.Insert(0, "1");
245     }
246     if (isMinusInserted) finalResult = finalResult.Insert(0, "-");
247     return finalResult.Trim();
248 }
249 private string removePluralsAndFeminineTypes(string textNum)
250 {
251     if (textNum == "" || numberSet.ContainsKey(textNum)) return textNum;
252     string result = textNum;
253     result = result.Replace("es", String.Empty).Replace("llon", "llón");
254     if (numberSet.ContainsKey(result)) return result;
255     result = textNum;
256     result = result.Replace("as", "os");
257     if (numberSet.ContainsKey(result)) return result;
258     result = result.Replace("os", "o");
259     if (numberSet.ContainsKey(result)) return result;
260     StringBuilder sb = new StringBuilder(textNum);
261     if (sb[sb.Length - 1] == 'a') sb[sb.Length - 1] = 'o';
262     result = sb.ToString();
263     if (numberSet.ContainsKey(result)) return result;
264     else return textNum;
265 }
266 /**

```

Use the return value of method 'Insert'. Why is this an issue? 4 minutes ago L242

Bug Major Open Not assigned 10min effort Comment No tags

Ilustración 24: Bug encontrado en el servicio

Téngase en cuenta que esta herramienta, al final, solo es una guía para mejorar la calidad del código, pero, en ningún caso, obliga a arreglar estos problemas. De todas formas, una parte importante de estos problemas (que existen en la versión nombrada “Versión final” de GitHub) estarán resueltos en el momento de la entrega de este proyecto.

Una vez realizado el análisis en el servicio de conversión, se realizará en el cliente web.

4.4.2. Cliente web

Dado que ya se ha explicado cómo se debe realizar un análisis en SonarQube, se van a ver directamente los resultados del cliente web, ya que el procedimiento es exactamente el mismo que en el apartado anterior. El resumen del análisis se ve en la siguiente ilustración:

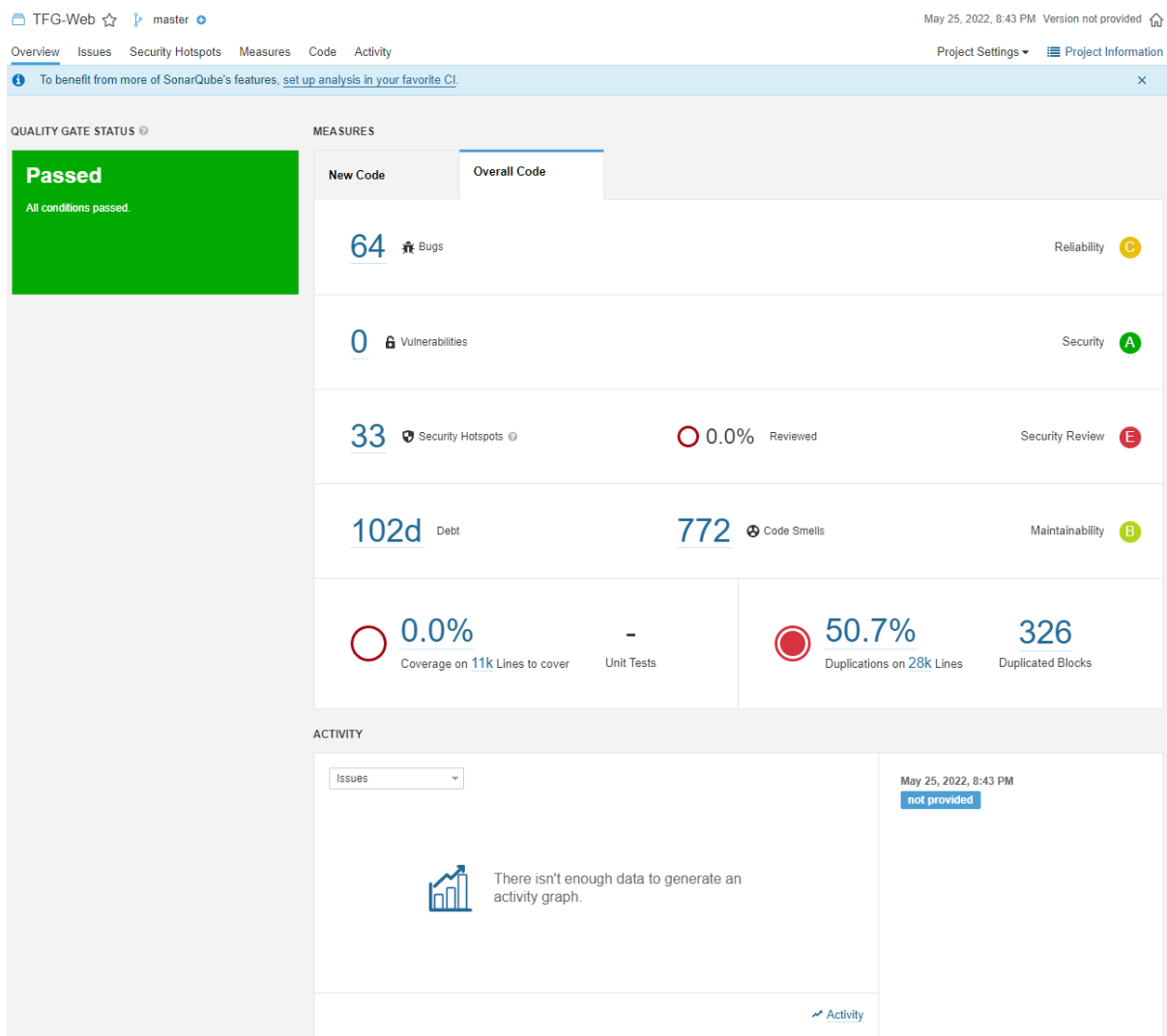
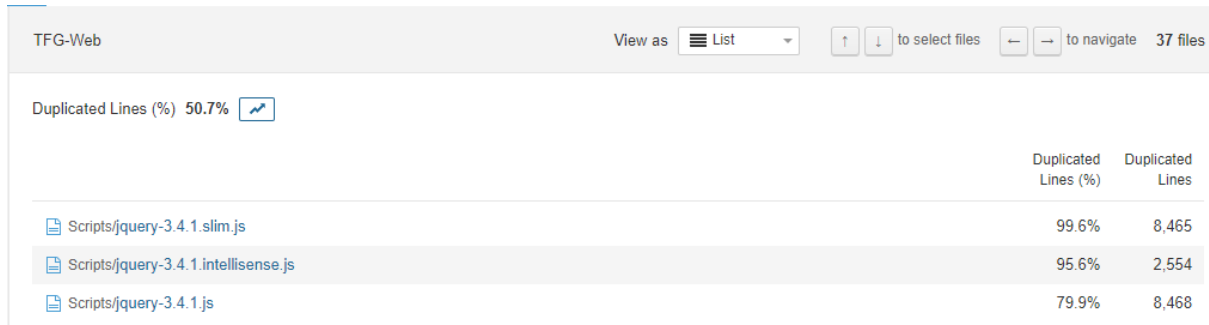


Ilustración 25: Resumen de los resultados del análisis en el cliente web

Como se puede ver, si bien es cierto que el denominado “Quality Gate Status” se ha aprobado, en esta ocasión se ve un número bastante mayor de fallos. Sin embargo, al indagar un poco, se puede apreciar que la mayoría de estos fallos vienen por el código de los ficheros que dan soporte a Bootstrap y jQuery, así como algunos ficheros autogenerados al crear el proyecto ASP.NET. Ello se puede ver de forma clara en la siguiente ilustración, donde se muestran los ficheros que tienen código duplicado:



	Duplicated Lines (%)	Duplicated Lines
Scripts/jquery-3.4.1.slim.js	99.6%	8,465
Scripts/jquery-3.4.1.intellisense.js	95.6%	2,554
Scripts/jquery-3.4.1.js	79.9%	8,468

Ilustración 26: Ficheros con código duplicado

Como se mencionaba en el apartado anterior, esta herramienta sirve de guía para saber que se debería modificar en el código, pero no obliga a hacer estos cambios. En este caso, no se van a modificar los ficheros que dan soporte a librerías (Bootstrap, jQuery) o aquellos que se hayan generado automáticamente.

En las dos siguientes ilustraciones se van a ver las incidencias menores (Code Smells) y los bugs detectados en la aplicación:

Bulk Change 1 / 64 issues 1d effort

About.aspx

- Replace this `` tag by ``. Why is this an issue? 1 day ago L7 accessibility
- Replace this `<i>` tag by ``. Why is this an issue? 1 day ago L8 accessibility
- Replace this `` tag by ``. Why is this an issue? 1 day ago L8 accessibility

Content/bootstrap-theme.css

- Unexpected duplicate "border-color" Why is this an issue? 6 months ago L74 No tags
- Unexpected duplicate "filter" Why is this an issue? 6 months ago L345 No tags
- Unexpected duplicate "filter" Why is this an issue? 6 months ago L372 No tags

Content/bootstrap.css

- Unexpected duplicate "text-decoration" Why is this an issue? 6 months ago L57 No tags
- Unexpected missing generic font family Why is this an issue? 6 months ago L275 No tags
- Unexpected duplicate "-webkit-transition" Why is this an issue? 6 months ago L2674 No tags
- Unexpected unknown media feature name "transform-3d" Why is this an issue? 6 months ago L6332 No tags
- Unexpected missing generic font family Why is this an issue? 6 months ago L6612 No tags

Error.aspx

- Add "lang" and/or "xml:lang" attributes to this "`<html>`" element Why is this an issue? 5 months ago L5 accessibility, wcag2-a

Scripts/CustomScripts/Autocomplete.js

- This branch duplicates the one on line 336 Why is this an issue? 3 months ago L339 pitfall, unused

Scripts/WebForms/SmartNav.js

- '`sc[].text`' is assigned to itself. Why is this an issue? 6 months ago L121 No tags

Ilustración 27: Vista parcial de los bugs de la aplicación

Bulk Change 1 / 772 issues | 102d effort

About.aspx.cs

Add a nested comment explaining why this method is empty, throw a 'NotSupportedException' or complete the implementation. Why is this an issue? 1 month ago ▾ L12 🔗 ⌵

Code Smell Critical Open ▾ Not assigned ▾ 5min effort [Comment](#) suspicious ▾

App_Start/BundleConfig.cs

Add a 'protected' constructor or the 'static' keyword to the class declaration. Why is this an issue? 6 months ago ▾ L10 🔗 ⌵

Code Smell Major Open ▾ Not assigned ▾ 10min effort [Comment](#) design ▾

Content/Site.css

Unexpected duplicate selector ".modal-closed", first used at line 1 Why is this an issue? 1 month ago ▾ L19 🔗 ⌵

Code Smell Major Open ▾ Not assigned ▾ 1min effort [Comment](#) No tags ▾

Content/bootstrap.css

Unexpected duplicate name monospace Why is this an issue? 6 months ago ▾ L112 🔗 ⌵

Code Smell Major Open ▾ Not assigned ▾ 1min effort [Comment](#) No tags ▾

Unexpected duplicate selector "html", first used at line 7 Why is this an issue? 6 months ago ▾ L1080 🔗 ⌵

Code Smell Major Open ▾ Not assigned ▾ 1min effort [Comment](#) No tags ▾

Unexpected duplicate selector "body", first used at line 12 Why is this an issue? 6 months ago ▾ L1084 🔗 ⌵

Code Smell Major Open ▾ Not assigned ▾ 1min effort [Comment](#) No tags ▾

Unexpected duplicate selector "a", first used at line 45 Why is this an issue? 6 months ago ▾ L1099 🔗 ⌵

Code Smell Major Open ▾ Not assigned ▾ 1min effort [Comment](#) No tags ▾

Unexpected duplicate selector "figure", first used at line 96 Why is this an issue? 6 months ago ▾ L1112 🔗 ⌵

Code Smell Major Open ▾ Not assigned ▾ 1min effort [Comment](#) No tags ▾

Unexpected duplicate selector "img", first used at line 90 Why is this an issue? 6 months ago ▾ L1115 🔗 ⌵

Code Smell Major Open ▾ Not assigned ▾ 1min effort [Comment](#) No tags ▾

Unexpected duplicate selector "hr", first used at line 99 Why is this an issue? 6 months ago ▾ L1146 🔗 ⌵

Code Smell Major Open ▾ Not assigned ▾ 1min effort [Comment](#) No tags ▾

Unexpected duplicate selector "code, kbd, pre, samp", first used at line 108 (no-duplicate-selectors) Why is this an issue? 6 months ago ▾ L1529 🔗 ⌵

Code Smell Major Open ▾ Not assigned ▾ 1min effort [Comment](#) No tags ▾

Unexpected duplicate selector "pre", first used at line 105 Why is this an issue? 6 months ago ▾ L1558 🔗 ⌵

Code Smell Major Open ▾ Not assigned ▾ 1min effort [Comment](#) No tags ▾

Unexpected duplicate selector "table", first used at line 186 Why is this an issue? 6 months ago ▾ L2349 🔗 ⌵

Code Smell Major Open ▾ Not assigned ▾ 1min effort [Comment](#) No tags ▾

Unexpected duplicate selector "fieldset", first used at line 171 Why is this an issue? 6 months ago ▾ L2589 🔗 ⌵

Code Smell Major Open ▾ Not assigned ▾ 1min effort [Comment](#) No tags ▾

Unexpected duplicate selector "legend", first used at line 176 Why is this an issue? 6 months ago ▾ L2595 🔗 ⌵

Ilustración 28: Vista parcial de las incidencias menores del código

En este caso, nuevamente se observa que la mayoría de los fallos detectados se encuentran en ficheros como "Bootstrap.css", que dan soporte a librerías, y que no se van a modificar. Sin embargo, también hay algo de información útil, como puede ser la siguiente:



Ilustración 29: Bugs en el código del fichero "About.aspx"

De esta forma, se hace interesante realizar modificaciones en este fichero, ya que ha sido desarrollado por el autor.

Al igual que en el caso del servicio de conversión, estos fallos se han dado en la versión de GitHub "Versión final", y ya aparecerán corregidos al entregar este proyecto.

Una vez visto el análisis de calidad del cliente web, se verá el último análisis realizado, que es el de la aplicación para realizar pruebas.

4.4.3. Aplicación para realizar pruebas

Dada la sencillez de esta aplicación, el análisis de calidad no ha puesto en evidencia muchos fallos, como se ve en la siguiente ilustración:

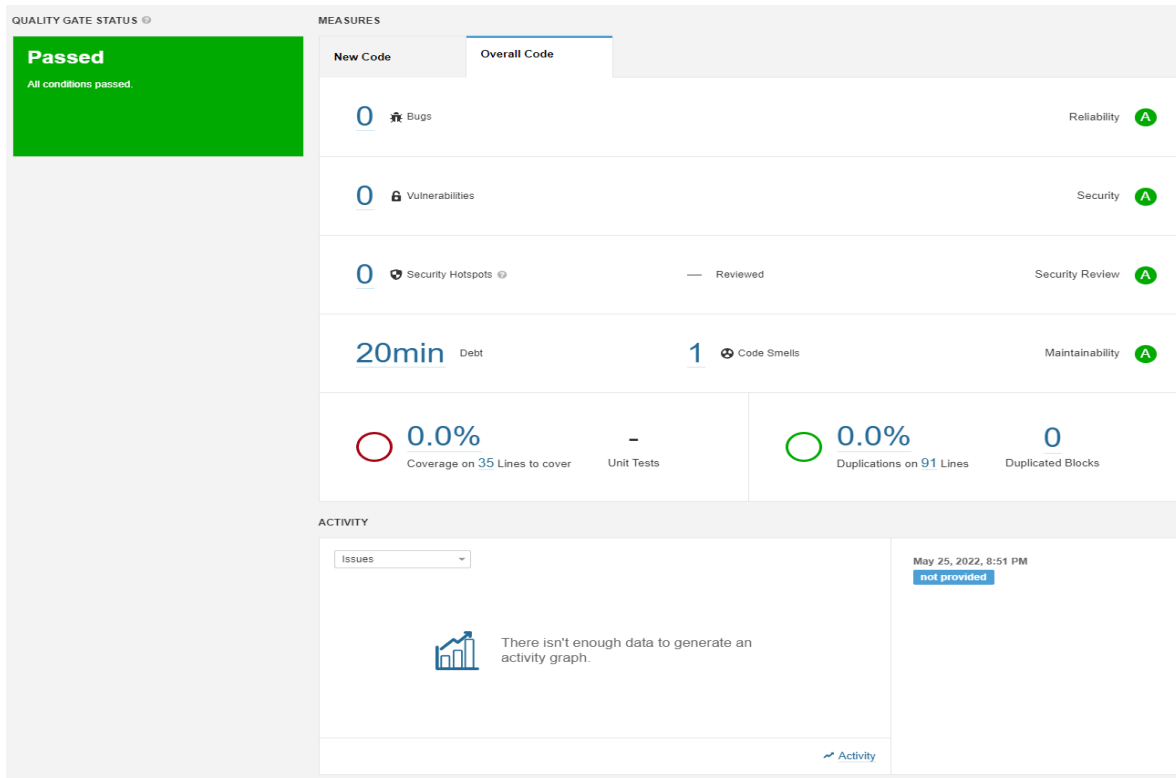


Ilustración 30: Resumen de los resultados del análisis de calidad en la aplicación para realizar pruebas

En este caso, solo se detectado una incidencia menor, que se puede observar en la siguiente ilustración:

Make the enclosing instance method 'static' or remove this set on the 'static' field.

Code Smell +1

1 of 1 shown

```

46     }
47     }
48     testNumbersButton.Enabled = true;
49     srInput.Close();
50     srOutput.Close();
51 } catch (Exception ex)
52 {
53     errorsLabel.Text = ex.Message;
54     if (srInput != null) srInput.Close();
55     if (srOutput != null) srOutput.Close();
56     testNumbersButton.Enabled = true;
57 }
58 }
59 /**
60  * <summary>Botón para elegir fichero de entrada.</summary>
61  */
62 private void buttonInputFile_Click(object sender, EventArgs e)
63 {
64     DialogResult result = folderBrowserDialog1.ShowDialog();
65     if (result == DialogResult.OK)
66     {
67         baseDirectory = folderBrowserDialog1.SelectedPath;
68
69         folderTextBox.Text = baseDirectory;
70         if (baseDirectory != "")
71         {
72             testNumbersButton.Enabled = true;
73         } else
74         {
75             testNumbersButton.Enabled = false;

```

Make the enclosing instance method 'static' or remove this set on the 'static' field. Why is this an issue? 16 days ago - L68 1

Code Smell Critical Open Not assigned 20min effort Comment multi-threading

Ilustración 31: Incidencia menor detectada en la aplicación para hacer pruebas

La sugerencia que se hace es nombrar el método que contiene esa línea de código como estático (algo que no se puede realizar por la naturaleza de los eventos en WinForms), o bien eliminar esa línea, lo cual no se va a hacer porque es necesaria. Por tanto, no se van a realizar correcciones en la aplicación para hacer test al servicio de conversión.

Una vez visto esto, concluye la parte del desarrollo del proyecto, y, en el siguiente apartado, se va a mencionar la normativa y legislación que debe seguir.

5. Normativa y legislación

En cuanto a la normativa y legislación que hay que cumplir, las exigencias no serán grandes, puesto que este proyecto no trata con datos personales ni guarda cookies o rastreadores en los navegadores de los usuarios.

Lo cierto es que un número, para algunas personas, puede ser considerado como dato personal, por ejemplo, si se refiere a su edad. Sin embargo, la inserción del número de por sí no constituye un dato personal, puesto que el usuario no se identifica en el sitio web para realizar la conversión, ni el número por sí solo representa un dato identificable de un usuario.

Recuérdese que la Ley Orgánica de Protección de Datos y Garantías de los Derechos Digitales [24] únicamente se aplica cuando aquellos que se van a tratar sean considerados datos personales. Para ello, se debe cumplir con uno de los siguientes requisitos: que sean identificados (por ejemplo, conociendo qué persona que los inserta) o identificables (por ejemplo, un número de DNI es identificable, al ser único y pertenecer a una persona – nótese que, una vez se tenga este número, se podrá saber a qué persona pertenece de forma sencilla, ya que es probable que aparezca en algún BOE) [25].

De todas formas, si una persona decidiera insertar datos personales, no se almacenarán en el servidor.

Las únicas normas que esta aplicación deberán cumplir serán las establecidas por la Real Academia Española. Así, la conversión de los números se deberá hacer de acuerdo con estas normas.

6. Conclusión y trabajos futuros

6.1. Conclusión

Con la realización de este trabajo, se han cumplido los objetivos iniciales planteados en “2.2. Objetivos iniciales”. Fundamentalmente, ya se tiene una aplicación web que permita convertir textos a números, que cuenta con una función de autocompletado para facilitar la inserción de estos. De esta forma, aquellas personas que tengan dificultades a la hora de aprender los números (niños o personas que no dominen el español), podrán hacerlo de una forma más sencilla y divertida.

Además, ahora se cuenta con un servicio que podrá ser aprovechado por otros investigadores para crear conversores más avanzados de números (escritos como textos) embebidos en textos, como los comentados en el apartado “1.1. Motivación”.

En esencia, con este proyecto se ha conseguido crear algo que no existía y que, sin duda, será muy útil.

La realización de este proyecto no únicamente ha aportado a la sociedad en un conjunto. El autor de este proyecto también ha conseguido aprender muchos conceptos fundamentales de cara al futuro profesional.

El abanico de cosas que se han aprendido es muy grande, puesto que abarca desde la mejora en las habilidades de programación en las tecnologías mencionadas hasta el aprendizaje sobre la gestión de un proyecto informático.

6.2. Trabajos futuros

Con este Trabajo de Fin de Grado, se abre un gran abanico de posibilidades para realizar trabajos futuros.

Se propone, a partir del servicio creado en este proyecto, realizar un procesador de textos que contengan nomenclatura numérica en español. Este será especialmente útil, por ejemplo, para fines estadísticos. Por ejemplo, obteniendo distintas sentencias judiciales, se podrá averiguar la media de días de multa que se dictaminen.

Otra posibilidad del trabajo futuro sería realizar un sistema de cuestionarios online (por ejemplo, para realizar exámenes) donde el usuario no esté obligado a insertar los números con cifras, sino que también tengan la opción de insertarlos como textos. A pesar de que, a priori, pudiera no parecer muy útil hacer algo así, un sistema de cuestionarios como el propuesto podría mejorar la exactitud de los números introducidos en los mismos, fundamentalmente, porque es mucho más difícil equivocarse escribiendo un número con nomenclatura numérica que utilizando cifras, ya que, al presionar solo una cifra por error, el significado del dato introducido cambia muy notoriamente (Y no se puede determinar si se trata de un dato introducido a propósito o por error), mientras que, el error que se pueda generar al introducir una letra de más, únicamente hará que se solicite al usuario revisar el número y volver a insertarlo.

En definitiva, hay muchas opciones que se abren a partir de la realización de este proyecto.

7. Fuentes de información

- [1] «STS, 26 de Septiembre de 2014 | VLex,» 26 09 2014. [En línea]. Available: <https://vlex.es/vid/-538756578>. [Último acceso: 09 03 2022].
- [2] F. Carreras-Riudavets, G. Rodríguez-Rodríguez y Z. Hernández-Figueroa, «Números TIP,» 2021. [En línea]. Available: <https://tulengua.es/numeros-texto/>.
- [3] CoolConversion.com, «Words to Numbers Converter,» [En línea]. Available: https://coolconversion.com/math/words-to-numbers-converter/_two%20quadrillion_in-numbers. [Último acceso: 29 03 2022].
- [4] Microsoft, «Microsoft Word,» [En línea]. Available: <https://www.microsoft.com/es-es/microsoft-365/word>. [Último acceso: 29 03 2022].
- [5] «OpenAI,» [En línea]. Available: <https://openai.com/>. [Último acceso: 29 03 2022].
- [6] «Teorema de Euclides | Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Teorema_de_Euclides-Euler. [Último acceso: 29 03 2022].
- [7] Microsoft Corporation, «UInt64 Struct,» [En línea]. Available: <https://docs.microsoft.com/en-us/dotnet/api/system.uint64?view=net-6.0>. [Último acceso: 12 04 2022].
- [8] Real Academia Española; Asociación de Academias de la Lengua Española, Diccionario panhispánico de dudas, Madrid: Penguin Random House, 2015.
- [9] Microsoft Corporation, «¿Qué es Windows Communication Foundation?,» 23 03 2022. [En línea]. Available: <https://docs.microsoft.com/es-es/dotnet/framework/wcf/whats-wcf>. [Último acceso: 29 03 2022].
- [10] Microsoft Corporation, «Herramientas y servicios de desarrollo | Visual Studio Enterprise,» [En línea]. Available: <https://visualstudio.microsoft.com/es/vs/enterprise/>. [Último acceso: 13 05 2022].
- [11] Microsoft Corporation, «The official IIS website,» [En línea]. Available: <https://www.iis.net>. [Último acceso: 13 05 2022].
- [12] A. Walker, «SOAP Web Services Tutorial: What is SOAP Protocol? EXAMPLE,» 16 04 2022. [En línea]. Available: <https://www.guru99.com/soap-simple-object-access-protocol.html>. [Último acceso: 13 05 2022].
- [13] Microsoft Corporation, «UpdatePanel Class,» [En línea]. Available: <https://docs.microsoft.com/en-us/dotnet/api/system.web.ui.updatepanel?view=netframework-4.8>. [Último acceso: 12 04 2022].

- [14] IONOS, «El modelo en cascada: desarrollo secuencial de software,» 11 03 2019. [En línea]. Available: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>. [Último acceso: 12 04 2022].
- [15] «TailwindCSS,» [En línea]. Available: <https://tailwindcss.com/>. [Último acceso: 29 03 2022].
- [16] A. R. Lazar, «Cliente web,» 06 2022. [En línea]. Available: <http://conversor-numeros-iatext.azurewebsites.net>. [Último acceso: 13 04 2022].
- [17] Mozilla Corporation, «Deep copy | MDN Docs,» 14 03 2022. [En línea]. Available: https://developer.mozilla.org/en-US/docs/Glossary/Deep_copy. [Último acceso: 12 04 2022].
- [18] A. R. Lazar, «Servicio de conversión de números,» [En línea]. Available: <https://tftservice20220201172317.azurewebsites.net/Service1.svc>. [Último acceso: 12 04 2022].
- [19] SonarQube, «SonarQube,» [En línea]. Available: <https://www.sonarqube.org>. [Último acceso: 25 05 2022].
- [20] Microsoft Corporation, «MSBuild,» [En línea]. Available: <https://docs.microsoft.com/en-us/visualstudio/msbuild/msbuild?view=vs-2022>. [Último acceso: 25 05 2022].
- [21] SonarQube, «SonarScanner for .NET,» [En línea]. Available: <https://docs.sonarqube.org/latest/analysis/scan/sonarscanner-for-msbuild/>. [Último acceso: 25 05 2022].
- [22] SonarQube, «Install the Server,» [En línea]. Available: <https://docs.sonarqube.org/latest/setup/install-server/>. [Último acceso: 25 05 2022].
- [23] N. Dao, «VISUAL STUDIO & MSBUILD TUTORIAL,» [En línea]. Available: <https://gist.github.com/nicolasdao/b1623cb3220be3b6dc8fe04b2b2bfe32>. [Último acceso: 25 05 2022].
- [24] Boletín Oficial del Estado, «LOPDyGDD | BOE,» 06 12 2018. [En línea]. Available: <https://www.boe.es/boe/dias/2018/12/06/pdfs/BOE-A-2018-16673.pdf>. [Último acceso: 13 04 2022].
- [25] Comisión Europea, «¿Qué son los datos personales?,» [En línea]. Available: https://ec.europa.eu/info/law/law-topic/data-protection/reform/what-personal-data_es. [Último acceso: 13 04 2022].
- [26] Microsoft Corporation, «Información general de ASP.NET,» 29 03 2022. [En línea]. Available: <https://docs.microsoft.com/es-es/aspnet/overview>. [Último acceso: 29 03 2022].

8. Anexos

8.1. Anexo I – Contenidos del almacén de números cero;0

un;1

uno;1

dos;2

mitad;/2

medio;/2

tres;3

tercio;/3

tercero;/3

cuatro;4

cuarto;/4

cinco;5

quinto;/5

seis;6

sexto;/6

siete;7

séptimo;/7

ocho;8

octavo;/8

nueve;9

noveno;/9

diez;10

décimo;/10

once;11

undécimo;/11
doce;12
duodécimo;/12
trece;13
catorce;14
quince;15
dieciséis;16
diecisiete;17
dieciocho;18
diecinueve;19
veinte;20
vigésimo;/20
veintiuno;21
veintiún;21
veintidós;22
veintitrés;23
veinticuatro;24
veinticinco;25
veintiséis;26
veintisiete;27
veintiocho;28
veintinueve;29
treinta;30
trigésimo;/30
treintaiún;31

treintaiuno;31
treintaidos;32
treintaitrés;33
treintaicuatros;34
treintaicinco;35
treintaiséis;36
treintaisiete;37
treintaiocho;38
treintainueve;39
cuarenta;40
cuadragésimo;/40
cuarentaiún;41
cuarentaiuno;41
cuarentaidos;42
cuarentaitrés;43
cuarentaicuatros;44
cuarentaicinco;45
cuarentaiséis;46
cuarentaisiete;47
cuarentaiocho;48
cuarentainueve;49
cincuenta;50
quincuagésimo;/50
cincuentaiún;51
cincuentaiuno;51

cincuentaidos;52
cincuentaitrés;53
cincuentaicuatros;54
cincuentaicinco;55
cincuentaiséis;56
cincuentaisiete;57
cincuentaiocho;58
cincuntainueve;59
sesenta;60
sexagésimo;/60
sesentaiún;61
sesentauno;61
sesentaidos;62
sesentaitrés;63
sesentaicuatros;64
sesentaicinco;65
sesentaiséis;66
sesentaisiete;67
sesentaiocho;68
sesuntainueve;69
setenta;70
septuagésimo;/70
setentaiún;71
setentauno;71
setentaidos;72

setentaitrés;73
setentaicuatro;74
setentaicinco;75
setentaiséis;76
setentaisiete;77
setentaiocho;78
setentainueve;79
ochenta;80
octogésimo;/80
ochentaiún;81
ochentaiuno;81
ochentaidos;82
ochentaitrés;83
ochentaicuatro;84
ochentaicinco;85
ochentaiséis;86
ochentaisiete;87
ochentaiocho;88
ochentainueve;89
noventa;90
nonagésimo;/90
noventaiún;91
noventaiuno;91
noventaidos;92
noventaitrés;93

noventaicuatro;94
noventaicinco;95
noventaiséis;96
noventaísiete;97
noventaiocho;98
noventainueve;99
cien;100
ciento;100
centena;100
centésimo;/100
doscientos;200
trescientos;300
cuatrocientos;400
quinientos;500
seiscientos;600
setecientos;700
ochocientos;800
novecientos;900
mil;1000
millón;1000000
millardo;1000000000
billón;1000000000000
trillón;1000000000000000000
cuatrillón;1000000000000000000000000
quintillón;10000000000000000000000000000000

sextillón;1000000000000000000000000000000000

septillón;100000000000000000000000000000000000

octillón;100000000000000000000000000000000000000

nonillón;1000000000000000000000000000000000000000

decillón;1000

undecillón;100

duodecillón;100

tredecillón;100

cuatordecillón;100

quindecillón;1000

sexdecillón;1000

septendecillón;1000

octodecillón;1000

novendecillón;1000

vigintillón;1000

8.2. Anexo II – Contenidos del fichero de entrada para realizar las pruebas

0:cero

1:uno

2:dos

3:tres

4:cuatro

5:cinco

6:seis

7:siete

8:ocho

9:nueve

10:diez

11:once

12:doce

13:trece

14:catorce

15:quince

16:dieciséis

17:diecisiete

18:dieciocho

19:diecinueve

20:veinte

21:veintiuno

22:veintidós

23:veintitrés

24:veinticuatro

25:veinticinco

26:veintiséis

27:veintisiete

28:veintiocho

29:veintinueve

30:treinta
31:treinta y uno
32:treinta y dos
33:treinta y tres
34:treinta y cuatro
35:treinta y cinco
36:treinta y seis
37:treinta y siete
38:treinta y ocho
39:treinta y nueve
40:cuarenta
41:cuarenta y uno
42:cuarenta y dos
43:cuarenta y tres
44:cuarenta y cuatro
45:cuarenta y cinco
46:cuarenta y seis
47:cuarenta y siete
48:cuarenta y ocho
49:cuarenta y nueve
50:cincuenta
51:cincuenta y uno
52:cincuenta y dos
53:cincuenta y tres
54:cincuenta y cuatro
55:cincuenta y cinco
56:cincuenta y seis
57:cincuenta y siete
58:cincuenta y ocho
59:cincuenta y nueve
60:sesenta

61:sesenta y uno
62:sesenta y dos
63:sesenta y tres
64:sesenta y cuatro
65:sesenta y cinco
66:sesenta y seis
67:sesenta y siete
68:sesenta y ocho
69:sesenta y nueve
70:setenta
71:setenta y uno
72:setenta y dos
73:setenta y tres
74:setenta y cuatro
75:setenta y cinco
76:setenta y seis
77:setenta y siete
78:setenta y ocho
79:setenta y nueve
80:ochenta
81:ochenta y uno
82:ochenta y dos
83:ochenta y tres
84:ochenta y cuatro
85:ochenta y cinco
86:ochenta y seis
87:ochenta y siete
88:ochenta y ocho
89:ochenta y nueve
90:noventa
91:noventa y uno

92:noventa y dos
93:noventa y tres
94:noventa y cuatro
95:noventa y cinco
96:noventa y seis
97:noventa y siete
98:noventa y ocho
99:noventa y nueve
100:cien
101:ciento uno
110:ciento diez
111:ciento once
122:ciento veintidós
168:ciento sesenta y ocho
198:ciento noventa y ocho
211:doscientos once
422:cuatrocientos veintidós
689:seiscientos ochenta y nueve
712:setecientos doce
879:ochocientos setenta y nueve
919:novecientos diecinueve
1000:mil
1001:mil uno
1010:mil diez
1011:mil once
1100:mil cien
1101:mil ciento uno
1110:mil ciento diez
1111:mil ciento once
2389:dos mil trescientos ochenta y nueve
4001:cuatro mil uno

5087:cinco mil ochenta y siete
6938:seis mil novecientos treinta y ocho
7907:siete mil novecientos siete
8004:ocho mil cuatro
9019:nueve mil diecinueve
10 000:diez mil
11 000:once mil
20 939:veinte mil novecientos treinta y nueve
30 001:treinta mil uno
49 012:cuarenta y nueve mil doce
58 700:cincuenta y ocho mil setecientos
69 810:sesenta y nueve mil ochocientos diez
78 569:setenta y ocho mil quinientos sesenta y nueve
81 293:ochenta y un mil doscientos noventa y tres
94 182:noventa y cuatro mil ciento ochenta y dos
2 093 129:dos millones noventa y tres mil ciento veintinueve
20 931 291:veinte millones novecientos treinta y un mil doscientos noventa y uno
20 931 294:veinte millones novecientos treinta y un mil doscientos noventa y cuatro
209 312 940:doscientos nueve millones trescientos doce mil novecientos cuarenta
2 093 129 406:dos mil noventa y tres millones ciento veintinueve mil cuatrocientos seis
20 931 294 061:veinte mil novecientos treinta y un millones doscientos noventa y cuatro mil sesenta y uno
209 312 940 617:doscientos nueve mil trescientos doce millones novecientos cuarenta mil seiscientos diecisiete
2 093 129 406 174:dos billones noventa y tres mil ciento veintinueve millones cuatrocientos seis mil ciento setenta y cuatro
20 931 294 061 748:veinte billones novecientos treinta y un mil doscientos noventa y cuatro millones sesenta y un mil setecientos cuarenta y ocho
209 312 940 617 487:doscientos nueve billones trescientos doce mil novecientos cuarenta millones seiscientos diecisiete mil cuatrocientos ochenta y siete
2 093 129 406 174 872:dos mil noventa y tres billones ciento veintinueve mil cuatrocientos seis millones ciento setenta y cuatro mil ochocientos setenta y dos

20 931 294 061 748 727:veinte mil novecientos treinta y un billones doscientos noventa y cuatro mil sesenta y un millones setecientos cuarenta y ocho mil setecientos veintisiete

209 312 940 617 487 275:doscientos nueve mil trescientos doce billones novecientos cuarenta mil seiscientos diecisiete millones cuatrocientos ochenta y siete mil doscientos setenta y cinco

2 093 129 406 174 872 757:dos trillones noventa y tres mil ciento veintinueve billones cuatrocientos seis mil ciento setenta y cuatro millones ochocientos setenta y dos mil setecientos cincuenta y siete

20 931 294 061 748 727 572:veinte trillones novecientos treinta y un mil doscientos noventa y cuatro billones sesenta y un mil setecientos cuarenta y ocho millones setecientos veintisiete mil quinientos setenta y dos

209 312 940 617 487 275 727:doscientos nueve trillones trescientos doce mil novecientos cuarenta billones seiscientos diecisiete mil cuatrocientos ochenta y siete millones doscientos setenta y cinco mil setecientos veintisiete

2 093 129 406 174 872 757 271:dos mil noventa y tres trillones ciento veintinueve mil cuatrocientos seis billones ciento setenta y cuatro mil ochocientos setenta y dos millones setecientos cincuenta y siete mil doscientos setenta y uno

20 931 294 061 748 727 572 717:veinte mil novecientos treinta y un trillones doscientos noventa y cuatro mil sesenta y un billones setecientos cuarenta y ocho mil setecientos veintisiete millones quinientos setenta y dos mil setecientos diecisiete

209 312 940 617 487 275 727 174:doscientos nueve mil trescientos doce trillones novecientos cuarenta mil seiscientos diecisiete billones cuatrocientos ochenta y siete mil doscientos setenta y cinco millones setecientos veintisiete mil ciento setenta y cuatro

2 093 129 406 174 872 757 271 746:dos cuatrillones noventa y tres mil ciento veintinueve trillones cuatrocientos seis mil ciento setenta y cuatro billones ochocientos setenta y dos mil setecientos cincuenta y siete millones doscientos setenta y un mil setecientos cuarenta y seis

20 931 294 061 748 727 572 717 460:veinte cuatrillones novecientos treinta y un mil doscientos noventa y cuatro trillones sesenta y un mil setecientos cuarenta y ocho billones setecientos veintisiete mil quinientos setenta y dos millones setecientos diecisiete mil cuatrocientos sesenta

209 312 940 617 487 275 727 174 607:doscientos nueve cuatrillones trescientos doce mil novecientos cuarenta trillones seiscientos diecisiete mil cuatrocientos ochenta y siete billones doscientos setenta y cinco mil setecientos veintisiete millones ciento setenta y cuatro mil seiscientos siete

2 093 129 406 174 872 757 271 746 072:dos mil noventa y tres cuatrillones ciento veintinueve mil cuatrocientos seis trillones ciento setenta y cuatro mil ochocientos setenta y dos billones setecientos cincuenta y siete mil doscientos setenta y un millones setecientos cuarenta y seis mil setenta y dos

20 931 294 061 748 727 572 717 460 724:veinte mil novecientos treinta y un cuatrillones doscientos noventa y cuatro mil sesenta y un trillones setecientos

cuarenta y ocho mil setecientos veintisiete billones quinientos setenta y dos mil setecientos diecisiete millones cuatrocientos sesenta mil setecientos veinticuatro

209 312 940 617 487 275 727 174 607 245: doscientos nueve mil trescientos doce cuatrillones novecientos cuarenta mil seiscientos diecisiete trillones cuatrocientos ochenta y siete mil doscientos setenta y cinco billones setecientos veintisiete mil ciento setenta y cuatro millones seiscientos siete mil doscientos cuarenta y cinco

2 093 129 406 174 872 757 271 746 072 451: dos quintillones noventa y tres mil ciento veintinueve cuatrillones cuatrocientos seis mil ciento setenta y cuatro trillones ochocientos setenta y dos mil setecientos cincuenta y siete billones doscientos setenta y un mil setecientos cuarenta y seis millones setenta y dos mil cuatrocientos cincuenta y uno

20 931 294 061 748 727 572 717 460 724 517: veinte quintillones novecientos treinta y un mil doscientos noventa y cuatro cuatrillones sesenta y un mil setecientos cuarenta y ocho trillones setecientos veintisiete mil quinientos setenta y dos billones setecientos diecisiete mil cuatrocientos sesenta millones setecientos veinticuatro mil quinientos diecisiete

209 312 940 617 487 275 727 174 607 245 174: doscientos nueve quintillones trescientos doce mil novecientos cuarenta cuatrillones seiscientos diecisiete mil cuatrocientos ochenta y siete trillones doscientos setenta y cinco mil setecientos veintisiete billones ciento setenta y cuatro mil seiscientos siete millones doscientos cuarenta y cinco mil ciento setenta y cuatro

2 093 129 406 174 872 757 271 746 072 451 745: dos mil noventa y tres quintillones ciento veintinueve mil cuatrocientos seis cuatrillones ciento setenta y cuatro mil ochocientos setenta y dos trillones setecientos cincuenta y siete mil doscientos setenta y un billones setecientos cuarenta y seis mil setenta y dos millones cuatrocientos cincuenta y un mil setecientos cuarenta y cinco

20 931 294 061 748 727 572 717 460 724 517 455: veinte mil novecientos treinta y un quintillones doscientos noventa y cuatro mil sesenta y un cuatrillones setecientos cuarenta y ocho mil setecientos veintisiete trillones quinientos setenta y dos mil setecientos diecisiete billones cuatrocientos sesenta mil setecientos veinticuatro millones quinientos diecisiete mil cuatrocientos cincuenta y cinco

209 312 940 617 487 275 727 174 607 245 174 552: doscientos nueve mil trescientos doce quintillones novecientos cuarenta mil seiscientos diecisiete cuatrillones cuatrocientos ochenta y siete mil doscientos setenta y cinco trillones setecientos veintisiete mil ciento setenta y cuatro billones seiscientos siete mil doscientos cuarenta y cinco millones ciento setenta y cuatro mil quinientos cincuenta y dos

2 093 129 406 174 872 757 271 746 072 451 745 528: dos sextillones noventa y tres mil ciento veintinueve quintillones cuatrocientos seis mil ciento setenta y cuatro cuatrillones ochocientos setenta y dos mil setecientos cincuenta y siete trillones doscientos setenta y un mil setecientos cuarenta y seis billones setenta y dos mil cuatrocientos cincuenta y un millones setecientos cuarenta y cinco mil quinientos veintiocho

20 931 294 061 748 727 572 717 460 724 517 455 285: veinte sextillones novecientos treinta y un mil doscientos noventa y cuatro quintillones sesenta y un mil setecientos cuarenta y ocho cuatrillones setecientos veintisiete mil quinientos

setenta y dos trillones setecientos diecisiete mil cuatrocientos sesenta billones setecientos veinticuatro mil quinientos diecisiete millones cuatrocientos cincuenta y cinco mil doscientos ochenta y cinco

209 312 940 617 487 275 727 174 607 245 174 552 858:doscientos nueve sextillones trescientos doce mil novecientos cuarenta quintillones seiscientos diecisiete mil cuatrocientos ochenta y siete cuatrillones doscientos setenta y cinco mil setecientos veintisiete trillones ciento setenta y cuatro mil seiscientos siete billones doscientos cuarenta y cinco mil ciento setenta y cuatro millones quinientos cincuenta y dos mil ochocientos cincuenta y ocho

2 093 129 406 174 872 757 271 746 072 451 745 528 581:dos mil noventa y tres sextillones ciento veintinueve mil cuatrocientos seis quintillones ciento setenta y cuatro mil ochocientos setenta y dos cuatrillones setecientos cincuenta y siete mil doscientos setenta y un trillones setecientos cuarenta y seis mil setenta y dos billones cuatrocientos cincuenta y un mil setecientos cuarenta y cinco millones quinientos veintiocho mil quinientos ochenta y uno

20 931 294 061 748 727 572 717 460 724 517 455 285 811:veinte mil novecientos treinta y un sextillones doscientos noventa y cuatro mil sesenta y un quintillones setecientos cuarenta y ocho mil setecientos veintisiete cuatrillones quinientos setenta y dos mil setecientos diecisiete trillones cuatrocientos sesenta mil setecientos veinticuatro billones quinientos diecisiete mil cuatrocientos cincuenta y cinco millones doscientos ochenta y cinco mil ochocientos once

209 312 940 617 487 275 727 174 607 245 174 552 858 111:doscientos nueve mil trescientos doce sextillones novecientos cuarenta mil seiscientos diecisiete quintillones cuatrocientos ochenta y siete mil doscientos setenta y cinco cuatrillones setecientos veintisiete mil ciento setenta y cuatro trillones seiscientos siete mil doscientos cuarenta y cinco billones ciento setenta y cuatro mil quinientos cincuenta y dos millones ochocientos cincuenta y ocho mil ciento once

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110:dos septillones noventa y tres mil ciento veintinueve sextillones cuatrocientos seis mil ciento setenta y cuatro quintillones ochocientos setenta y dos mil setecientos cincuenta y siete cuatrillones doscientos setenta y un mil setecientos cuarenta y seis trillones setenta y dos mil cuatrocientos cincuenta y un billones setecientos cuarenta y cinco mil quinientos veintiocho millones quinientos ochenta y un mil ciento diez

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102:veinte septillones novecientos treinta y un mil doscientos noventa y cuatro sextillones sesenta y un mil setecientos cuarenta y ocho quintillones setecientos veintisiete mil quinientos setenta y dos cuatrillones setecientos diecisiete mil cuatrocientos sesenta trillones setecientos veinticuatro mil quinientos diecisiete billones cuatrocientos cincuenta y cinco mil doscientos ochenta y cinco millones ochocientos once mil ciento dos

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027:doscientos nueve septillones trescientos doce mil novecientos cuarenta sextillones seiscientos diecisiete mil cuatrocientos ochenta y siete quintillones doscientos setenta y cinco mil setecientos veintisiete cuatrillones ciento setenta y cuatro mil seiscientos siete trillones doscientos cuarenta y cinco mil ciento setenta y cuatro billones quinientos cincuenta y dos mil ochocientos cincuenta y ocho millones ciento once mil veintisiete

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278:dos mil noventa y tres septillones ciento veintinueve mil cuatrocientos seis sextillones ciento setenta y cuatro mil ochocientos setenta y dos quintillones setecientos cincuenta y siete mil doscientos setenta y un cuatrillones setecientos cuarenta y seis mil setenta y dos trillones cuatrocientos cincuenta y un mil setecientos cuarenta y cinco billones quinientos veintiocho mil quinientos ochenta y un millones ciento diez mil doscientos setenta y ocho

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781:veinte mil novecientos treinta y un septillones doscientos noventa y cuatro mil sesenta y un sextillones setecientos cuarenta y ocho mil setecientos veintisiete quintillones quinientos setenta y dos mil setecientos diecisiete cuatrillones cuatrocientos sesenta mil setecientos veinticuatro trillones quinientos diecisiete mil cuatrocientos cincuenta y cinco billones doscientos ochenta y cinco mil ochocientos once millones ciento dos mil setecientos ochenta y uno

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811:doscientos nueve mil trescientos doce septillones novecientos cuarenta mil seiscientos diecisiete sextillones cuatrocientos ochenta y siete mil doscientos setenta y cinco quintillones setecientos veintisiete mil ciento setenta y cuatro cuatrillones seiscientos siete mil doscientos cuarenta y cinco trillones ciento setenta y cuatro mil quinientos cincuenta y dos billones ochocientos cincuenta y ocho mil ciento once millones veintisiete mil ochocientos once

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114:dos octillones noventa y tres mil ciento veintinueve septillones cuatrocientos seis mil ciento setenta y cuatro sextillones ochocientos setenta y dos mil setecientos cincuenta y siete quintillones doscientos setenta y un mil setecientos cuarenta y seis cuatrillones setenta y dos mil cuatrocientos cincuenta y un trillones setecientos cuarenta y cinco mil quinientos veintiocho billones quinientos ochenta y un mil ciento diez millones doscientos setenta y ocho mil ciento catorce

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145:veinte octillones novecientos treinta y un mil doscientos noventa y cuatro septillones sesenta y un mil setecientos cuarenta y ocho sextillones setecientos veintisiete mil quinientos setenta y dos quintillones setecientos diecisiete mil cuatrocientos sesenta cuatrillones setecientos veinticuatro mil quinientos diecisiete trillones cuatrocientos cincuenta y cinco mil doscientos ochenta y cinco billones ochocientos once mil ciento dos millones setecientos ochenta y un mil ciento cuarenta y cinco

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454:doscientos nueve octillones trescientos doce mil novecientos cuarenta septillones seiscientos diecisiete mil cuatrocientos ochenta y siete sextillones doscientos setenta y cinco mil setecientos veintisiete quintillones ciento setenta y cuatro mil seiscientos siete cuatrillones doscientos cuarenta y cinco mil ciento setenta y cuatro trillones quinientos cincuenta y dos mil ochocientos cincuenta y ocho billones ciento once mil veintisiete millones ochocientos once mil cuatrocientos cincuenta y cuatro

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548:dos mil noventa y tres octillones ciento veintinueve mil cuatrocientos seis septillones ciento setenta y cuatro mil ochocientos setenta y dos sextillones setecientos cincuenta y siete mil doscientos setenta y un quintillones setecientos cuarenta y seis mil setenta y dos cuatrillones cuatrocientos cincuenta y un mil setecientos cuarenta y

cinco trillones quinientos veintiocho mil quinientos ochenta y un billones ciento diez mil doscientos setenta y ocho millones ciento catorce mil quinientos cuarenta y ocho

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488:veinte mil novecientos treinta y un octillones doscientos noventa y cuatro mil sesenta y un septillones setecientos cuarenta y ocho mil setecientos veintisiete sextillones quinientos setenta y dos mil setecientos diecisiete quintillones cuatrocientos sesenta mil setecientos veinticuatro cuatrillones quinientos diecisiete mil cuatrocientos cincuenta y cinco trillones doscientos ochenta y cinco mil ochocientos once billones ciento dos mil setecientos ochenta y un millones ciento cuarenta y cinco mil cuatrocientos ochenta y ocho

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881:doscientos nueve mil trescientos doce octillones novecientos cuarenta mil seiscientos diecisiete septillones cuatrocientos ochenta y siete mil doscientos setenta y cinco sextillones setecientos veintisiete mil ciento setenta y cuatro quintillones seiscientos siete mil doscientos cuarenta y cinco cuatrillones ciento setenta y cuatro mil quinientos cincuenta y dos trillones ochocientos cincuenta y ocho mil ciento once billones veintisiete mil ochocientos once millones cuatrocientos cincuenta y cuatro mil ochocientos ochenta y uno

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816:dos nonillones noventa y tres mil ciento veintinueve octillones cuatrocientos seis mil ciento setenta y cuatro septillones ochocientos setenta y dos mil setecientos cincuenta y siete sextillones doscientos setenta y un mil setecientos cuarenta y seis quintillones setenta y dos mil cuatrocientos cincuenta y un cuatrillones setecientos cuarenta y cinco mil quinientos veintiocho trillones quinientos ochenta y un mil ciento diez billones doscientos setenta y ocho mil ciento catorce millones quinientos cuarenta y ocho mil ochocientos dieciséis

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488 168:veinte nonillones novecientos treinta y un mil doscientos noventa y cuatro octillones sesenta y un mil setecientos cuarenta y ocho septillones setecientos veintisiete mil quinientos setenta y dos sextillones setecientos diecisiete mil cuatrocientos sesenta quintillones setecientos veinticuatro mil quinientos diecisiete cuatrillones cuatrocientos cincuenta y cinco mil doscientos ochenta y cinco trillones ochocientos once mil ciento dos billones setecientos ochenta y un mil ciento cuarenta y cinco millones cuatrocientos ochenta y ocho mil ciento sesenta y ocho

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881 688:doscientos nueve nonillones trescientos doce mil novecientos cuarenta octillones seiscientos diecisiete mil cuatrocientos ochenta y siete septillones doscientos setenta y cinco mil setecientos veintisiete sextillones ciento setenta y cuatro mil seiscientos siete quintillones doscientos cuarenta y cinco mil ciento setenta y cuatro cuatrillones quinientos cincuenta y dos mil ochocientos cincuenta y ocho trillones ciento once mil veintisiete billones ochocientos once mil cuatrocientos cincuenta y cuatro millones ochocientos ochenta y un mil seiscientos ochenta y ocho

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816 885:dos mil noventa y tres nonillones ciento veintinueve mil cuatrocientos seis octillones ciento setenta y cuatro mil ochocientos setenta y dos septillones setecientos cincuenta y siete mil doscientos setenta y un sextillones setecientos cuarenta y seis mil setenta

y dos quintillones cuatrocientos cincuenta y un mil setecientos cuarenta y cinco
cuatrillones quinientos veintiocho mil quinientos ochenta y un trillones ciento diez
mil doscientos setenta y ocho billones ciento catorce mil quinientos cuarenta y ocho
millones ochocientos dieciséis mil ochocientos ochenta y cinco

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488 168 858:veinte
mil novecientos treinta y un nonillones doscientos noventa y cuatro mil sesenta y un
octillones setecientos cuarenta y ocho mil setecientos veintisiete septillones
quinientos setenta y dos mil setecientos diecisiete sextillones cuatrocientos sesenta
mil setecientos veinticuatro quintillones quinientos diecisiete mil cuatrocientos
cincuenta y cinco cuatrillones doscientos ochenta y cinco mil ochocientos once
trillones ciento dos mil setecientos ochenta y un billones ciento cuarenta y cinco
mil cuatrocientos ochenta y ocho millones ciento sesenta y ocho mil ochocientos
cincuenta y ocho

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881 688
584:doscientos nueve mil trescientos doce nonillones novecientos cuarenta mil
seiscientos diecisiete octillones cuatrocientos ochenta y siete mil doscientos
setenta y cinco septillones setecientos veintisiete mil ciento setenta y cuatro
sextillones seiscientos siete mil doscientos cuarenta y cinco quintillones ciento
setenta y cuatro mil quinientos cincuenta y dos cuatrillones ochocientos cincuenta y
ocho mil ciento once trillones veintisiete mil ochocientos once billones
cuatrocientos cincuenta y cuatro mil ochocientos ochenta y un millones seiscientos
ochenta y ocho mil quinientos ochenta y cuatro

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816 885 843:dos
decillones noventa y tres mil ciento veintinueve nonillones cuatrocientos seis mil
ciento setenta y cuatro octillones ochocientos setenta y dos mil setecientos
cincuenta y siete septillones doscientos setenta y un mil setecientos cuarenta y seis
sextillones setenta y dos mil cuatrocientos cincuenta y un quintillones setecientos
cuarenta y cinco mil quinientos veintiocho cuatrillones quinientos ochenta y un mil
ciento diez trillones doscientos setenta y ocho mil ciento catorce billones
quinientos cuarenta y ocho mil ochocientos dieciséis millones ochocientos ochenta y
cinco mil ochocientos cuarenta y tres

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488 168 858
435:veinte decillones novecientos treinta y un mil doscientos noventa y cuatro
nonillones sesenta y un mil setecientos cuarenta y ocho octillones setecientos
veintisiete mil quinientos setenta y dos septillones setecientos diecisiete mil
cuatrocientos sesenta sextillones setecientos veinticuatro mil quinientos diecisiete
quintillones cuatrocientos cincuenta y cinco mil doscientos ochenta y cinco
cuatrillones ochocientos once mil ciento dos trillones setecientos ochenta y un mil
ciento cuarenta y cinco billones cuatrocientos ochenta y ocho mil ciento sesenta y
ocho millones ochocientos cincuenta y ocho mil cuatrocientos treinta y cinco

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881 688 584
354:doscientos nueve decillones trescientos doce mil novecientos cuarenta nonillones
seiscientos diecisiete mil cuatrocientos ochenta y siete octillones doscientos
setenta y cinco mil setecientos veintisiete septillones ciento setenta y cuatro mil
seiscientos siete sextillones doscientos cuarenta y cinco mil ciento setenta y cuatro
quintillones quinientos cincuenta y dos mil ochocientos cincuenta y ocho cuatrillones
ciento once mil veintisiete trillones ochocientos once mil cuatrocientos cincuenta y

cuatro billones ochocientos ochenta y un mil seiscientos ochenta y ocho millones quinientos ochenta y cuatro mil trescientos cincuenta y cuatro

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816 885 843 546:dos mil noventa y tres decillones ciento veintinueve mil cuatrocientos seis nonillones ciento setenta y cuatro mil ochocientos setenta y dos octillones setecientos cincuenta y siete mil doscientos setenta y un septillones setecientos cuarenta y seis mil setenta y dos sextillones cuatrocientos cincuenta y un mil setecientos cuarenta y cinco quintillones quinientos veintiocho mil quinientos ochenta y un cuatrillones ciento diez mil doscientos setenta y ocho trillones ciento catorce mil quinientos cuarenta y ocho billones ochocientos dieciséis mil ochocientos ochenta y cinco millones ochocientos cuarenta y tres mil quinientos cuarenta y seis

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488 168 858 435 461:veinte mil novecientos treinta y un decillones doscientos noventa y cuatro mil sesenta y un nonillones setecientos cuarenta y ocho mil setecientos veintisiete octillones quinientos setenta y dos mil setecientos diecisiete septillones cuatrocientos sesenta mil setecientos veinticuatro sextillones quinientos diecisiete mil cuatrocientos cincuenta y cinco quintillones doscientos ochenta y cinco mil ochocientos once cuatrillones ciento dos mil setecientos ochenta y un trillones ciento cuarenta y cinco mil cuatrocientos ochenta y ocho billones ciento sesenta y ocho mil ochocientos cincuenta y ocho millones cuatrocientos treinta y cinco mil cuatrocientos sesenta y uno

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881 688 584 354 614:doscientos nueve mil trescientos doce decillones novecientos cuarenta mil seiscientos diecisiete nonillones cuatrocientos ochenta y siete mil doscientos setenta y cinco octillones setecientos veintisiete mil ciento setenta y cuatro septillones seiscientos siete mil doscientos cuarenta y cinco sextillones ciento setenta y cuatro mil quinientos cincuenta y dos quintillones ochocientos cincuenta y ocho mil ciento once cuatrillones veintisiete mil ochocientos once trillones cuatrocientos cincuenta y cuatro mil ochocientos ochenta y un billones seiscientos ochenta y ocho mil quinientos ochenta y cuatro millones trescientos cincuenta y cuatro mil seiscientos catorce

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816 885 843 546 146:dos undecillones noventa y tres mil ciento veintinueve decillones cuatrocientos seis mil ciento setenta y cuatro nonillones ochocientos setenta y dos mil setecientos cincuenta y siete octillones doscientos setenta y un mil setecientos cuarenta y seis septillones setenta y dos mil cuatrocientos cincuenta y un sextillones setecientos cuarenta y cinco mil quinientos veintiocho quintillones quinientos ochenta y un mil ciento diez cuatrillones doscientos setenta y ocho mil ciento catorce trillones quinientos cuarenta y ocho mil ochocientos dieciséis billones ochocientos ochenta y cinco mil ochocientos cuarenta y tres millones quinientos cuarenta y seis mil ciento cuarenta y seis

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488 168 858 435 461 467:veinte undecillones novecientos treinta y un mil doscientos noventa y cuatro decillones sesenta y un mil setecientos cuarenta y ocho nonillones setecientos veintisiete mil quinientos setenta y dos octillones setecientos diecisiete mil cuatrocientos sesenta septillones setecientos veinticuatro mil quinientos diecisiete sextillones cuatrocientos cincuenta y cinco mil doscientos ochenta y cinco

quintillones ochocientos once mil ciento dos cuatrillones setecientos ochenta y un mil ciento cuarenta y cinco trillones cuatrocientos ochenta y ocho mil ciento sesenta y ocho billones ochocientos cincuenta y ocho mil cuatrocientos treinta y cinco millones cuatrocientos sesenta y un mil cuatrocientos sesenta y siete

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881 688 584 354 614 674: doscientos nueve undecillones trescientos doce mil novecientos cuarenta decillones seiscientos diecisiete mil cuatrocientos ochenta y siete nonillones doscientos setenta y cinco mil setecientos veintisiete octillones ciento setenta y cuatro mil seiscientos siete septillones doscientos cuarenta y cinco mil ciento setenta y cuatro sextillones quinientos cincuenta y dos mil ochocientos cincuenta y ocho quintillones ciento once mil veintisiete cuatrillones ochocientos once mil cuatrocientos cincuenta y cuatro trillones ochocientos ochenta y un mil seiscientos ochenta y ocho billones quinientos ochenta y cuatro mil trescientos cincuenta y cuatro millones seiscientos catorce mil seiscientos setenta y cuatro

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816 885 843 546 146 748: dos mil noventa y tres undecillones ciento veintinueve mil cuatrocientos seis decillones ciento setenta y cuatro mil ochocientos setenta y dos nonillones setecientos cincuenta y siete mil doscientos setenta y un octillones setecientos cuarenta y seis mil setenta y dos septillones cuatrocientos cincuenta y un mil setecientos cuarenta y cinco sextillones quinientos veintiocho mil quinientos ochenta y un quintillones ciento diez mil doscientos setenta y ocho cuatrillones ciento catorce mil quinientos cuarenta y ocho trillones ochocientos dieciséis mil ochocientos ochenta y cinco billones ochocientos cuarenta y tres mil quinientos cuarenta y seis millones ciento cuarenta y seis mil setecientos cuarenta y ocho

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488 168 858 435 461 467 483: veinte mil novecientos treinta y un undecillones doscientos noventa y cuatro mil sesenta y un decillones setecientos cuarenta y ocho mil setecientos veintisiete nonillones quinientos setenta y dos mil setecientos diecisiete octillones cuatrocientos sesenta mil setecientos veinticuatro septillones quinientos diecisiete mil cuatrocientos cincuenta y cinco sextillones doscientos ochenta y cinco mil ochocientos once quintillones ciento dos mil setecientos ochenta y un cuatrillones ciento cuarenta y cinco mil cuatrocientos ochenta y ocho trillones ciento sesenta y ocho mil ochocientos cincuenta y ocho billones cuatrocientos treinta y cinco mil cuatrocientos sesenta y un millones cuatrocientos sesenta y siete mil cuatrocientos ochenta y tres

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881 688 584 354 614 674 837: doscientos nueve mil trescientos doce undecillones novecientos cuarenta mil seiscientos diecisiete decillones cuatrocientos ochenta y siete mil doscientos setenta y cinco nonillones setecientos veintisiete mil ciento setenta y cuatro octillones seiscientos siete mil doscientos cuarenta y cinco septillones ciento setenta y cuatro mil quinientos cincuenta y dos sextillones ochocientos cincuenta y ocho mil ciento once quintillones veintisiete mil ochocientos once cuatrillones cuatrocientos cincuenta y cuatro mil ochocientos ochenta y un trillones seiscientos ochenta y ocho mil quinientos ochenta y cuatro billones trescientos cincuenta y cuatro mil seiscientos catorce millones seiscientos setenta y cuatro mil ochocientos treinta y siete

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816 885 843 546
146 748 376:dos duodecillones noventa y tres mil ciento veintinueve undecillones
cuatrocientos seis mil ciento setenta y cuatro decillones ochocientos setenta y dos
mil setecientos cincuenta y siete nonillones doscientos setenta y un mil setecientos
cuarenta y seis octillones setenta y dos mil cuatrocientos cincuenta y un septillones
setecientos cuarenta y cinco mil quinientos veintiocho sextillones quinientos ochenta
y un mil ciento diez quintillones doscientos setenta y ocho mil ciento catorce
cuatrillones quinientos cuarenta y ocho mil ochocientos dieciséis trillones
ochocientos ochenta y cinco mil ochocientos cuarenta y tres billones quinientos
cuarenta y seis mil ciento cuarenta y seis millones setecientos cuarenta y ocho mil
trescientos setenta y seis

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488 168 858 435
461 467 483 763:veinte duodecillones novecientos treinta y un mil doscientos noventa
y cuatro undecillones sesenta y un mil setecientos cuarenta y ocho decillones
setecientos veintisiete mil quinientos setenta y dos nonillones setecientos
diecisiete mil cuatrocientos sesenta octillones setecientos veinticuatro mil
quinientos diecisiete septillones cuatrocientos cincuenta y cinco mil doscientos
ochenta y cinco sextillones ochocientos once mil ciento dos quintillones setecientos
ochenta y un mil ciento cuarenta y cinco cuatrillones cuatrocientos ochenta y ocho
mil ciento sesenta y ocho trillones ochocientos cincuenta y ocho mil cuatrocientos
treinta y cinco billones cuatrocientos sesenta y un mil cuatrocientos sesenta y siete
millones cuatrocientos ochenta y tres mil setecientos sesenta y tres

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881 688 584 354
614 674 837 636:doscientos nueve duodecillones trescientos doce mil novecientos
cuarenta undecillones seiscientos diecisiete mil cuatrocientos ochenta y siete
decillones doscientos setenta y cinco mil setecientos veintisiete nonillones ciento
setenta y cuatro mil seiscientos siete octillones doscientos cuarenta y cinco mil
ciento setenta y cuatro septillones quinientos cincuenta y dos mil ochocientos
cincuenta y ocho sextillones ciento once mil veintisiete quintillones ochocientos
once mil cuatrocientos cincuenta y cuatro cuatrillones ochocientos ochenta y un mil
seiscientos ochenta y ocho trillones quinientos ochenta y cuatro mil trescientos
cincuenta y cuatro billones seiscientos catorce mil seiscientos setenta y cuatro
millones ochocientos treinta y siete mil seiscientos treinta y seis

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816 885 843 546
146 748 376 366:dos mil noventa y tres duodecillones ciento veintinueve mil
cuatrocientos seis undecillones ciento setenta y cuatro mil ochocientos setenta y dos
decillones setecientos cincuenta y siete mil doscientos setenta y un nonillones
setecientos cuarenta y seis mil setenta y dos octillones cuatrocientos cincuenta y un
mil setecientos cuarenta y cinco septillones quinientos veintiocho mil quinientos
ochenta y un sextillones ciento diez mil doscientos setenta y ocho quintillones
ciento catorce mil quinientos cuarenta y ocho cuatrillones ochocientos dieciséis mil
ochocientos ochenta y cinco trillones ochocientos cuarenta y tres mil quinientos
cuarenta y seis billones ciento cuarenta y seis mil setecientos cuarenta y ocho
millones trescientos setenta y seis mil trescientos sesenta y seis

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488 168 858 435
461 467 483 763 661:veinte mil novecientos treinta y un duodecillones doscientos
noventa y cuatro mil sesenta y un undecillones setecientos cuarenta y ocho mil
setecientos veintisiete decillones quinientos setenta y dos mil setecientos

diecisiete nonillones cuatrocientos sesenta mil setecientos veinticuatro octillones quinientos diecisiete mil cuatrocientos cincuenta y cinco septillones doscientos ochenta y cinco mil ochocientos once sextillones ciento dos mil setecientos ochenta y un quintillones ciento cuarenta y cinco mil cuatrocientos ochenta y ocho cuatrillones ciento sesenta y ocho mil ochocientos cincuenta y ocho trillones cuatrocientos treinta y cinco mil cuatrocientos sesenta y un billones cuatrocientos sesenta y siete mil cuatrocientos ochenta y tres millones setecientos sesenta y tres mil seiscientos sesenta y uno

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881 688 584 354 614 674 837 636 611:doscientos nueve mil trescientos doce duodecillones novecientos cuarenta mil seiscientos diecisiete undecillones cuatrocientos ochenta y siete mil doscientos setenta y cinco decillones setecientos veintisiete mil ciento setenta y cuatro nonillones seiscientos siete mil doscientos cuarenta y cinco octillones ciento setenta y cuatro mil quinientos cincuenta y dos septillones ochocientos cincuenta y ocho mil ciento once sextillones veintisiete mil ochocientos once quintillones cuatrocientos cincuenta y cuatro mil ochocientos ochenta y un cuatrillones seiscientos ochenta y ocho mil quinientos ochenta y cuatro trillones trescientos cincuenta y cuatro mil seiscientos catorce billones seiscientos setenta y cuatro mil ochocientos treinta y siete millones seiscientos treinta y seis mil seiscientos once

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816 885 843 546 146 748 376 366 111:dos tredecillones noventa y tres mil ciento veintinueve duodecillones cuatrocientos seis mil ciento setenta y cuatro undecillones ochocientos setenta y dos mil setecientos cincuenta y siete decillones doscientos setenta y un mil setecientos cuarenta y seis nonillones setenta y dos mil cuatrocientos cincuenta y un octillones setecientos cuarenta y cinco mil quinientos veintiocho septillones quinientos ochenta y un mil ciento diez sextillones doscientos setenta y ocho mil ciento catorce quintillones quinientos cuarenta y ocho mil ochocientos dieciséis cuatrillones ochocientos ochenta y cinco mil ochocientos cuarenta y tres trillones quinientos cuarenta y seis mil ciento cuarenta y seis billones setecientos cuarenta y ocho mil trescientos setenta y seis millones trescientos sesenta y seis mil ciento once

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488 168 858 435 461 467 483 763 661 118:veinte tredecillones novecientos treinta y un mil doscientos noventa y cuatro duodecillones sesenta y un mil setecientos cuarenta y ocho undecillones setecientos veintisiete mil quinientos setenta y dos decillones setecientos diecisiete mil cuatrocientos sesenta nonillones setecientos veinticuatro mil quinientos diecisiete octillones cuatrocientos cincuenta y cinco mil doscientos ochenta y cinco septillones ochocientos once mil ciento dos sextillones setecientos ochenta y un mil ciento cuarenta y cinco quintillones cuatrocientos ochenta y ocho mil ciento sesenta y ocho cuatrillones ochocientos cincuenta y ocho mil cuatrocientos treinta y cinco trillones cuatrocientos sesenta y un mil cuatrocientos sesenta y siete billones cuatrocientos ochenta y tres mil setecientos sesenta y tres millones seiscientos sesenta y un mil ciento dieciocho

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881 688 584 354 614 674 837 636 611 185:doscientos nueve tredecillones trescientos doce mil novecientos cuarenta duodecillones seiscientos diecisiete mil cuatrocientos ochenta y siete undecillones doscientos setenta y cinco mil setecientos veintisiete decillones ciento setenta y cuatro mil seiscientos siete nonillones doscientos cuarenta y cinco

mil ciento setenta y cuatro octillones quinientos cincuenta y dos mil ochocientos cincuenta y ocho septillones ciento once mil veintisiete sextillones ochocientos once mil cuatrocientos cincuenta y cuatro quintillones ochocientos ochenta y un mil seiscientos ochenta y ocho cuatrillones quinientos ochenta y cuatro mil trescientos cincuenta y cuatro trillones seiscientos catorce mil seiscientos setenta y cuatro billones ochocientos treinta y siete mil seiscientos treinta y seis millones seiscientos once mil ciento ochenta y cinco

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816 885 843 546 146 748 376 366 111 855: dos mil noventa y tres tredecillones ciento veintinueve mil cuatrocientos seis duodecillones ciento setenta y cuatro mil ochocientos setenta y dos undecillones setecientos cincuenta y siete mil doscientos setenta y un decillones setecientos cuarenta y seis mil setenta y dos nonillones cuatrocientos cincuenta y un mil setecientos cuarenta y cinco octillones quinientos veintiocho mil quinientos ochenta y un septillones ciento diez mil doscientos setenta y ocho sextillones ciento catorce mil quinientos cuarenta y ocho quintillones ochocientos dieciséis mil ochocientos ochenta y cinco cuatrillones ochocientos cuarenta y tres mil quinientos cuarenta y seis trillones ciento cuarenta y seis mil setecientos cuarenta y ocho billones trescientos setenta y seis mil trescientos sesenta y seis millones ciento once mil ochocientos cincuenta y cinco

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488 168 858 435 461 467 483 763 661 118 555: veinte mil novecientos treinta y un tredecillones doscientos noventa y cuatro mil sesenta y un duodecillones setecientos cuarenta y ocho mil setecientos veintisiete undecillones quinientos setenta y dos mil setecientos diecisiete decillones cuatrocientos sesenta mil setecientos veinticuatro nonillones quinientos diecisiete mil cuatrocientos cincuenta y cinco octillones doscientos ochenta y cinco mil ochocientos once septillones ciento dos mil setecientos ochenta y un sextillones ciento cuarenta y cinco mil cuatrocientos ochenta y ocho quintillones ciento sesenta y ocho mil ochocientos cincuenta y ocho cuatrillones cuatrocientos treinta y cinco mil cuatrocientos sesenta y un trillones cuatrocientos sesenta y siete mil cuatrocientos ochenta y tres billones setecientos sesenta y tres mil seiscientos sesenta y un millones ciento dieciocho mil quinientos cincuenta y cinco

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881 688 584 354 614 674 837 636 611 185 556: doscientos nueve mil trescientos doce tredecillones novecientos cuarenta mil seiscientos diecisiete duodecillones cuatrocientos ochenta y siete mil doscientos setenta y cinco undecillones setecientos veintisiete mil ciento setenta y cuatro decillones seiscientos siete mil doscientos cuarenta y cinco nonillones ciento setenta y cuatro mil quinientos cincuenta y dos octillones ochocientos cincuenta y ocho mil ciento once septillones veintisiete mil ochocientos once sextillones cuatrocientos cincuenta y cuatro mil ochocientos ochenta y un quintillones seiscientos ochenta y ocho mil quinientos ochenta y cuatro cuatrillones trescientos cincuenta y cuatro mil seiscientos catorce trillones seiscientos setenta y cuatro mil ochocientos treinta y siete billones seiscientos treinta y seis mil seiscientos once millones ciento ochenta y cinco mil quinientos cincuenta y seis

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816 885 843 546 146 748 376 366 111 855 562: dos catordecillones noventa y tres mil ciento veintinueve tredecillones cuatrocientos seis mil ciento setenta y cuatro duodecillones ochocientos setenta y dos mil setecientos cincuenta y siete

undecillones doscientos setenta y un mil setecientos cuarenta y seis decillones setenta y dos mil cuatrocientos cincuenta y un nonillones setecientos cuarenta y cinco mil quinientos veintiocho octillones quinientos ochenta y un mil ciento diez septillones doscientos setenta y ocho mil ciento catorce sextillones quinientos cuarenta y ocho mil ochocientos dieciséis quintillones ochocientos ochenta y cinco mil ochocientos cuarenta y tres cuatrillones quinientos cuarenta y seis mil ciento cuarenta y seis trillones setecientos cuarenta y ocho mil trescientos setenta y seis billones trescientos sesenta y seis mil ciento once millones ochocientos cincuenta y cinco mil quinientos sesenta y dos

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488 168 858 435 461 467 483 763 661 118 555 627:veinte cuatordecillones novecientos treinta y un mil doscientos noventa y cuatro tredecillones sesenta y un mil setecientos cuarenta y ocho duodecillones setecientos veintisiete mil quinientos setenta y dos undecillones setecientos diecisiete mil cuatrocientos sesenta decillones setecientos veinticuatro mil quinientos diecisiete nonillones cuatrocientos cincuenta y cinco mil doscientos ochenta y cinco octillones ochocientos once mil ciento dos septillones setecientos ochenta y un mil ciento cuarenta y cinco sextillones cuatrocientos ochenta y ocho mil ciento sesenta y ocho quintillones ochocientos cincuenta y ocho mil cuatrocientos treinta y cinco cuatrillones cuatrocientos sesenta y un mil cuatrocientos sesenta y siete trillones cuatrocientos ochenta y tres mil setecientos sesenta y tres billones seiscientos sesenta y un mil ciento dieciocho millones quinientos cincuenta y cinco mil seiscientos veintisiete

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881 688 584 354 614 674 837 636 611 185 556 277:doscientos nueve cuatordecillones trescientos doce mil novecientos cuarenta tredecillones seiscientos diecisiete mil cuatrocientos ochenta y siete duodecillones doscientos setenta y cinco mil setecientos veintisiete undecillones ciento setenta y cuatro mil seiscientos siete decillones doscientos cuarenta y cinco mil ciento setenta y cuatro nonillones quinientos cincuenta y dos mil ochocientos cincuenta y ocho octillones ciento once mil veintisiete septillones ochocientos once mil cuatrocientos cincuenta y cuatro sextillones ochocientos ochenta y un mil seiscientos ochenta y ocho quintillones quinientos ochenta y cuatro mil trescientos cincuenta y cuatro cuatrillones seiscientos catorce mil seiscientos setenta y cuatro trillones ochocientos treinta y siete mil seiscientos treinta y seis billones seiscientos once mil ciento ochenta y cinco millones quinientos cincuenta y seis mil doscientos setenta y siete

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816 885 843 546 146 748 376 366 111 855 562 775:dos mil noventa y tres cuatordecillones ciento veintinueve mil cuatrocientos seis tredecillones ciento setenta y cuatro mil ochocientos setenta y dos duodecillones setecientos cincuenta y siete mil doscientos setenta y un undecillones setecientos cuarenta y seis mil setenta y dos decillones cuatrocientos cincuenta y un mil setecientos cuarenta y cinco nonillones quinientos veintiocho mil quinientos ochenta y un octillones ciento diez mil doscientos setenta y ocho septillones ciento catorce mil quinientos cuarenta y ocho sextillones ochocientos dieciséis mil ochocientos ochenta y cinco quintillones ochocientos cuarenta y tres mil quinientos cuarenta y seis cuatrillones ciento cuarenta y seis mil setecientos cuarenta y ocho trillones trescientos setenta y seis mil trescientos sesenta y seis billones ciento once mil ochocientos cincuenta y cinco millones quinientos sesenta y dos mil setecientos setenta y cinco

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488 168 858 435
461 467 483 763 661 118 555 627 751:veinte mil novecientos treinta y un
cuatordecillones doscientos noventa y cuatro mil sesenta y un tredecillones
setecientos cuarenta y ocho mil setecientos veintisiete duodecillones quinientos
setenta y dos mil setecientos diecisiete undecillones cuatrocientos sesenta mil
setecientos veinticuatro decillones quinientos diecisiete mil cuatrocientos cincuenta
y cinco nonillones doscientos ochenta y cinco mil ochocientos once octillones ciento
dos mil setecientos ochenta y un septillones ciento cuarenta y cinco mil
cuatrocientos ochenta y ocho sextillones ciento sesenta y ocho mil ochocientos
cincuenta y ocho quintillones cuatrocientos treinta y cinco mil cuatrocientos sesenta
y un cuatrillones cuatrocientos sesenta y siete mil cuatrocientos ochenta y tres
trillones setecientos sesenta y tres mil seiscientos sesenta y un billones ciento
dieciocho mil quinientos cincuenta y cinco millones seiscientos veintisiete mil
setecientos cincuenta y uno

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881 688 584 354
614 674 837 636 611 185 556 277 510:doscientos nueve mil trescientos doce
cuatordecillones novecientos cuarenta mil seiscientos diecisiete tredecillones
cuatrocientos ochenta y siete mil doscientos setenta y cinco duodecillones
setecientos veintisiete mil ciento setenta y cuatro undecillones seiscientos siete
mil doscientos cuarenta y cinco decillones ciento setenta y cuatro mil quinientos
cincuenta y dos nonillones ochocientos cincuenta y ocho mil ciento once octillones
veintisiete mil ochocientos once septillones cuatrocientos cincuenta y cuatro mil
ochocientos ochenta y un sextillones seiscientos ochenta y ocho mil quinientos
ochenta y cuatro quintillones trescientos cincuenta y cuatro mil seiscientos catorce
cuatrillones seiscientos setenta y cuatro mil ochocientos treinta y siete trillones
seiscientos treinta y seis mil seiscientos once billones ciento ochenta y cinco mil
quinientos cincuenta y seis millones doscientos setenta y siete mil quinientos diez

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816 885 843 546
146 748 376 366 111 855 562 775 103:dos quincecillones noventa y tres mil ciento
veintinueve cuatordecillones cuatrocientos seis mil ciento setenta y cuatro
tredecillones ochocientos setenta y dos mil setecientos cincuenta y siete
duodecillones doscientos setenta y un mil setecientos cuarenta y seis undecillones
setenta y dos mil cuatrocientos cincuenta y un decillones setecientos cuarenta y
cinco mil quinientos veintiocho nonillones quinientos ochenta y un mil ciento diez
octillones doscientos setenta y ocho mil ciento catorce septillones quinientos
cuarenta y ocho mil ochocientos dieciséis sextillones ochocientos ochenta y cinco mil
ochocientos cuarenta y tres quintillones quinientos cuarenta y seis mil ciento
cuarenta y seis cuatrillones setecientos cuarenta y ocho mil trescientos setenta y
seis trillones trescientos sesenta y seis mil ciento once billones ochocientos
cincuenta y cinco mil quinientos sesenta y dos millones setecientos setenta y cinco
mil ciento tres

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488 168 858 435
461 467 483 763 661 118 555 627 751 036:veinte quincecillones novecientos treinta y
un mil doscientos noventa y cuatro cuatordecillones sesenta y un mil setecientos
cuarenta y ocho tredecillones setecientos veintisiete mil quinientos setenta y dos
duodecillones setecientos diecisiete mil cuatrocientos sesenta undecillones
setecientos veinticuatro mil quinientos diecisiete decillones cuatrocientos cincuenta
y cinco mil doscientos ochenta y cinco nonillones ochocientos once mil ciento dos
octillones setecientos ochenta y un mil ciento cuarenta y cinco septillones

cuatrocientos ochenta y ocho mil ciento sesenta y ocho sextillones ochocientos cincuenta y ocho mil cuatrocientos treinta y cinco quintillones cuatrocientos sesenta y un mil cuatrocientos sesenta y siete cuatrillones cuatrocientos ochenta y tres mil setecientos sesenta y tres trillones seiscientos sesenta y un mil ciento dieciocho billones quinientos cincuenta y cinco mil seiscientos veintisiete millones setecientos cincuenta y un mil treinta y seis

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881 688 584 354 614 674 837 636 611 185 556 277 510 360: doscientos nueve quidecillones trescientos doce mil novecientos cuarenta cuatordecillones seiscientos diecisiete mil cuatrocientos ochenta y siete tredecillones doscientos setenta y cinco mil setecientos veintisiete duodecillones ciento setenta y cuatro mil seiscientos siete undecillones doscientos cuarenta y cinco mil ciento setenta y cuatro decillones quinientos cincuenta y dos mil ochocientos cincuenta y ocho nonillones ciento once mil veintisiete octillones ochocientos once mil cuatrocientos cincuenta y cuatro septillones ochocientos ochenta y un mil seiscientos ochenta y ocho sextillones quinientos ochenta y cuatro mil trescientos cincuenta y cuatro quintillones seiscientos catorce mil seiscientos setenta y cuatro cuatrillones ochocientos treinta y siete mil seiscientos treinta y seis trillones seiscientos once mil ciento ochenta y cinco billones quinientos cincuenta y seis mil doscientos setenta y siete millones quinientos diez mil trescientos sesenta

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816 885 843 546 146 748 376 366 111 855 562 775 103 603: dos mil noventa y tres quidecillones ciento veintinueve mil cuatrocientos seis cuatordecillones ciento setenta y cuatro mil ochocientos setenta y dos tredecillones setecientos cincuenta y siete mil doscientos setenta y un duodecillones setecientos cuarenta y seis mil setenta y dos undecillones cuatrocientos cincuenta y un mil setecientos cuarenta y cinco decillones quinientos veintiocho mil quinientos ochenta y un nonillones ciento diez mil doscientos setenta y ocho octillones ciento catorce mil quinientos cuarenta y ocho septillones ochocientos dieciséis mil ochocientos ochenta y cinco sextillones ochocientos cuarenta y tres mil quinientos cuarenta y seis quintillones ciento cuarenta y seis mil setecientos cuarenta y ocho cuatrillones trescientos setenta y seis mil trescientos sesenta y seis trillones ciento once mil ochocientos cincuenta y cinco billones quinientos sesenta y dos mil setecientos setenta y cinco millones ciento tres mil seiscientos tres

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488 168 858 435 461 467 483 763 661 118 555 627 751 036 036: veinte mil novecientos treinta y un quidecillones doscientos noventa y cuatro mil sesenta y un cuatordecillones setecientos cuarenta y ocho mil setecientos veintisiete tredecillones quinientos setenta y dos mil setecientos diecisiete duodecillones cuatrocientos sesenta mil setecientos veinticuatro undecillones quinientos diecisiete mil cuatrocientos cincuenta y cinco decillones doscientos ochenta y cinco mil ochocientos once nonillones ciento dos mil setecientos ochenta y un octillones ciento cuarenta y cinco mil cuatrocientos ochenta y ocho septillones ciento sesenta y ocho mil ochocientos cincuenta y ocho sextillones cuatrocientos treinta y cinco mil cuatrocientos sesenta y un quintillones cuatrocientos sesenta y siete mil cuatrocientos ochenta y tres cuatrillones setecientos sesenta y tres mil seiscientos sesenta y un trillones ciento dieciocho mil quinientos cincuenta y cinco billones seiscientos veintisiete mil setecientos cincuenta y un millones treinta y seis mil treinta y seis

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881 688 584 354
614 674 837 636 611 185 556 277 510 360 364:doscientos nueve mil trescientos doce
quincecillones novecientos cuarenta mil seiscientos diecisiete cuatordecillones
cuatrocientos ochenta y siete mil doscientos setenta y cinco tredecillones
setecientos veintisiete mil ciento setenta y cuatro duodecillones seiscientos siete
mil doscientos cuarenta y cinco undecillones ciento setenta y cuatro mil quinientos
cincuenta y dos decillones ochocientos cincuenta y ocho mil ciento once nonillones
veintisiete mil ochocientos once octillones cuatrocientos cincuenta y cuatro mil
ochocientos ochenta y un septillones seiscientos ochenta y ocho mil quinientos
ochenta y cuatro sextillones trescientos cincuenta y cuatro mil seiscientos catorce
quintillones seiscientos setenta y cuatro mil ochocientos treinta y siete
cuatrillones seiscientos treinta y seis mil seiscientos once trillones ciento ochenta
y cinco mil quinientos cincuenta y seis billones doscientos setenta y siete mil
quinientos diez millones trescientos sesenta mil trescientos sesenta y cuatro

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816 885 843 546
146 748 376 366 111 855 562 775 103 603 642:dos sexdecillones noventa y tres mil
ciento veintinueve quincecillones cuatrocientos seis mil ciento setenta y cuatro
cuatordecillones ochocientos setenta y dos mil setecientos cincuenta y siete
tredecillones doscientos setenta y un mil setecientos cuarenta y seis duodecillones
setenta y dos mil cuatrocientos cincuenta y un undecillones setecientos cuarenta y
cinco mil quinientos veintiocho decillones quinientos ochenta y un mil ciento diez
nonillones doscientos setenta y ocho mil ciento catorce octillones quinientos
cuarenta y ocho mil ochocientos dieciséis septillones ochocientos ochenta y cinco mil
ochocientos cuarenta y tres sextillones quinientos cuarenta y seis mil ciento
cuarenta y seis quintillones setecientos cuarenta y ocho mil trescientos setenta y
seis cuatrillones trescientos sesenta y seis mil ciento once trillones ochocientos
cincuenta y cinco mil quinientos sesenta y dos billones setecientos setenta y cinco
mil ciento tres millones seiscientos tres mil seiscientos cuarenta y dos

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488 168 858 435
461 467 483 763 661 118 555 627 751 036 036 423:veinte sexdecillones novecientos
treinta y un mil doscientos noventa y cuatro quincecillones sesenta y un mil
setecientos cuarenta y ocho cuatordecillones setecientos veintisiete mil quinientos
setenta y dos tredecillones setecientos diecisiete mil cuatrocientos sesenta
duodecillones setecientos veinticuatro mil quinientos diecisiete undecillones
cuatrocientos cincuenta y cinco mil doscientos ochenta y cinco decillones ochocientos
once mil ciento dos nonillones setecientos ochenta y un mil ciento cuarenta y cinco
octillones cuatrocientos ochenta y ocho mil ciento sesenta y ocho septillones
ochocientos cincuenta y ocho mil cuatrocientos treinta y cinco sextillones
cuatrocientos sesenta y un mil cuatrocientos sesenta y siete quintillones
cuatrocientos ochenta y tres mil setecientos sesenta y tres cuatrillones seiscientos
sesenta y un mil ciento dieciocho trillones quinientos cincuenta y cinco mil
seiscientos veintisiete billones setecientos cincuenta y un mil treinta y seis
millones treinta y seis mil cuatrocientos veintitrés

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881 688 584 354
614 674 837 636 611 185 556 277 510 360 364 238:doscientos nueve sexdecillones
trescientos doce mil novecientos cuarenta quincecillones seiscientos diecisiete mil
cuatrocientos ochenta y siete cuatordecillones doscientos setenta y cinco mil
setecientos veintisiete tredecillones ciento setenta y cuatro mil seiscientos siete
duodecillones doscientos cuarenta y cinco mil ciento setenta y cuatro undecillones

quinientos cincuenta y dos mil ochocientos cincuenta y ocho decillones ciento once mil veintisiete nonillones ochocientos once mil cuatrocientos cincuenta y cuatro octillones ochocientos ochenta y un mil seiscientos ochenta y ocho septillones quinientos ochenta y cuatro mil trescientos cincuenta y cuatro sextillones seiscientos catorce mil seiscientos setenta y cuatro quintillones ochocientos treinta y siete mil seiscientos treinta y seis cuatrillones seiscientos once mil ciento ochenta y cinco trillones quinientos cincuenta y seis mil doscientos setenta y siete billones quinientos diez mil trescientos sesenta millones trescientos sesenta y cuatro mil doscientos treinta y ocho

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816 885 843 546 146 748 376 366 111 855 562 775 103 603 642 385: dos mil noventa y tres sexdecillones ciento veintinueve mil cuatrocientos seis quincecillones ciento setenta y cuatro mil ochocientos setenta y dos catordecillones setecientos cincuenta y siete mil doscientos setenta y un tredecillones setecientos cuarenta y seis mil setenta y dos duodecillones cuatrocientos cincuenta y un mil setecientos cuarenta y cinco undecillones quinientos veintiocho mil quinientos ochenta y un decillones ciento diez mil doscientos setenta y ocho nonillones ciento catorce mil quinientos cuarenta y ocho octillones ochocientos dieciséis mil ochocientos ochenta y cinco septillones ochocientos cuarenta y tres mil quinientos cuarenta y seis sextillones ciento cuarenta y seis mil setecientos cuarenta y ocho quintillones trescientos setenta y seis mil trescientos sesenta y seis cuatrillones ciento once mil ochocientos cincuenta y cinco trillones quinientos sesenta y dos mil setecientos setenta y cinco billones ciento tres mil seiscientos tres millones seiscientos cuarenta y dos mil trescientos ochenta y cinco

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488 168 858 435 461 467 483 763 661 118 555 627 751 036 036 423 852: veinte mil novecientos treinta y un sexdecillones doscientos noventa y cuatro mil sesenta y un quincecillones setecientos cuarenta y ocho mil setecientos veintisiete catordecillones quinientos setenta y dos mil setecientos diecisiete tredecillones cuatrocientos sesenta mil setecientos veinticuatro duodecillones quinientos diecisiete mil cuatrocientos cincuenta y cinco undecillones doscientos ochenta y cinco mil ochocientos once decillones ciento dos mil setecientos ochenta y un nonillones ciento cuarenta y cinco mil cuatrocientos ochenta y ocho octillones ciento sesenta y ocho mil ochocientos cincuenta y ocho septillones cuatrocientos treinta y cinco mil cuatrocientos sesenta y un sextillones cuatrocientos sesenta y siete mil cuatrocientos ochenta y tres quintillones setecientos sesenta y tres mil seiscientos sesenta y un cuatrillones ciento dieciocho mil quinientos cincuenta y cinco trillones seiscientos veintisiete mil setecientos cincuenta y un billones treinta y seis mil treinta y seis millones cuatrocientos veintitrés mil ochocientos cincuenta y dos

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881 688 584 354 614 674 837 636 611 185 556 277 510 360 364 238 526: doscientos nueve mil trescientos doce sexdecillones novecientos cuarenta mil seiscientos diecisiete quincecillones cuatrocientos ochenta y siete mil doscientos setenta y cinco catordecillones setecientos veintisiete mil ciento setenta y cuatro tredecillones seiscientos siete mil doscientos cuarenta y cinco duodecillones ciento setenta y cuatro mil quinientos cincuenta y dos undecillones ochocientos cincuenta y ocho mil ciento once decillones veintisiete mil ochocientos once nonillones cuatrocientos cincuenta y cuatro mil ochocientos ochenta y un octillones seiscientos ochenta y ocho mil quinientos ochenta y cuatro septillones trescientos cincuenta y cuatro mil seiscientos catorce

sextillones seiscientos setenta y cuatro mil ochocientos treinta y siete quintillones seiscientos treinta y seis mil seiscientos once cuatrillones ciento ochenta y cinco mil quinientos cincuenta y seis trillones doscientos setenta y siete mil quinientos diez billones trescientos sesenta mil trescientos sesenta y cuatro millones doscientos treinta y ocho mil quinientos veintiséis

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816 885 843 546 146 748 376 366 111 855 562 775 103 603 642 385 262: dos septendecillones noventa y tres mil ciento veintinueve sexdecillones cuatrocientos seis mil ciento setenta y cuatro quincecillones ochocientos setenta y dos mil setecientos cincuenta y siete catordecillones doscientos setenta y un mil setecientos cuarenta y seis tredecillones setenta y dos mil cuatrocientos cincuenta y un duodecillones setecientos cuarenta y cinco mil quinientos veintiocho undecillones quinientos ochenta y un mil ciento diez decillones doscientos setenta y ocho mil ciento catorce nonillones quinientos cuarenta y ocho mil ochocientos dieciséis octillones ochocientos ochenta y cinco mil ochocientos cuarenta y tres septillones quinientos cuarenta y seis mil ciento cuarenta y seis sextillones setecientos cuarenta y ocho mil trescientos setenta y seis quintillones trescientos sesenta y seis mil ciento once cuatrillones ochocientos cincuenta y cinco mil quinientos sesenta y dos trillones setecientos setenta y cinco mil ciento tres billones seiscientos tres mil seiscientos cuarenta y dos millones trescientos ochenta y cinco mil doscientos sesenta y dos

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488 168 858 435 461 467 483 763 661 118 555 627 751 036 036 423 852 627: veinte septendecillones novecientos treinta y un mil doscientos noventa y cuatro sexdecillones sesenta y un mil setecientos cuarenta y ocho quincecillones setecientos veintisiete mil quinientos setenta y dos catordecillones setecientos diecisiete mil cuatrocientos sesenta tredecillones setecientos veinticuatro mil quinientos diecisiete duodecillones cuatrocientos cincuenta y cinco mil doscientos ochenta y cinco undecillones ochocientos once mil ciento dos decillones setecientos ochenta y un mil ciento cuarenta y cinco nonillones cuatrocientos ochenta y ocho mil ciento sesenta y ocho octillones ochocientos cincuenta y ocho mil cuatrocientos treinta y cinco septillones cuatrocientos sesenta y un mil cuatrocientos sesenta y siete sextillones cuatrocientos ochenta y tres mil setecientos sesenta y tres quintillones seiscientos sesenta y un mil ciento dieciocho cuatrillones quinientos cincuenta y cinco mil seiscientos veintisiete trillones setecientos cincuenta y un mil treinta y seis billones treinta y seis mil cuatrocientos veintitrés millones ochocientos cincuenta y dos mil seiscientos veintisiete

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881 688 584 354 614 674 837 636 611 185 556 277 510 360 364 238 526 275: doscientos nueve septendecillones trescientos doce mil novecientos cuarenta sexdecillones seiscientos diecisiete mil cuatrocientos ochenta y siete quincecillones doscientos setenta y cinco mil setecientos veintisiete catordecillones ciento setenta y cuatro mil seiscientos siete tredecillones doscientos cuarenta y cinco mil ciento setenta y cuatro duodecillones quinientos cincuenta y dos mil ochocientos cincuenta y ocho undecillones ciento once mil veintisiete decillones ochocientos once mil cuatrocientos cincuenta y cuatro nonillones ochocientos ochenta y un mil seiscientos ochenta y ocho octillones quinientos ochenta y cuatro mil trescientos cincuenta y cuatro septillones seiscientos catorce mil seiscientos setenta y cuatro sextillones ochocientos treinta y siete mil seiscientos treinta y seis quintillones seiscientos

once mil ciento ochenta y cinco cuatrillones quinientos cincuenta y seis mil doscientos setenta y siete trillones quinientos diez mil trescientos sesenta billones trescientos sesenta y cuatro mil doscientos treinta y ocho millones quinientos veintiséis mil doscientos setenta y cinco

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816 885 843 546 146 748 376 366 111 855 562 775 103 603 642 385 262 754:dos mil noventa y tres septendecillones ciento veintinueve mil cuatrocientos seis sexdecillones ciento setenta y cuatro mil ochocientos setenta y dos quidecillones setecientos cincuenta y siete mil doscientos setenta y un cuatordecillones setecientos cuarenta y seis mil setenta y dos tredecillones cuatrocientos cincuenta y un mil setecientos cuarenta y cinco duodecillones quinientos veintiocho mil quinientos ochenta y un undecillones ciento diez mil doscientos setenta y ocho decillones ciento catorce mil quinientos cuarenta y ocho nonillones ochocientos dieciséis mil ochocientos ochenta y cinco octillones ochocientos cuarenta y tres mil quinientos cuarenta y seis septillones ciento cuarenta y seis mil setecientos cuarenta y ocho sextillones trescientos setenta y seis mil trescientos sesenta y seis quintillones ciento once mil ochocientos cincuenta y cinco cuatrillones quinientos sesenta y dos mil setecientos setenta y cinco trillones ciento tres mil seiscientos tres billones seiscientos cuarenta y dos mil trescientos ochenta y cinco millones doscientos sesenta y dos mil setecientos cincuenta y cuatro

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488 168 858 435 461 467 483 763 661 118 555 627 751 036 036 423 852 627 541:veinte mil novecientos treinta y un septendecillones doscientos noventa y cuatro mil sesenta y un sexdecillones setecientos cuarenta y ocho mil setecientos veintisiete quidecillones quinientos setenta y dos mil setecientos diecisiete cuatordecillones cuatrocientos sesenta mil setecientos veinticuatro tredecillones quinientos diecisiete mil cuatrocientos cincuenta y cinco duodecillones doscientos ochenta y cinco mil ochocientos once undecillones ciento dos mil setecientos ochenta y un decillones ciento cuarenta y cinco mil cuatrocientos ochenta y ocho nonillones ciento sesenta y ocho mil ochocientos cincuenta y ocho octillones cuatrocientos treinta y cinco mil cuatrocientos sesenta y un septillones cuatrocientos sesenta y siete mil cuatrocientos ochenta y tres sextillones setecientos sesenta y tres mil seiscientos sesenta y un quintillones ciento dieciocho mil quinientos cincuenta y cinco cuatrillones seiscientos veintisiete mil setecientos cincuenta y un trillones treinta y seis mil treinta y seis billones cuatrocientos veintitrés mil ochocientos cincuenta y dos millones seiscientos veintisiete mil quinientos cuarenta y uno

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881 688 584 354 614 674 837 636 611 185 556 277 510 360 364 238 526 275 416:doscientos nueve mil trescientos doce septendecillones novecientos cuarenta mil seiscientos diecisiete sexdecillones cuatrocientos ochenta y siete mil doscientos setenta y cinco quidecillones setecientos veintisiete mil ciento setenta y cuatro cuatordecillones seiscientos siete mil doscientos cuarenta y cinco tredecillones ciento setenta y cuatro mil quinientos cincuenta y dos duodecillones ochocientos cincuenta y ocho mil ciento once undecillones veintisiete mil ochocientos once decillones cuatrocientos cincuenta y cuatro mil ochocientos ochenta y un nonillones seiscientos ochenta y ocho mil quinientos ochenta y cuatro octillones trescientos cincuenta y cuatro mil seiscientos catorce septillones seiscientos setenta y cuatro mil ochocientos treinta y siete sextillones seiscientos treinta y seis mil seiscientos once quintillones ciento ochenta y cinco mil quinientos cincuenta y seis cuatrillones doscientos

setenta y siete mil quinientos diez trillones trescientos sesenta mil trescientos sesenta y cuatro billones doscientos treinta y ocho mil quinientos veintiséis millones doscientos setenta y cinco mil cuatrocientos dieciséis

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816 885 843 546 146 748 376 366 111 855 562 775 103 603 642 385 262 754 168: dos octodecillones noventa y tres mil ciento veintinueve septendecillones cuatrocientos seis mil ciento setenta y cuatro sexdecillones ochocientos setenta y dos mil setecientos cincuenta y siete quincecillones doscientos setenta y un mil setecientos cuarenta y seis catordecillones setenta y dos mil cuatrocientos cincuenta y un tredecillones setecientos cuarenta y cinco mil quinientos veintiocho duodecillones quinientos ochenta y un mil ciento diez undecillones doscientos setenta y ocho mil ciento catorce decillones quinientos cuarenta y ocho mil ochocientos dieciséis nonillones ochocientos ochenta y cinco mil ochocientos cuarenta y tres octillones quinientos cuarenta y seis mil ciento cuarenta y seis septillones setecientos cuarenta y ocho mil trescientos setenta y seis sextillones trescientos sesenta y seis mil ciento once quintillones ochocientos cincuenta y cinco mil quinientos sesenta y dos cuatrillones setecientos setenta y cinco mil ciento tres trillones seiscientos tres mil seiscientos cuarenta y dos billones trescientos ochenta y cinco mil doscientos sesenta y dos millones setecientos cincuenta y cuatro mil ciento sesenta y ocho

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488 168 858 435 461 467 483 763 661 118 555 627 751 036 036 423 852 627 541 680: veinte octodecillones novecientos treinta y un mil doscientos noventa y cuatro septendecillones sesenta y un mil setecientos cuarenta y ocho sexdecillones setecientos veintisiete mil quinientos setenta y dos quincecillones setecientos diecisiete mil cuatrocientos sesenta catordecillones setecientos veinticuatro mil quinientos diecisiete tredecillones cuatrocientos cincuenta y cinco mil doscientos ochenta y cinco duodecillones ochocientos once mil ciento dos undecillones setecientos ochenta y un mil ciento cuarenta y cinco decillones cuatrocientos ochenta y ocho mil ciento sesenta y ocho nonillones ochocientos cincuenta y ocho mil cuatrocientos treinta y cinco octillones cuatrocientos sesenta y un mil cuatrocientos sesenta y siete septillones cuatrocientos ochenta y tres mil setecientos sesenta y tres sextillones seiscientos sesenta y un mil ciento dieciocho quintillones quinientos cincuenta y cinco mil seiscientos veintisiete cuatrillones setecientos cincuenta y un mil treinta y seis trillones treinta y seis mil cuatrocientos veintitrés billones ochocientos cincuenta y dos mil seiscientos veintisiete millones quinientos cuarenta y un mil seiscientos ochenta

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881 688 584 354 614 674 837 636 611 185 556 277 510 360 364 238 526 275 416 806: doscientos nueve octodecillones trescientos doce mil novecientos cuarenta septendecillones seiscientos diecisiete mil cuatrocientos ochenta y siete sexdecillones doscientos setenta y cinco mil setecientos veintisiete quincecillones ciento setenta y cuatro mil seiscientos siete catordecillones doscientos cuarenta y cinco mil ciento setenta y cuatro tredecillones quinientos cincuenta y dos mil ochocientos cincuenta y ocho duodecillones ciento once mil veintisiete undecillones ochocientos once mil cuatrocientos cincuenta y cuatro decillones ochocientos ochenta y un mil seiscientos ochenta y ocho nonillones quinientos ochenta y cuatro mil trescientos cincuenta y cuatro octillones seiscientos catorce mil seiscientos setenta y cuatro septillones ochocientos treinta y siete mil seiscientos treinta y seis sextillones seiscientos once mil ciento ochenta y cinco quintillones quinientos cincuenta y seis mil

doscientos setenta y siete cuatrillones quinientos diez mil trescientos sesenta trillones trescientos sesenta y cuatro mil doscientos treinta y ocho billones quinientos veintiséis mil doscientos setenta y cinco millones cuatrocientos dieciséis mil ochocientos seis

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816 885 843 546 146 748 376 366 111 855 562 775 103 603 642 385 262 754 168 066:dos mil noventa y tres octodecillones ciento veintinueve mil cuatrocientos seis septendecillones ciento setenta y cuatro mil ochocientos setenta y dos sexdecillones setecientos cincuenta y siete mil doscientos setenta y un quidecillones setecientos cuarenta y seis mil setenta y dos catordecillones cuatrocientos cincuenta y un mil setecientos cuarenta y cinco tredecillones quinientos veintiocho mil quinientos ochenta y un duodecillones ciento diez mil doscientos setenta y ocho undecillones ciento catorce mil quinientos cuarenta y ocho decillones ochocientos dieciséis mil ochocientos ochenta y cinco nonillones ochocientos cuarenta y tres mil quinientos cuarenta y seis octillones ciento cuarenta y seis mil setecientos cuarenta y ocho septillones trescientos setenta y seis mil trescientos sesenta y seis sextillones ciento once mil ochocientos cincuenta y cinco quintillones quinientos sesenta y dos mil setecientos setenta y cinco cuatrillones ciento tres mil seiscientos tres trillones seiscientos cuarenta y dos mil trescientos ochenta y cinco billones doscientos sesenta y dos mil setecientos cincuenta y cuatro millones ciento sesenta y ocho mil sesenta y seis

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488 168 858 435 461 467 483 763 661 118 555 627 751 036 036 423 852 627 541 680 660:veinte mil novecientos treinta y un octodecillones doscientos noventa y cuatro mil sesenta y un septendecillones setecientos cuarenta y ocho mil setecientos veintisiete sexdecillones quinientos setenta y dos mil setecientos diecisiete quidecillones cuatrocientos sesenta mil setecientos veinticuatro catordecillones quinientos diecisiete mil cuatrocientos cincuenta y cinco tredecillones doscientos ochenta y cinco mil ochocientos once duodecillones ciento dos mil setecientos ochenta y un undecillones ciento cuarenta y cinco mil cuatrocientos ochenta y ocho decillones ciento sesenta y ocho mil ochocientos cincuenta y ocho nonillones cuatrocientos treinta y cinco mil cuatrocientos sesenta y un octillones cuatrocientos sesenta y siete mil cuatrocientos ochenta y tres septillones setecientos sesenta y tres mil seiscientos sesenta y un sextillones ciento dieciocho mil quinientos cincuenta y cinco quintillones seiscientos veintisiete mil setecientos cincuenta y un cuatrillones treinta y seis mil treinta y seis trillones cuatrocientos veintitrés mil ochocientos cincuenta y dos billones seiscientos veintisiete mil quinientos cuarenta y un millones seiscientos ochenta mil seiscientos sesenta

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881 688 584 354 614 674 837 636 611 185 556 277 510 360 364 238 526 275 416 806 607:doscientos nueve mil trescientos doce octodecillones novecientos cuarenta mil seiscientos diecisiete septendecillones cuatrocientos ochenta y siete mil doscientos setenta y cinco sexdecillones setecientos veintisiete mil ciento setenta y cuatro quidecillones seiscientos siete mil doscientos cuarenta y cinco catordecillones ciento setenta y cuatro mil quinientos cincuenta y dos tredecillones ochocientos cincuenta y ocho mil ciento once duodecillones veintisiete mil ochocientos once undecillones cuatrocientos cincuenta y cuatro mil ochocientos ochenta y un decillones seiscientos ochenta y ocho mil quinientos ochenta y cuatro nonillones trescientos cincuenta y cuatro mil seiscientos catorce octillones seiscientos setenta y cuatro mil ochocientos treinta y siete septillones seiscientos treinta y seis mil seiscientos once sextillones ciento

ochenta y cinco mil quinientos cincuenta y seis quintillones doscientos setenta y siete mil quinientos diez cuatrillones trescientos sesenta mil trescientos sesenta y cuatro trillones doscientos treinta y ocho mil quinientos veintiséis billones doscientos setenta y cinco mil cuatrocientos dieciséis millones ochocientos seis mil seiscientos siete

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816 885 843 546 146 748 376 366 111 855 562 775 103 603 642 385 262 754 168 066 072:dos
novendecillones noventa y tres mil ciento veintinueve octodecillones cuatrocientos seis mil ciento setenta y cuatro septendecillones ochocientos setenta y dos mil setecientos cincuenta y siete sexdecillones doscientos setenta y un mil setecientos cuarenta y seis quincecillones setenta y dos mil cuatrocientos cincuenta y un cuatordecillones setecientos cuarenta y cinco mil quinientos veintiocho tredecillones quinientos ochenta y un mil ciento diez duodecillones doscientos setenta y ocho mil ciento catorce undecillones quinientos cuarenta y ocho mil ochocientos dieciséis decillones ochocientos ochenta y cinco mil ochocientos cuarenta y tres nonillones quinientos cuarenta y seis mil ciento cuarenta y seis octillones setecientos cuarenta y ocho mil trescientos setenta y seis septillones trescientos sesenta y seis mil ciento once sextillones ochocientos cincuenta y cinco mil quinientos sesenta y dos quintillones setecientos setenta y cinco mil ciento tres cuatrillones seiscientos tres mil seiscientos cuarenta y dos trillones trescientos ochenta y cinco mil doscientos sesenta y dos billones setecientos cincuenta y cuatro mil ciento sesenta y ocho millones sesenta y seis mil setenta y dos

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488 168 858 435 461 467 483 763 661 118 555 627 751 036 036 423 852 627 541 680 660 728:veinte
novendecillones novecientos treinta y un mil doscientos noventa y cuatro octodecillones sesenta y un mil setecientos cuarenta y ocho septendecillones setecientos veintisiete mil quinientos setenta y dos sexdecillones setecientos diecisiete mil cuatrocientos sesenta quincecillones setecientos veinticuatro mil quinientos diecisiete cuatordecillones cuatrocientos cincuenta y cinco mil doscientos ochenta y cinco tredecillones ochocientos once mil ciento dos duodecillones setecientos ochenta y un mil ciento cuarenta y cinco undecillones cuatrocientos ochenta y ocho mil ciento sesenta y ocho decillones ochocientos cincuenta y ocho mil cuatrocientos treinta y cinco nonillones cuatrocientos sesenta y un mil cuatrocientos sesenta y siete octillones cuatrocientos ochenta y tres mil setecientos sesenta y tres septillones seiscientos sesenta y un mil ciento dieciocho sextillones quinientos cincuenta y cinco mil seiscientos veintisiete quintillones setecientos cincuenta y un mil treinta y seis cuatrillones treinta y seis mil cuatrocientos veintitrés trillones ochocientos cincuenta y dos mil seiscientos veintisiete billones quinientos cuarenta y un mil seiscientos ochenta millones seiscientos sesenta mil setecientos veintiocho

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881 688 584 354 614 674 837 636 611 185 556 277 510 360 364 238 526 275 416 806 607 282:doscientos
nueve novendecillones trescientos doce mil novecientos cuarenta octodecillones seiscientos diecisiete mil cuatrocientos ochenta y siete septendecillones doscientos setenta y cinco mil setecientos veintisiete sexdecillones ciento setenta y cuatro mil seiscientos siete quincecillones doscientos cuarenta y cinco mil ciento setenta y cuatro cuatordecillones quinientos cincuenta y dos mil ochocientos cincuenta y ocho tredecillones ciento once mil veintisiete duodecillones ochocientos once mil cuatrocientos cincuenta y cuatro undecillones ochocientos ochenta y un mil seiscientos ochenta y ocho decillones quinientos ochenta y cuatro mil trescientos

cincuenta y cuatro nonillones seiscientos catorce mil seiscientos setenta y cuatro octillones ochocientos treinta y siete mil seiscientos treinta y seis septillones seiscientos once mil ciento ochenta y cinco sextillones quinientos cincuenta y seis mil doscientos setenta y siete quintillones quinientos diez mil trescientos sesenta cuatrillones trescientos sesenta y cuatro mil doscientos treinta y ocho trillones quinientos veintiséis mil doscientos setenta y cinco billones cuatrocientos dieciséis mil ochocientos seis millones seiscientos siete mil doscientos ochenta y dos

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816 885 843 546 146 748 376 366 111 855 562 775 103 603 642 385 262 754 168 066 072 824: dos mil noventa y tres novendecillones ciento veintinueve mil cuatrocientos seis octodecillones ciento setenta y cuatro mil ochocientos setenta y dos septendecillones setecientos cincuenta y siete mil doscientos setenta y un sexdecillones setecientos cuarenta y seis mil setenta y dos quincecillones cuatrocientos cincuenta y un mil setecientos cuarenta y cinco cuatordecillones quinientos veintiocho mil quinientos ochenta y un tredecillones ciento diez mil doscientos setenta y ocho duodecillones ciento catorce mil quinientos cuarenta y ocho undecillones ochocientos dieciséis mil ochocientos ochenta y cinco decillones ochocientos cuarenta y tres mil quinientos cuarenta y seis nonillones ciento cuarenta y seis mil setecientos cuarenta y ocho octillones trescientos setenta y seis mil trescientos sesenta y seis septillones ciento once mil ochocientos cincuenta y cinco sextillones quinientos sesenta y dos mil setecientos setenta y cinco quintillones ciento tres mil seiscientos tres cuatrillones seiscientos cuarenta y dos mil trescientos ochenta y cinco trillones doscientos sesenta y dos mil setecientos cincuenta y cuatro billones ciento sesenta y ocho mil sesenta y seis millones setenta y dos mil ochocientos veinticuatro

20 931 294 061 748 727 572 717 460 724 517 455 285 811 102 781 145 488 168 858 435 461 467 483 763 661 118 555 627 751 036 036 423 852 627 541 680 660 728 242: veinte mil novecientos treinta y un novendecillones doscientos noventa y cuatro mil sesenta y un octodecillones setecientos cuarenta y ocho mil setecientos veintisiete septendecillones quinientos setenta y dos mil setecientos diecisiete sexdecillones cuatrocientos sesenta mil setecientos veinticuatro quincecillones quinientos diecisiete mil cuatrocientos cincuenta y cinco cuatordecillones doscientos ochenta y cinco mil ochocientos once tredecillones ciento dos mil setecientos ochenta y un duodecillones ciento cuarenta y cinco mil cuatrocientos ochenta y ocho undecillones ciento sesenta y ocho mil ochocientos cincuenta y ocho decillones cuatrocientos treinta y cinco mil cuatrocientos sesenta y un nonillones cuatrocientos sesenta y siete mil cuatrocientos ochenta y tres octillones setecientos sesenta y tres mil seiscientos sesenta y un septillones ciento dieciocho mil quinientos cincuenta y cinco sextillones seiscientos veintisiete mil setecientos cincuenta y un quintillones treinta y seis mil treinta y seis cuatrillones cuatrocientos veintitrés mil ochocientos cincuenta y dos trillones seiscientos veintisiete mil quinientos cuarenta y un billones seiscientos ochenta mil seiscientos sesenta millones setecientos veintiocho mil doscientos cuarenta y dos

209 312 940 617 487 275 727 174 607 245 174 552 858 111 027 811 454 881 688 584 354 614 674 837 636 611 185 556 277 510 360 364 238 526 275 416 806 607 282 423: doscientos nueve mil trescientos doce novendecillones novecientos cuarenta mil seiscientos diecisiete octodecillones cuatrocientos ochenta y siete mil doscientos setenta y cinco septendecillones setecientos veintisiete mil ciento setenta y cuatro sexdecillones seiscientos siete mil doscientos cuarenta y cinco quincecillones ciento setenta y cuatro mil quinientos cincuenta y dos cuatordecillones ochocientos

cincuenta y ocho mil ciento once tredecillones veintisiete mil ochocientos once duodecillones cuatrocientos cincuenta y cuatro mil ochocientos ochenta y un undecillones seiscientos ochenta y ocho mil quinientos ochenta y cuatro decillones trescientos cincuenta y cuatro mil seiscientos catorce nonillones seiscientos setenta y cuatro mil ochocientos treinta y siete octillones seiscientos treinta y seis mil seiscientos once septillones ciento ochenta y cinco mil quinientos cincuenta y seis sextillones doscientos setenta y siete mil quinientos diez quintillones trescientos sesenta mil trescientos sesenta y cuatro cuatrillones doscientos treinta y ocho mil quinientos veintiséis trillones doscientos setenta y cinco mil cuatrocientos dieciséis billones ochocientos seis mil seiscientos siete millones doscientos ochenta y dos mil cuatrocientos veintitrés

2 093 129 406 174 872 757 271 746 072 451 745 528 581 110 278 114 548 816 885 843 546 146 748 376 366 111 855 562 775 103 603 642 385 262 754 168 066 072 824 231: dos vigintillones noventa y tres mil ciento veintinueve novendecillones cuatrocientos seis mil ciento setenta y cuatro octodecillones ochocientos setenta y dos mil setecientos cincuenta y siete septendecillones doscientos setenta y un mil setecientos cuarenta y seis sexdecillones setenta y dos mil cuatrocientos cincuenta y un quincecillones setecientos cuarenta y cinco mil quinientos veintiocho cuatordecillones quinientos ochenta y un mil ciento diez tredecillones doscientos setenta y ocho mil ciento catorce duodecillones quinientos cuarenta y ocho mil ochocientos dieciséis undecillones ochocientos ochenta y cinco mil ochocientos cuarenta y tres decillones quinientos cuarenta y seis mil ciento cuarenta y seis nonillones setecientos cuarenta y ocho mil trescientos setenta y seis octillones trescientos sesenta y seis mil ciento once septillones ochocientos cincuenta y cinco mil quinientos sesenta y dos sextillones setecientos setenta y cinco mil ciento tres quintillones seiscientos tres mil seiscientos cuarenta y dos cuatrillones trescientos ochenta y cinco mil doscientos sesenta y dos trillones setecientos cincuenta y cuatro mil ciento sesenta y ocho billones sesenta y seis mil setenta y dos millones ochocientos veinticuatro mil doscientos treinta y uno

1087/10 298 371: mil ochenta y siete diezmil millones doscientos noventa y ocho mil trescientos setenta y unavos

491 029 102 098/1 029 381 023 918 203: cuatrocientos noventa y un mil veintinueve millones ciento dos mil noventa y ocho milveintinueve billones trescientos ochenta y un milveintitres millones novecientos dieciocho mil doscientos tresavos

1 290 381 293/2: mil doscientos noventa millones trescientos ochenta y un mil doscientos noventa y tres medios

1/102 948 143 181 290 481 281: un cientodostrillones novecientos cuarenta y ocho milcientocuatrentaitres billonescientochenta y un mil doscientos noventa millones cuatrocientos ochenta y un mil doscientos ochenta y unavo

10 238 131 381 290 318 120 712/120 938 190 238 214 204 824 498: diez mil doscientos treinta y ocho trillones ciento treinta y un mil trescientos ochenta y un billones doscientos noventa mil trescientos dieciocho millones ciento veinte mil setecientos doce cientoveintemil novecientos treinta y ocho trillonescientonoventamildoscientostreinta y ocho billones doscientos catorcemildoscientoscuatromillones ochocientos veinticuatro mil cuatrocientos noventa y ochoavos

1,123 98:uno con doce mil trescientas noventa y ocho cienmilésimas

12 938 193,120 931 312 03:doce millones novecientos treinta y ocho mil ciento noventa y tres coma doce mil noventa y tres millones ciento treinta y una mil doscientas tres cienmilmillonésimas

1,000 000 000 001 293 483:uno y un millón doscientas noventa y tres mil cuatrocientas ochenta y tres trillonésimas

19 283,001 233 441 241 3:diecinueve mil doscientos ochenta y tres con doce mil trescientos treinta y cuatro millones cuatrocientas doce mil cuatrocientas trece diezbillonésimas

20 945 729 084 572 234 908 573 954,000 000 100 203 457 280 452 435 773:veinte cuatrillones novecientos cuarenta y cinco mil setecientos veintinueve trillones ochenta y cuatro mil quinientos setenta y dos billones doscientos treinta y cuatro mil novecientos ocho millones quinientos setenta y tres mil novecientos cincuenta y cuatro con cien trillones doscientos tres mil cuatrocientos cincuenta y siete billones doscientos ochenta mil cuatrocientos cincuenta y dos millones cuatrocientas treinta y cinco mil setecientas setenta y tres milcuatrillonésimas