



ULPGC
Universidad de
Las Palmas de
Gran Canaria

eii

ESCUELA DE
INGENIERÍA INFORMÁTICA

Trabajo de Fin de Grado

Conversor de cifras a texto en árabe

TITULACIÓN: Grado en Ingeniería Informática

AUTOR: HOUDA AIT ABDESLAM

TUTORIZADO POR:

Francisco Javier Carreras Riudavets

Fecha [07/2025]

SOLICITUD DE DEFENSA DE TRABAJO DE FIN DE TÍTULO

D./D^a Houda Ait Abdeslam, autor del trabajo 'Convertor de cifras a texto en árabe', correspondiente a la titulación Ingeniería informática.

SOLICITA

que se inicie el procedimiento de defensa del mismo, para lo que se adjunta la documentación requerida, haciendo constar que

se autoriza / no se autoriza la grabación en audio de la exposición y turno de preguntas.

Asimismo, con respecto al registro de la propiedad intelectual/industrial del TFT, declara que:

Se ha iniciado o hay intención de iniciarlo (defensa no pública).

No está previsto.

Y para que así conste firma la presente. (fecha en firma electrónica)

El/La estudiante

Fdo. _____

A rellenar y firmar **obligatoriamente** por el/la/los/las tutores

En relación con la presente solicitud, se informa: (firmar donde corresponda)

Positivamente (en caso de detección de copia, esta firma quedará invalidada)

Fdo. _____

Negativamente (justificación en TFT05)

Fdo. _____

DIRECTOR DE LA ESCUELA DE INGENIERÍA INFORMÁTICA

Agradecimientos

Agradezco de todo mi corazón a todas las personas que me han acompañado y apoyado durante estos cuatro años de la carrera.

Resumen

El objetivo principal de este Trabajo de Fin de Grado (TFG) es el desarrollo de un servicio web (WCF) capaz de convertir cifras numéricas a texto en árabe. El servicio ofrecerá varios formatos de conversión, como el cardinal en su versión femenina y masculina, el ordinal, el romano y las cifras arábicas orientales. También se proporcionarán ejemplos y notas sobre cada tipo de conversión. Además, se desarrollará una página web usando ASP.net que utilizará este servicio para proporcionar una interfaz gráfica fácil de usar, permitiendo a los usuarios introducir cifras numéricas y obtener todas las conversiones disponibles.

Abstract

The main objective of this project is to develop a web service (WCF) capable of converting numbers into Arabic text. The service will offer several conversion formats, such as cardinal form (in both its masculine and feminine versions), ordinal form, roman numbers, and Eastern Arabic numerals, along with examples and interesting notes about each conversion. Additionally, a web page will be developed using ASP.net that will use this service to provide an easy-to-use graphical interface, allowing users to input numerical figures and obtain all available conversions.

Índice general

1. Estado actual y objetivos iniciales	9
1.1. Aplicaciones similares	9
1.1.1. Primer conversor: AccountingW	9
1.1.2. Segundo conversor: Numbers.un-web.com	9
1.1.3. Otras herramientas de inteligencia artificial	10
1.1.4. El proyecto que se va a desarrollar:	10
1.2. Objetivos iniciales	10
1.3. Análisis general de los números en árabe:	11
1.3.1. Cifras cardinales	11
1.3.2. Números ordinales:	17
1.3.3. Números orientales:	18
1.3.4. Números multiplicativos:	19
1.3.5. Números negativos:	20
1.4. La importancia de una herramienta para convertir números a texto	21
1.4.1. Casos en los que se escriben números en texto en árabe	21
1.5. Motivación:	21
2. Justificación de las competencias específicas cubiertas	23
3. Aportaciones	24
3.1. Alineamiento con los objetivos de desarrollo sostenible	26
4. Diseño	28
4.1. Tecnologías:	28
4.1.1. Programación orientada a objetos (POO)	28
4.1.2. Visual Studio	29
4.1.3. Servicio Web - WCF (Windows Communication Foundation)	29
4.1.4. ASP.NET	30
4.1.5. C#	30
4.1.6. Paralelismo	31
4.2. Backend:	31
4.2.1. Estructura del servicio web	31
4.2.2. Entrada	31
4.2.3. Salida	32

4.2.4.	Descripción de las clases de conversión del backend	33
4.2.5.	Integración con el frontend	44
4.3.	Frontend:	44
4.3.1.	Vistas de las conversiones	47
5.	Desarrollo	50
5.1.	Metodología	50
5.1.1.	Fases de desarrollo	50
5.1.2.	Control de versiones	51
6.	Conclusiones y trabajos futuros	53
6.1.	Conclusiones	53
6.2.	Trabajos futuros	53

Índice de figuras

3.1. Vista de usuarios activos según Google Analytics.	26
4.1. Vista de la interfaz gráfica de conversión	45
4.2. Vista: numero formateado	45
4.3. Vista de acordeón de resultados	46
4.4. Vista: conversión cardinal	47
4.5. Vista: conversión cardinal con opciones y notas	48
4.6. Vista: conversión ordinal	48
4.7. Vista: conversión a romano	49
4.8. Vista: conversión multiplicativa	49

Índice de tablas

1.1. Números en árabe del 0 al 10 (masculino y femenino)	12
1.2. Números del 11 al 19 en árabe	13
1.3. Decenas del 20 al 90 en árabe	14
1.4. Números 100 y 200 en árabe	14
1.5. Centenas en árabe	15
1.6. Miles en árabe	15
1.7. Números de mil a nueve mil en árabe	16
1.8. Grandes números en árabe	16
1.9. Números orientales en árabe	19
1.10. Números multiplicativos en árabe	20
1.11. Números negativos en árabe	20
3.1. Grado de relación del TFT con los objetivos de desarrollo sostenible.	27
4.1. Método convertir de la clase <code>ConversorNumerosArabes</code>	35
4.2. Método <code>ConvertirCardinal</code> de la clase <code>ConversorNumerosArabes</code>	36
4.3. Método <code>ConvertirGrupo</code> de la clase <code>ConversorNumerosArabes</code>	36
4.4. Método <code>EscalaDeNumeros</code> de la clase <code>ConversorNumerosArabes</code>	37
4.5. Tabla del Método <code>ConvertirOrdinal</code> de la clase <code>ConversorOrdinal</code>	39
4.6. Método <code>ConvertirGrupoOrdinal</code> de la clase <code>ConversorOrdinal</code>	39
4.7. Método <code>ConvertirANumeroOriental</code> de la clase <code>ConversorOriental</code>	41
4.8. Método <code>ConvertirARomano</code> de la clase <code>ConversorRomano</code>	43
4.9. Método <code>ConvertirMultiplicativo</code> de la clase <code>ConversorMultiplicativo</code>	44

1. Estado actual y objetivos iniciales

Este proyecto tiene como objetivo crear un servicio web que convierte cifras a texto en árabe, devolviendo diferentes formatos de los números. En el siguiente apartado voy a comentar el estado actual de conversores similares que ya existen en la web, y cómo se comparan con el que se busca desarrollar aquí.

1.1. Aplicaciones similares

Para evaluar el estado actual de conversores similares, presento a continuación dos conversores que encontré por internet, explicando tanto sus limitaciones como las características únicas que hacen destacar al proyecto que se va a desarrollar.

1.1.1. Primer conversor: AccountingW

El primer ejemplo que quiero presentar es la página citada en [1]. Esta web convierte números a texto en árabe, pero tiene varias limitaciones:

- ✓ **Formato masculino únicamente:** La conversión solo se realiza en formato masculino.
- ✓ **Solo ofrece la conversión en el formato cardinal**
- ✓ **Límite de dígitos:** Tiene un límite de hasta 18 dígitos, lo que restringe su uso con números grandes.

1.1.2. Segundo conversor: Numbers.un-web.com

El segundo ejemplo es la página de numbers.un-web.com [2]. Este conversor presenta un funcionamiento similar al primero, pero con las siguientes limitaciones:

- ✓ **Solo ofrece la conversión en el formato cardinal:** Al igual que el primero, solo convierte cifras a texto en un formato único (cardinal).

- ✓ **Límite de dígitos más bajo:** Este conversor tiene un límite de 12 dígitos, lo que restringe su uso, especialmente para convertir números más grandes o complejos.
- ✓ **No diferencia de género:** Tampoco ofrece la opción para cambiar el género a femenino.

1.1.3. Otras herramientas de inteligencia artificial

También he probado algunas herramientas de inteligencia artificial como ChatGPT [3], DeepSeek [4] o Microsoft Copilot [5], que también pueden ser usadas para convertir números a texto. Sin embargo, tienen la siguiente limitación:

- ✓ Estas herramientas no están diseñadas específicamente para la tarea de conversión de números a texto, lo que puede llevar a resultados inconsistentes o incorrectos, especialmente cuando se trata de números muy grandes o estructuras más complejas.

1.1.4. El proyecto que se va a desarrollar:

El proyecto que se está desarrollando ofrece una solución mucho más completa y flexible que las herramientas mencionadas, con las siguientes características:

- ✓ **Soporte para múltiples formatos:** El proyecto soporta varios formatos de conversión, incluyendo cardinal, ordinal, números romanos, números arábigos orientales y multiplicativo, lo que permite una conversión mucho más completa.
- ✓ **Soporte para género:** Se devuelve la conversión del número en formato cardinal, en masculino y femenino, adaptándose a las necesidades gramaticales del árabe.
- ✓ **Gran capacidad para números:** El conversor puede convertir números de hasta 3582 dígitos, lo que lo hace adecuado para aplicaciones que requieren la conversión de números extremadamente grandes.
- ✓ **Ejemplos y notas:** También incluye ejemplos y notas explicativas para cada tipo de conversión.

En resumen, los conversores existentes son útiles para tareas básicas, pero el proyecto que se está desarrollando ofrece una solución mucho más completa que se adapta a una variedad de necesidades y contextos.

1.2. Objetivos iniciales

La idea principal de este proyecto es crear una herramienta que convierte cifras numéricas a texto en árabe. No se trata solo de devolver una conversión básica en el formato cardinal, sino de ofrecer distintas formas de conversión como cardinal, ordinal, romano o incluso cifras arábigas orientales, además de considerar detalles como el género (masculino o femenino)

cuando sea necesario. Desde el principio se marcaron varios objetivos claros para poder llevarlo a cabo:

- ✓ Desarrollar un servicio web Windows Communication Foundation (WCF) que se encargue de hacer la conversión de forma automática, siguiendo las reglas del idioma árabe y asegurando que los resultados sean correctos.
- ✓ Crear una aplicación web con ASP.NET, donde el usuario pueda introducir un número y ver todos los resultados disponibles de la conversión. La idea es que sea fácil de usar y que muestre también ejemplos o notas que ayuden a entender mejor la conversión.
- ✓ Incluir varios formatos de conversión, como:
 - Cardinal (tanto en masculino como en femenino).
 - Ordinal (primero, segundo, etc.).
 - Romano (I, II, III...).
 - Multiplicativo.
 - Cifras arábigas orientales (usadas comúnmente en muchos países árabes).

1.3. Análisis general de los números en árabe:

Los números cardinales son aquellos que indican una cantidad de elementos en un conjunto. Son fundamentales en cualquier idioma, ya que se usan para contar y expresar la cantidad exacta de algo. En árabe, la construcción de los números cardinales tiene ciertas particularidades que se deben tener en cuenta para asegurar una conversión correcta.

1.3.1. Cifras cardinales

En árabe, los números cardinales se expresan utilizando los números naturales, pero con una particularidad importante: Los números tienen diferentes formas para el masculino y el femenino, y la forma correcta suele depender del género del sustantivo que acompaña el número.

1.3.1.1. Números del 0 al 10:

A continuación, se muestra cómo se estructuran los números del 1 al 10 en árabe en la tabla 1.1.

Número	masculino	femenino
0	صفر	صفر
1	واحد	وَاحِدَةٌ
2	اثنان	اثنان
3	ثَلَاثٌ	ثَلَاثَةٌ
4	أَرْبَعٌ	أَرْبَعَةٌ
5	خَمْسٌ	خَمْسَةٌ
6	سِتٌّ	سِتَّةٌ
7	سَبْعٌ	سَبْعَةٌ
8	ثَمَانِي	ثَمَانِيَةٌ
9	تِسْعٌ	تِسْعَةٌ
10	عَشْرٌ	عَشْرَةٌ

Tabla 1.1. Números en árabe del 0 al 10 (masculino y femenino)

Ejemplo:

- ✓ كِتَابٌ وَاحِدٌ ("un libro", masculino).
- ✓ طَاوِلَةٌ وَاحِدَةٌ ("una mesa", femenino).

Números del 3 al 10: Usan género inverso: si el sustantivo es masculino, el número va en femenino, y viceversa.

Ejemplo:

- ✓ ثَلَاثَةُ كُتُبٍ ("tres libros", el sustantivo es masculino, pero el número ثَلَاثَةُ es femenino).
- ✓ ثَلَاثُ طَاوِلَاتٍ ("tres mesas", el sustantivo es femenino, pero el número ثَلَاثُ es masculino).

1.3.1.2. Números del 11 al 19:

Los números del 11 al 19 se forman combinando los números del 1 al 9 con la palabra عَشْرٌ que significa "diez". Esta combinación se realiza de la siguiente manera mostrada en la tabla 1.2:

Número	En árabe
11	أحد عشر
12	اثنا عشر
13	ثلاثة عشر
14	أربعة عشر
15	خمسة عشر
16	سنة عشر
17	سبعة عشر
18	ثمانية عشر
19	تسعة عشر

Tabla 1.2. Números del 11 al 19 en árabe

Aunque en el sistema numérico árabe, los números entre el 11 y el 19 son combinaciones fijas, el género de los números sigue siendo importante cuando se usan en combinación con sustantivos. Los números del 13 al 19, si el sustantivo al que acompaña el número es femenino, tendrán la parte de la unidad en forma masculina y la parte de la decena en forma femenina, y lo contrario ocurre con los números 11 y 12.

Ejemplos:

- ✓ **13:** ثلاثة عشر - *masculino* (cuando se refiere a un sustantivo masculino).
- ✓ **13:** ثلاث عشرة - *femenino* (cuando se refiere a un sustantivo femenino).
- ✓ **12:** اثنا عشر - *masculino* (cuando se refiere a un sustantivo masculino).
- ✓ **12:** اثنا عشر - *femenino* (cuando se refiere a un sustantivo femenino).

1.3.1.3. Números del 20 al 90:

A partir del número 20, los números se estructuran mediante la combinación de las unidades con las decenas. A continuación, se muestran las palabras para las decenas en esta tabla 1.3:

Número	En árabe
10	عشرة
20	عشرون
30	ثلاثون
40	أربعون
50	خمسون
60	ستون
70	سبعون
80	ثمانون
90	تسعون

Tabla 1.3. Decenas del 20 al 90 en árabe

Estos números se combinan con las unidades, utilizando la palabra "و", que significa "y", para unir las decenas con las unidades. Ejemplos:

- ✓ 21: واحد وعشرون - "uno y veinte".
- ✓ 35: خمسة وثلاثون - "cinco y treinta".
- ✓ 48: ثمانية وأربعون - "ocho y cuarenta".

Al igual que con los números anteriores, el género del número depende del sustantivo al que se refiere.

1.3.1.4. Números del 100 al 900:

Las centenas de 100 a 900 en árabe se forman de dos maneras, dependiendo del número. Para 100 y 200, existen palabras fijas, que se reflejan en la tabla 1.4:

Número	En árabe
100	مائة
200	مئتان

Tabla 1.4. Números 100 y 200 en árabe

Para los números del 300 al 900, se combinan las palabras usando las unidades del 3 al 9 con el مائة (cien). Esta combinación tiene dos formas de escritura:

- ✓ Forma simple: Usamos directamente la unidad seguida de مائة (cien).

- ✓ Forma compleja: se combina la unidad y la centena en una sola palabra en su forma genitiva.

Ahora, se presenta en la tabla 1.5 cómo se forman los números de las centenas en ambas formas:

Número	En árabe
300	ثَلَاثُ مِئَةٍ / ثلاثمائة
400	أَرْبَعُ مِئَةٍ / أربعمائة
500	خَمْسُ مِئَةٍ / خمسمائة
600	سِتُّ مِئَةٍ / ست مائة
700	سَبْعُ مِئَةٍ / سبعمائة
800	ثَمَانُ مِئَةٍ / ثمانمائة
900	تِسْعُ مِئَةٍ / تسعمائة

Tabla 1.5. Centenas en árabe

En cuanto al género, las centenas pueden utilizarse indistintamente en masculino o femenino.

Ejemplos de centenas con unidades y decenas:

- ✓ 150: مِئَةٌ وَخَمْسُونَ - "cien y cincuenta".
- ✓ 273: مِئَتَانِ وَثَلَاثَةٌ وَسَبْعُونَ - "doscientos y tres y setenta".
- ✓ 489: أَرْبَعَمِئَةٌ وَتِسْعَةٌ وَأَرْبَعُونَ - "cuatrocientos y nueve y cuarenta".

1.3.1.5. Números del 1000 al 9000:

Los números entre 1000 y 9000 se forman de manera simple, con palabras fijas para 1000 y 2000, y un patrón estructurado para los números mayores:

Las palabras para el 1000 y 2000, se reflejan en la tabla 1.6:

Número	En árabe
1000	أَلْفٌ
2000	أَلْفَانِ

Tabla 1.6. Miles en árabe

Para los números a partir de 3000, se utiliza la palabra **آلاف** (miles), que es la forma plural de **ألف** (mil) y la unidad correspondiente (del 3 al 9), como se ve en la siguiente tabla 1.7.

Número	En árabe
1000	ألف
2000	ألفان
3000	ثلاثة آلاف
4000	أربعة آلاف
5000	خمسة آلاف
6000	ستة آلاف
7000	سبعة آلاف
8000	ثمانية آلاف
9000	تسعة آلاف

Tabla 1.7. Números de mil a nueve mil en árabe

El género del número cambia dependiendo del sustantivo al que se refiere, y esto afecta a la unidad que precede a "mil" (la palabra "mil" en sí misma no cambia). Es decir, el género del número se ajusta de acuerdo con el sustantivo, y el número que va antes de "mil" es el que cambia de género.

1.3.1.6. A partir del número 1.000.000

Los números grandes, a partir de 1,000,000, se estructuran de manera específica con términos fijos internacionales para cada orden de magnitud. Comenzando con el número 1,000,000 (**مليون**), que significa "millón", se continúa con términos como **مليار** para "billón", y **تريليون** para "trillón", etc...

Esas magnitudes se presentan en la siguiente tabla 1.8

Número	En árabe
1,000	ألف
1,000,000	مليون
1,000,000,000	مليار
1,000,000,000,000	تريليون
1,000,000,000,000,000	كوادريليون
1,000,000,000,000,000,000	كوينتيليون
1,000,000,000,000,000,000,000	سكستيليون
1,000,000,000,000,000,000,000,000	سبتيليون

Tabla 1.8. Grandes números en árabe

1.3.2. Números ordinales:

Los números ordinales se utilizan para indicar la posición o el orden de algo dentro de una secuencia. Son similares a los adjetivos y se colocan antes o después del sustantivo al que acompañan, coincidiendo con él en género y número. A continuación, se muestran las formas de los números ordinales en árabe para diferentes rangos.

✓ Primero y Segundo:

- Para "primero" se utiliza **أَوَّل** en masculino y **أولى** en femenino.
- Para "segundo" se usa **ثَانٍ** en masculino y **ثَانِيَةٌ** en femenino.

✓ Tercero en adelante:

Los números ordinales del 3 al 10 se forman con palabras predefinidas derivadas de su forma cardinal:

- Tercero: **ثَالِث** en masculino y **ثَالِثَةٌ** en femenino.
- Cuarto: **رَابِع** en masculino y **رَابِعَةٌ** en femenino.
- Quinto: **خَامِس** en masculino y **خَامِسَةٌ** en femenino.
- Sexto: **سَادِس** en masculino y **سَادِسَةٌ** en femenino.
- Séptimo: **سَابِع** en masculino y **سَابِعَةٌ** en femenino.
- Octavo: **ثَامِن** en masculino y **ثَامِنَةٌ** en femenino.
- Noveno: **تَاسِع** en masculino y **تَاسِعَةٌ** en femenino.
- Décimo: **عَاشِر** en masculino y **عَاشِرَةٌ** en femenino.

✓ A partir del 11:

Los ordinales de estos números se construyen de forma similar a los cardinales. Para obtener el ordinal, se toma el número base (del 1 al 9) en formato ordinal con el sufijo **ال** y se le añade el resto del número en su forma natural. Por ejemplo:

- Undécimo: **الْحَادِي عَشَرَ** en masculino y **الْحَادِيَّةُ عَشْرَةٌ** en femenino.
- Duodécimo: **الثَّانِي عَشَرَ** en masculino y **الثَّانِيَّةُ عَشْرَةٌ** en femenino.
- Decimotercero: **الثَّالِثَ عَشَرَ** en masculino y **الثَّالِثَةُ عَشْرَةٌ** en femenino.
- Decimocuarto: **الرَّابِعَ عَشَرَ** en masculino y **الرَّابِعَةُ عَشْرَةٌ** en femenino.
- Decimoquinto: **الخَامِسَ عَشَرَ** en masculino y **الخَامِسَةُ عَشْرَةٌ** en femenino.
- Decimosexto: **السَّادِسَ عَشَرَ** en masculino y **السَّادِسَةُ عَشْرَةٌ** en femenino.
- Decimoséptimo: **السَّابِعَ عَشَرَ** en masculino y **السَّابِعَةُ عَشْرَةٌ** en femenino.
- Decimoctavo: **الثَّامِنَ عَشَرَ** en masculino y **الثَّامِنَةُ عَشْرَةٌ** en femenino.

- Decimonoveno: *التاسِعَ عَشَرَ* en masculino y *التاسِعَةَ عَشْرَةَ* en femenino.
- Vigésimo: *العشرون* en masculino y en femenino.

1.3.3. Números orientales:

Los números orientales son aquellos que se utilizan comúnmente en muchos países árabes y en otras partes del mundo musulmán. Son una variante de los números árabes estándar y se caracterizan por una forma escrita diferente, aunque la pronunciación es la misma. Estos números se emplean con frecuencia en documentos oficiales, señales de tráfico y en situaciones cotidianas, dependiendo del contexto y la región.

Su origen se remonta a los números indios, que fueron adaptados por los matemáticos árabes en la Edad Media. Durante este período, los matemáticos árabes adoptaron y perfeccionaron el sistema numérico decimal indio, desarrollando lo que hoy conocemos como los números arábigos, que más tarde se transmitieron a Europa y al resto del mundo. A medida que estos números se difundieron, los países árabes conservaron una versión gráfica distinta, dando lugar a los números orientales que se usan actualmente. Los números que conocemos en gran parte del mundo occidental son, en realidad, los que fueron introducidos por los árabes, cuya forma fue modificada a lo largo del tiempo para adaptarse a los estándares internacionales.

La principal diferencia entre los números orientales y los números árabes tradicionales radica en la forma gráfica de los dígitos. Mientras que los números árabes occidentales (usados en gran parte del mundo) utilizan una forma específica de dígitos, los números orientales presentan una apariencia distinta. Estos números son ampliamente utilizados en regiones del Golfo Pérsico, como Irán, Irak, Yemen, Omán, y en países del Levante, como Egipto, Jordania, Líbano y Siria.

En la siguiente tabla 1.9 se presenta cómo se escriben:

Número	Números orientales
0	٠
1	١
2	٢
3	٣
4	٤
5	٥
6	٦
7	٧
8	٨
9	٩
10	١٠
11	١١
12	١٢
13	١٣
14	١٤
15	١٥
16	١٦
17	١٧
18	١٨
19	١٩
20	٢٠

Tabla 1.9. Números orientales en árabe

En muchos países árabes, especialmente en documentos formales, periódicos y algunos entornos académicos, se emplean los números orientales en lugar de los números árabes occidentales. Sin embargo, en la vida diaria, es común ver tanto los números occidentales como los orientales, dependiendo de la preferencia regional o el contexto del texto.

1.3.4. Números multiplicativos:

Los números multiplicativos en árabe se utilizan para expresar las "veces". Es decir, se usan para indicar cuántas veces se realiza una acción o cuántas unidades de algo están involucradas. Para formar los números multiplicativos, simplemente se agrega la palabra "مرة", que significa "vez", al número cardinal correspondiente.

Por ejemplo:

- ✓ "مرة واحدة" significa "una vez", donde "واحدة" es el número 1 en femenino, indicando que la acción ocurre una sola vez.
- ✓ "ثلاث مرات" significa "tres veces", donde "ثلاث" es el número 3 y "مرات" es el plural de "مرة", indicando que la acción ocurre tres veces.

A continuación se muestra la tabla de los números multiplicativos del 1 al 10:

Número	En árabe
1	مرة واحدة
2	مرتين
3	ثلاث مرات
4	أربع مرات
5	خمس مرات
6	ست مرات
7	سبع مرات
8	ثمانى مرات
9	تسع مرات
10	عشر مرات

Tabla 1.10. Números multiplicativos en árabe

1.3.5. Números negativos:

Los números negativos se expresan colocando el signo negativo antes del número, en árabe se utiliza la palabra سالب, que significa "negativo". Esta palabra se coloca antes del número para indicar que se trata de un valor negativo. El uso de سالب es común en situaciones matemáticas, financieras, científicas y en cualquier contexto donde se necesite denotar que un número está por debajo de cero.

Ejemplos en la tabla 1.11

Número	Texto en árabe
-1	سالب واحد
-2	سالب اثنان
-3	سالب ثلاثة
-4	سالب أربعة
-5	سالب خمسة
-6	سالب ستة
-7	سالب سبعة
-8	سالب ثمانية
-9	سالب تسعة
-10	سالب عشرة

Tabla 1.11. Números negativos en árabe

1.4. La importancia de una herramienta para convertir números a texto

Tener una herramienta que convierta números a texto en árabe es esencial en algunos casos, especialmente en documentos legales o oficiales. En situaciones como cheques, contratos, facturas o cualquier otro documento que implique cantidades de dinero, es fundamental que los números se expresen tanto en cifras como en palabras. Esto no solo facilita la lectura, sino que también ayuda a prevenir posibles confusiones o fraudes, ya que la escritura del número en palabras asegura que se interprete de la misma forma por todas las partes involucradas.

1.4.1. Casos en los que se escriben números en texto en árabe

- ✓ **Cuando el número comienza una oración:** En general, los números que se encuentran al principio de una frase se recomienda escribirse en texto.
- ✓ **Números menores de cien:** En árabe, se recomienda escribir en texto los números del 1 al 99 en la escritura académica general y en textos no técnicos.
 - Ejemplo: خمسون طالب ("Cincuenta estudiantes").
- ✓ **Números redondos:** También se recomienda escribir en texto números como cien (مائة), mil (ألف), millones, etc., que son fáciles de escribir en una o dos palabras.
- ✓ **Números ordinales:** Los números ordinales (primero, segundo, tercero, etc.) deben escribirse en texto, especialmente cuando se mencionan en nombres o en situaciones que requieren formalidad.
 - Ejemplo: المرتبة الأولى ("El primer puesto").
- ✓ **En documentos legales u oficiales:** Cuando la claridad es necesaria, como en cheques, contratos, facturas o documentos legales, es necesario escribir los números en texto para evitar confusiones.
 - Ejemplo: مبلغ تسعمائة وأربعون جنيهًا ("La cantidad de novecientos cuarenta libras").
- ✓ **Cuando se mencionan cifras en contexto histórico o cultural:** En árabe, es más común escribir los números que representan períodos históricos o grandes eventos, como siglos o movimientos importantes, en texto.
 - Ejemplo: القرن العشرين ("El siglo XX").

1.5. Motivación:

La motivación detrás de este proyecto radica en la necesidad de herramientas que faciliten la conversión correcta de números a texto en árabe. La precisión en la escritura numérica es

crucial, sobre todo cuando se trata de cantidades grandes o complejas, donde un error podría tener consecuencias significativas.

A lo largo del desarrollo de este proyecto, se ha identificado que las soluciones existentes en la web no son lo suficientemente completas para cubrir todos los formatos necesarios, como los números cardinales, ordinales o las cifras arábigas orientales. Esto genera la oportunidad de crear una herramienta más completa, que no solo convierta números a texto en árabe, sino que también respete las particularidades del idioma y la forma en que deben estructurarse ciertos números. De esta forma, el proyecto se plantea como una solución que cubre una necesidad existente.

2. Justificación de las competencias específicas cubiertas

En cuanto a las competencias específicas cubiertas en este trabajo de fin de título, se destacan las siguientes:

- ✓ **CI1: Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos.** El trabajo consiste en desarrollar un servicio web para la conversión de cifras a texto en árabe, lo que implica el diseño y desarrollo de un servicio web que sea fiable, segura y de calidad.
- ✓ **CI6: Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas.** En este proyecto se implementan algoritmos para convertir números a distintos formatos, como cardinal, ordinal, fraccionario y romano, lo que permite aplicar los conocimientos sobre algoritmos para resolver problemas específicos.
- ✓ **CI8: Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente.** El trabajo incluye la creación de una aplicación que debe ser robusta, segura y eficiente, lo cual es fundamental para garantizar su funcionalidad y rendimiento a lo largo del tiempo.
- ✓ **TI2: Capacidad para seleccionar, diseñar, desplegar, integrar, evaluar, construir, gestionar, explotar y mantener las tecnologías de hardware, software y redes.** El uso de tecnologías como ASP.NET y WCF para la creación del servicio web cubre esta competencia, ya que se gestionan y se integran diversas tecnologías en el desarrollo del proyecto.
- ✓ **TFG: Ejercicio original a realizar individualmente y presentar y defender ante un tribunal universitario.** Este trabajo es un proyecto original en el que se sintetizan las competencias adquiridas en el Grado en Ingeniería Informática. El proyecto será defendido ante un tribunal, demostrando la aplicación práctica de los conocimientos adquiridos durante la carrera.

Este trabajo cubre diversas competencias específicas, desde el diseño y desarrollo de aplicaciones hasta la aplicación de metodologías de programación y la integración de tecnologías adecuadas, lo que me permite demostrar cómo he integrado los conocimientos adquiridos a lo largo de mi formación en ingeniería informática.

3. Aportaciones

Este proyecto ofrece una contribución significativa en diversos ámbitos socioeconómicos, técnicos y científicos, proporcionando soluciones valiosas para una amplia gama de sectores. A continuación, se detallan algunas de las áreas clave en las que esta herramienta de conversión de cifras a texto en árabe tiene un impacto directo:

- ✓ **Ámbito legal y administrativo**

En el entorno legal y administrativo, el proyecto mejora la claridad y precisión en documentos cruciales como contratos, cheques, facturas y otros textos legales. La herramienta garantiza que las cifras expresadas tanto en cifras como en palabras estén correctamente convertidas, lo que previene errores e interpretaciones erróneas, reduciendo el riesgo de fraudes. Esta utilidad resulta especialmente valiosa para abogados, notarios y entidades financieras, quienes necesitan una forma precisa y eficiente de redactar documentos.

- ✓ **Sector bancario y financiero**

En el ámbito financiero, especialmente en bancos y entidades de crédito, la herramienta facilita la correcta escritura de cifras en cheques, cálculos de intereses y documentos de pago. Su implementación puede ser útil en la validación de transacciones y documentos, reduciendo así los errores y mejorando la seguridad en las operaciones financieras.

- ✓ **Sector educativo y académico**

En el sector educativo, la herramienta puede ser una excelente herramienta de apoyo para profesores de matemáticas y literatura, ayudando a los estudiantes a comprender mejor los conceptos numéricos y su uso en contextos formales. Además, en campos como contabilidad y economía, donde las cifras juegan un papel fundamental, esta herramienta ayuda a ofrecer claridad en los contenidos académicos y facilita el aprendizaje.

- ✓ **Sectores públicos y gobiernos**

Los gobiernos y entidades públicas que gestionan una gran cantidad de trámites administrativos y documentos oficiales se beneficiarán enormemente de esta herramienta. La automatización de la conversión de cifras a texto en documentos legales, informes presupuestarios y leyes asegura la transparencia administrativa y mejora la precisión en la redacción de textos oficiales.

- ✓ **Sectores comerciales y empresariales**

En el sector comercial, empresas de diversos tamaños pueden integrar esta herramienta para automatizar la conversión de cifras a texto en contratos comerciales, presupuestos

y facturación electrónica. También podría mejorar la gestión administrativa en la generación de documentos comerciales, ayudando a mejorar la eficiencia y reduciendo la carga de trabajo manual.

- ✓ **Aplicaciones en inteligencia artificial y procesamiento de lenguaje natural**
Este proyecto también tiene implicaciones en el campo de la inteligencia artificial y el procesamiento de lenguaje natural. La conversión precisa de números a texto abre la puerta a aplicaciones avanzadas en sistemas de reconocimiento de voz y asistentes virtuales en árabe, mejorando la interacción entre humanos y máquinas, y brindando soluciones más efectivas para contextos multilingües.
- ✓ **Sector de comercio electrónico**
Las plataformas de comercio electrónico que operan en países árabes pueden integrar esta herramienta para la correcta generación de facturas y recibos de pago, tanto en cifras como en texto. Esta capacidad de conversión automatizada mejora la fiabilidad de las transacciones, lo que es clave para proporcionar una experiencia de cliente sin errores y conforme a la normativa legal.
- ✓ **Turismo y hospitalidad**
En el sector turístico, particularmente en países árabes, esta herramienta puede ser útil para la creación de documentos informativos y guías turísticas que requieren la conversión de cifras a texto en árabe. Hoteles, agencias de viajes y operadores turísticos pueden aprovecharla para mejorar la claridad y la precisión en los contratos y los materiales informativos ofrecidos a los clientes.
- ✓ **Aplicaciones en salud**
En el sector salud, donde se manejan cantidades y datos precisos en informes médicos, recetas y facturación de tratamientos, la herramienta puede garantizar que los números sean expresados correctamente, contribuyendo a la seguridad y fiabilidad de los documentos médicos. La conversión automática también puede ser útil en la gestión administrativa del sistema de salud.
- ✓ **Creación de herramientas de accesibilidad**
Para las personas con discapacidad visual o dificultades en la lectura, esta herramienta podría ser integrada en lectores de pantalla o sistemas de asistencia para facilitar la comprensión de documentos que contienen cifras complejas. Esto contribuiría a la creación de soluciones inclusivas, mejorando la accesibilidad a la información en diferentes contextos.

En conclusión, este proyecto no solo tiene un impacto técnico significativo, al mejorar la conversión de números a texto en árabe, sino que también tiene amplias aplicaciones en sectores clave como el legal, financiero, educativo, comercial, tecnológico y gubernamental. Además, al integrarse con tecnologías de inteligencia artificial y procesamiento de lenguaje natural, ofrece potencial de crecimiento hacia aplicaciones más avanzadas y diversas en el futuro.

Además se ha planeado la integración del servicio en un proyecto de mayor envergadura

del Instituto Universitario de Análisis y Aplicaciones Textuales (IATEXT) llamado Números TIP, un proyecto que ofrece una aplicación web que permite a los usuarios convertir números a texto en varios idiomas y formatos.

El proyecto ha recibido más de 19 millones de consultas de más de 5 millones de usuarios de todo el mundo en los últimos 12 meses, según Google Analytics, como se ve en la figura 3.1.



Figure 3.1: Vista de usuarios activos según Google Analytics.

3.1. Alineamiento con los objetivos de desarrollo sostenible

Justificar el alineamiento del TFT con los **ODS** con los que presente un mayor grado de relación, en función de lo indicado en la tabla 3.1.

Tabla 3.1. Grado de relación del TFT con los objetivos de desarrollo sostenible.

ODS	Grado de relación			
	0 No procede	1 Bajo	2 Medio	3 Alto
1 Fin de la Pobreza	x			
2 Hambre cero	x			
3 Salud y Bienestar	x			
4 Educación de calidad			x	
5 Igualdad de género	x			
6 Agua limpia y saneamiento	x			
7 Energía Asequible y no contaminante	x			
8 Trabajo decente y crecimiento económico	x			
9 Industria, Innovación e Infraestructuras	x			
10 Reducción de las desigualdades	x			
11 Ciudades y comunidades sostenibles	x			
12 Producción y consumo sostenibles	x			
13 Acción por el clima	x			
14 Vida submarina	x			
15 Vida de ecosistemas terrestres	x			
16 Paz, justicia e instituciones sólidas	x			
17 Alianzas para lograr objetivos	x			

4. Diseño

El proyecto se desarrolló en Visual Studio, con el frontend implementado con ASP.NET y un servicio web desarrollado con C#. El frontend permite a los usuarios ingresar cifras y visualizar la conversión a texto en árabe en varias formas, como cardinal, ordinal, romano, y en formato oriental (dígitos árabes tradicionales).

4.1. Tecnologías:

Antes de adentrarnos en el desarrollo específico del proyecto, es importante dedicar un apartado a las tecnologías que se utilizarán para llevar a cabo la conversión de cifras a texto en árabe, así como su implementación. En este apartado, se explican las tecnologías clave que se emplearán, detallando sus características y beneficios.

4.1.1. Programación orientada a objetos (POO)

La programación orientada a objetos (POO) [6] es un paradigma que organiza el código en "objetos", que son instancias de clases. Cada objeto contiene atributos, que representan los datos, y métodos, que son funciones que operan sobre esos datos. Este enfoque modular permite crear aplicaciones más escalables y fáciles de mantener, lo que resulta ideal para proyectos complejos, como el sistema de conversión de cifras a texto en árabe.

La POO ofrece varias características clave que la hacen altamente eficiente para el desarrollo de software. Una de ellas es la reusabilidad, ya que el código de las clases y objetos puede ser reutilizado en distintas partes del proyecto, o incluso en proyectos futuros, lo que optimiza el tiempo de desarrollo. También fomenta la modularidad, ya que permite dividir el proyecto en componentes independientes, facilitando la gestión y el mantenimiento del código. La abstracción es otro aspecto importante, ya que oculta los detalles internos de las clases y solo expone la funcionalidad esencial, mejorando tanto la seguridad como la simplicidad del sistema. Además, la POO mejora la facilidad de mantenimiento, ya que las clases bien definidas permiten realizar modificaciones sin afectar otras partes del código.

La implementación de la POO en el proyecto, como la conversión de números a diferentes formatos (cardinal, ordinal, romano, oriental), es crucial para la organización y comprensión

del sistema. Las clases encargadas de representar los conversores de números se estructuran de manera independiente, lo que facilita tanto su comprensión como su mantenimiento.

4.1.2. Visual Studio

Visual Studio [7][8] es un entorno de desarrollo integrado (IDE) altamente popular entre los desarrolladores de software. Es ampliamente utilizado para el desarrollo de aplicaciones en diversos lenguajes de programación, como C#, VB.NET, C++, Python, entre otros. Visual Studio proporciona un conjunto de herramientas poderosas para escribir, depurar, probar y mantener aplicaciones de manera eficiente. Su integración con tecnologías de Microsoft, como .NET Framework, facilita el desarrollo de aplicaciones robustas y de alto rendimiento.

También ofrece un editor de código avanzado que facilita el trabajo gracias a funcionalidades como el autocompletado, la depuración en tiempo real, el resaltado de sintaxis y la refactorización de código, lo que permite escribir código de manera más eficiente y libre de errores.

Además, Visual Studio tiene una integración completa con Git, lo que permite llevar un control de versiones eficaz y facilita la colaboración entre los miembros de un equipo de desarrollo. La depuración avanzada es otra característica clave, ya que ofrece potentes herramientas que permiten encontrar y corregir errores rápidamente, lo cual es esencial cuando se trabaja en proyectos complejos.

Por último, Visual Studio es altamente extensible, lo que significa que puedes agregar una gran variedad de extensiones que mejoran tu flujo de trabajo y adaptan el entorno a tus necesidades específicas, haciendo que el desarrollo sea aún más productivo y eficiente.

Ventajas en este proyecto: Visual Studio es una buena herramienta para desarrollar este proyecto debido a su compatibilidad con el desarrollo en C# y su integración con ASP.NET y WCF, dos tecnologías clave para este proyecto.

4.1.3. Servicio Web - WCF (Windows Communication Foundation)

Windows Communication Foundation (WCF) [9] es una plataforma de Microsoft que permite crear aplicaciones distribuidas y servicios web. Al formar parte del .NET Framework [10], facilita la construcción de aplicaciones que se comunican a través de diferentes protocolos, como HTTP, TCP y más. Es ideal para proyectos que necesitan interoperar con diferentes plataformas y que deben manejar grandes volúmenes de datos de forma eficiente.

Una de sus características más destacadas es la interoperabilidad, ya que permite que los servicios creados en WCF se conecten y trabajen con otros sistemas, basados en Java o PHP. Además, incluye herramientas de seguridad que permiten proteger las comunicaciones, como la autenticación, la autorización, la encriptación de mensajes y la gestión de identidades.

WCF también está diseñado para ser altamente escalable, lo que lo convierte en una excelente opción para aplicaciones que necesitan gestionar grandes cantidades de usuarios de manera simultánea. Gracias a su flexibilidad, permite crear servicios que pueden ser consumidos de manera síncrona o asíncrona, adaptándose a las necesidades específicas de cada cliente y mejorando la eficiencia en la comunicación.

4.1.4. ASP.NET

ASP.NET es un marco de desarrollo de aplicaciones web creado por Microsoft que se utiliza para construir aplicaciones dinámicas y servicios web. Se basa en el .NET Framework y es compatible con varios lenguajes de programación, siendo C# el más utilizado. ASP.NET permite el desarrollo de aplicaciones web modernas, escalables y seguras mediante un enfoque estructurado y modular [11].

Una de las principales ventajas de ASP.NET es su desarrollo rápido, ya que proporciona herramientas y bibliotecas que agilizan la creación de aplicaciones web. Además, ofrece compatibilidad con tecnologías web estándar, lo que facilita la integración de interfaces de usuario modernas. En términos de seguridad, ASP.NET incluye características integradas para proteger las aplicaciones contra ataques comunes, como la inyección de SQL, XSS y CSRF. Por último, el despliegue sencillo de aplicaciones desarrolladas en ASP.NET permite su implementación en servidores web que soporten el .NET Framework o .NET Core.

4.1.5. C#

C# (C-Sharp) es un lenguaje de programación de alto nivel desarrollado por Microsoft como parte del .NET Framework. C# es un lenguaje orientado a objetos que facilita la creación de aplicaciones robustas, seguras y escalables. Su sintaxis está inspirada en C, C++ y Java, y se ha convertido en uno de los lenguajes más populares en el desarrollo de software [12].

Una de las características destacadas de C# es su gestión automática de memoria a través del recolector de basura (garbage collector), lo que reduce la posibilidad de errores relacionados con la memoria y mejora la estabilidad de las aplicaciones. Además, C# ofrece un sistema de tipos fuerte, lo que significa que las conversiones de tipos no compatibles generan errores en tiempo de compilación, ayudando a detectar problemas antes de la ejecución del programa.

C# también es conocido por su compatibilidad multiplataforma, lo que permite desarrollar aplicaciones que pueden ejecutarse en diferentes sistemas operativos, como Windows, Linux y macOS. Esta característica se ve potenciada por el uso de .NET Core, una versión modular y de código abierto del .NET Framework, que facilita la creación de aplicaciones que se ejecutan en diversas plataformas sin necesidad de modificaciones significativas en el código.

4.1.6. Paralelismo

Se ha implementado concurrencia para ejecutar las diferentes conversiones en paralelo utilizando la librería Task Parallel Library (TPL). En el contexto del proyecto, se emplea TPL para manejar varias conversiones de números en diferentes formatos, como cardinal, ordinal, romano y multiplicativo, permitiendo que se realicen en paralelo de forma eficiente.

La Task Parallel Library (TPL) [13] es una herramienta de .NET para gestionar la ejecución de múltiples tareas de manera asíncrona. Proporciona una interfaz sencilla para crear, ejecutar y controlar tareas en paralelo. A través de clases como Task, Parallel y Task.WhenAll, TPL permite la ejecución de operaciones simultáneas, lo que resulta en aplicaciones más rápidas y escalables. TPL maneja automáticamente la asignación de tareas a los hilos disponibles, optimizando el uso de recursos y gestionando la sincronización de manera interna.

4.2. Backend:

El backend del proyecto está compuesto por un servicio web desarrollado en C# utilizando WCF. Este servicio se encarga de recibir las solicitudes de conversión y devolver los resultados correspondientes. A través de este servicio, la página web de frontend interactúa con el sistema de conversión de números a texto en árabe, manejando los diferentes tipos de conversiones y formatos de entrada.

4.2.1. Estructura del servicio web

El servicio web está implementado en la clase `ServicioNumerosArabe`, que implementa la interfaz `IServicioNumerosArabes`. Esta interfaz define el método principal, `Traducir`, el cual recibe un número en formato de texto y un idioma de destino, y devuelve una tupla que contiene una cabecera y una lista de conversiones.

Y se organiza de la siguiente forma:

- ✓ **Cabecera de conversión:** Proporciona detalles sobre la conversión, como el número de entrada, el número formateado, y el título de cada sección de conversión.
- ✓ **Lista de conversiones:** Contiene las conversiones en diferentes formatos como cardinal, ordinal, romano, multiplicativo, y en formato oriental. Cada conversión se devuelve junto con una lista de ejemplos y notas relevantes para el tipo de conversión.

4.2.2. Entrada

El proceso de conversión se inicia cuando se envía una solicitud al servicio web, pasando el número ingresado por el usuario. El número puede ser expresado en varios formatos, tales

como números enteros, decimales, o números seguidos de una divisa. Según el tipo de número y el formato, el servicio ejecuta los métodos apropiados que convierten ese número a texto.

✓ **Entrada de número permitidas:**

- Número entero:
 - El número puede comenzar opcionalmente con un signo más (+) o menos (-), indicando si es positivo o negativo.
 - Debe haber entre 1 y 3582 dígitos numéricos. Y no puede haber otros caracteres que no sean números.
 - Después de los dígitos, puede haber un símbolo de divisa como (\$, €, £, ¥), que se utiliza para representar moneda.
- Número decimal:
 - Un signo más (+) o menos (-) puede preceder al número, indicando si el número es positivo o negativo.
 - Debe haber entre 1 y 3582 dígitos numéricos antes del punto decimal.
 - Un punto (.) debe aparecer, separando la parte entera de la parte decimal. La parte decimal puede tener hasta 3582 dígitos numéricos.
 - La suma total de los dígitos de la parte decimal y entera, no puede superar 3582 dígitos.

4.2.3. Salida

El tipo de resultados que proporciona el servicio depende de la longitud y el formato del número a convertir.

- ✓ **Cardinal y decimal:** Estos resultados son generados por la clase `ConvertorNumerosArabes`, que convierte el número a texto siguiendo las reglas del idioma árabe.
- ✓ **Ordinal:** Similar a los cardinales, pero con una modificación en el formato para indicar la posición en una secuencia.
- ✓ **Romano:** Utiliza la clase `ConvertorRomano` para convertir un número en su formato romano. Por ejemplo, el número "5" se convierte en "V".
- ✓ **Multiplicativo:** Para números como "2" que se expresan como "dos veces" en árabe, se usa la clase `ConvertorMultiplicativo`.
- ✓ **Oriental:** Los números en formato occidental se convierten a su equivalente en dígitos árabes tradicionales utilizando la clase `ConvertorOriental`.

4.2.4. Descripción de las clases de conversión del backend

A continuación, se detallan los métodos utilizados en el backend para la conversión de números en árabe. Cada uno de estos métodos se encuentra dentro de una clase específica que se encarga de realizar la conversión de números en diferentes formatos (cardinal, ordinal, romano, oriental, multiplicativo).

4.2.4.1. Clase ‘ConversorNumerosArabes‘

La clase ‘ConversorNumerosArabes‘ se encarga de convertir números enteros o decimales en formato cardinal en texto árabe. Esta clase utiliza varios arrays predefinidos que contienen las representaciones en árabe de unidades, decenas, centenas y escalas mayores.

- ✓ **Entrada:** La clase toma como entrada un número en formato de texto (cadena de caracteres) que representa un número entero.
- ✓ **Proceso:** La clase divide el número en grupos de tres dígitos, procesando cada grupo de manera individual. Convierte cada parte del número a su equivalente en árabe, uniendo los resultados con ”,” (y) cuando es necesario.
- ✓ **Salida:** La salida es el número convertido a texto árabe en formato cardinal.

4.2.4.2. Método ‘Convertir‘

El método ‘Convertir‘ es el principal encargado de procesar un número (ya sea entero o decimal) para su conversión. Este proceso se lleva a cabo de manera sistemática:

- ✓ **Entrada:**
 - **numero** (string): El número que se va a convertir.
 - **genitivo** (bool): Indica si el número debe ser genitivo.
 - **femenino** (bool): Indica si el número debe ser femenino.
- ✓ **Proceso:**
 - Si el número comienza con un signo negativo (“-”), se marca como negativo y se elimina el signo.
 - Si el número tiene una moneda asociada (como “\$”, “€”, “£”, etc.), y si es así, elimina el símbolo de la moneda y lo almacena para ser agregado más tarde.
 - Si el número contiene un punto (.), se separa en parte entera y parte decimal.
 - La parte entera y decimal se convierte utilizando el método ‘ConvertirCardinal‘, que se explica en el siguiente apartado.
 - Si existe una parte decimal, se agrega la palabra فاصل seguida de la conversión de la parte decimal.

- Si se detectó una moneda, se agrega su nombre en árabe.
 - Si el número es negativo, se antepone la palabra `سالِب`.
- ✓ **Salida:** El número convertido en formato árabe, con la parte entera, la parte decimal (si existe), y la moneda (si se detectó). Si es negativo, también incluirá el prefijo `سالِب`.

Método ‘ConvertirCardinal‘

El método ‘ConvertirCardinal‘ se encarga de convertir un número entero (sin decimales) a su representación en árabe. Este proceso se realiza en tres etapas:

✓ **Entrada:**

- **numero** (string): El número entero a convertir.
- **genitivo** (bool): Indica si el número debe ser genitivo.
- **femenino** (bool): Indica si el número debe ser femenino.

✓ **Proceso:**

- El número se rellena con ceros a la izquierda si no tiene una longitud múltiplo de tres.
- El número se divide en grupos de tres dígitos (por ejemplo, "123456" se convierte en "123" y "456"), llamando al método ‘ConvertirGrupo‘.
- Cada grupo de tres dígitos se convierte utilizando las unidades, decenas y centenas, y se le agrega la escala correspondiente (mil, millón, etc.) si es necesario.
- Los resultados de los grupos se combinan con la palabra `و` (y) para formar el número final.

- ✓ **Salida:** El número entero convertido a texto árabe en formato cardinal, con las escalas correspondientes.

Método ‘ConvertirGrupo‘

Este método se encarga de convertir un grupo de tres dígitos a su forma en árabe. El grupo puede estar compuesto por centenas, decenas y unidades.

✓ **Entrada:**

- **numero** (int): El grupo de tres dígitos a convertir.
- **genitivo** (bool): Indica si el número debe ser genitivo.
- **femenino** (bool): Indica si el número debe ser femenino.

✓ **Proceso:**

- El número se descompone en centenas, decenas y unidades.
- Se convierte cada parte a su forma en árabe utilizando los arrays de unidades, decenas y centenas.
- Si el número tiene una forma genitiva, se realiza el ajuste correspondiente.

✓ **Salida:** El grupo convertido a su forma en árabe.

Método ‘EscalaDeNumeros‘

Este método maneja las escalas de los números, como "mil" (ألف), "millón" (مليون), "mil millones" (مليار), etc.

✓ **Entrada:**

- **escalaIndice** (int): El índice de la escala.
- **grupoValor** (int): El valor del grupo.

✓ **Proceso:** El método selecciona la escala correspondiente (singular, dual o plural) y la devuelve.

✓ **Salida:** El nombre de la escala correspondiente (por ejemplo, ألف para mil).

Tabla del método convertir 4.1

Nombre	Convertir
Descripción	Convierte un número en formato de texto a su equivalente en árabe, teniendo en cuenta las unidades, decenas y centenas, y agregando la moneda si es necesario.
Argumentos de entrada	<ul style="list-style-type: none"> ✓ numero (string): Número a convertir a árabe. ✓ genitivo (bool): Indica si el número debe ser en genitivo. ✓ femenino (bool): Indica si el número debe ser femenino.
Argumentos de salida	<ul style="list-style-type: none"> ✓ resultado (string): Número convertido a su forma en árabe con su respectiva moneda y signo de negativo si es necesario.
Valor devuelto	<ul style="list-style-type: none"> ✓ string: Número en formato árabe con su respectiva moneda y signo negativo (si es necesario).

Tabla 4.1. Método convertir de la clase ConversorNumerosArabes

Tabla del método ConvertirCardinal 4.2

Nombre	ConvertirCardinal
Descripción	Convierte un número a su forma cardinal en árabe, dividiendo el número en grupos de tres dígitos y convirtiéndolos individualmente.
Argumentos de entrada	<ul style="list-style-type: none"> ✓ numero (string): Número a convertir a formato cardinal. ✓ genitivo (bool): Indica si el número debe ser en genitivo. ✓ femenino (bool): Indica si el número debe ser femenino.
Argumentos de salida	<ul style="list-style-type: none"> ✓ resultado (string): Número convertido a su forma cardinal.
Valor devuelto	<ul style="list-style-type: none"> ✓ string: Número en formato cardinal en árabe.

Tabla 4.2. Método ConvertirCardinal de la clase ConversorNumerosArabes

Tabla del Método ConvertirGrupo 4.3

Nombre	ConvertirGrupo
Descripción	Convierte un número de hasta tres dígitos a su representación en árabe, teniendo en cuenta las centenas, decenas y unidades.
Argumentos de entrada	<ul style="list-style-type: none"> ✓ numero (int): Número a convertir a formato árabe. ✓ genitivo (bool): Indica si el número debe ser en genitivo. ✓ femenino (bool): Indica si el número debe ser femenino.
Argumentos de salida	<ul style="list-style-type: none"> ✓ restoTexto (string): Número de tres dígitos convertido a su forma árabe.
Valor devuelto	<ul style="list-style-type: none"> ✓ string: Número de tres dígitos convertido a su representación en árabe.

Tabla 4.3. Método ConvertirGrupo de la clase ConversorNumerosArabes

Tabla del Método EscalaDeNumeros 4.4

Nombre	EscalaDeNumeros
Descripción	Determina la escala (unidad, mil, millón, etc.) de un número dependiendo de su valor y su grupo.
Argumentos de entrada	<ul style="list-style-type: none"> ✓ escalaIndice (int): Índice de la escala para el número (unidad, mil, millón, etc.). ✓ grupoValor (int): Valor del grupo de tres dígitos.
Argumentos de salida	<ul style="list-style-type: none"> ✓ scale.Singular (string): Singular para la escala. ✓ scale.Dual (string): Dual para la escala. ✓ scale.Plural (string): Plural para la escala.
Valor devuelto	<ul style="list-style-type: none"> ✓ string: Nombre de la escala correspondiente en árabe.

Tabla 4.4. Método EscalaDeNumeros de la clase ConversorNumerosArabes

4.2.4.3. Clase ‘ConversorOrdinal’

La clase `ConversorOrdinal` se encarga de convertir números a su forma ordinal en árabe. A diferencia de la clase `ConversorNumerosArabes`, que convierte números cardinales, esta clase se especializa en convertir números a su representación ordinal, tomando en cuenta el género (masculino o femenino).

- ✓ **Entrada:** La clase recibe un número en formato String que representa un número entero.
- ✓ **Proceso:** La clase procesa el número, convirtiéndolo a su forma ordinal en árabe. Este proceso incluye la consideración de las unidades, decenas, centenas y escalas, con la flexibilidad de manejar números en forma femenina y masculina. Se puede incluir un prefijo especial "ال" al inicio del número para su correcta representación en árabe.
- ✓ **Salida:** La salida es el número convertido a texto árabe en formato ordinal, con el prefijo apropiado ("ال") cuando sea necesario, y teniendo en cuenta el género seleccionado.

Método ‘ConvertirOrdinal’

El método `ConvertirOrdinal` es el principal responsable de convertir un número entero en su forma ordinal en árabe. Este proceso se lleva a cabo de manera detallada:

- ✓ **Entrada:**
 - **numero** (string): El número que se va a convertir a ordinal.
 - **femenino** (bool): Indica si el número debe ser femenino. Por defecto, el número será masculino.

✓ **Proceso:**

- El método verifica que el número no esté vacío ni contenga caracteres no numéricos.
- Si el número contiene ceros a la izquierda, estos se eliminan.
- El número se divide en grupos de tres dígitos para su conversión individual.
- Cada grupo de tres dígitos se convierte utilizando la función `ConvertirGrupoOrdinal`, que convierte unidades, decenas y centenas en su forma ordinal en árabe.
- Si el número tiene una escala mayor (mil, millón, etc.), el método aplica las escalas correspondientes utilizando la función `EscalaDeNumeros`.
- Si el número es mayor que 10, se trata de los “ordinales para los números en las decenas” (por ejemplo, 20, 30, etc.), usando las representaciones adecuadas para los ordinales de las decenas en árabe.

✓ **Salida:** El número convertido a su forma ordinal en árabe, representado con el formato adecuado según su género y escalas.

Método ‘`ConvertirGrupoOrdinal`’

Este método se encarga de convertir un grupo de tres dígitos en su forma ordinal en árabe, manejando las unidades, decenas y centenas:

✓ **Entrada:**

- `numero` (int): El grupo de tres dígitos a convertir.
- `femenino` (bool): Indica si el número debe ser femenino.
- `escalaIndice` (int): Índice de la escala correspondiente (para mil, millón, etc.).

✓ **Proceso:**

- El número se descompone en centenas, decenas y unidades.
- Las partes del número (centenas, decenas y unidades) se convierten a su forma ordinal utilizando los arrays predefinidos.
- Si el número tiene una forma genitiva, se ajusta el formato para reflejar dicha característica.
- El método también maneja los ordinales de los números de las decenas (por ejemplo, "عشرون" o "ثلاثون") usando un array predefinido.

✓ **Salida:** El grupo de tres dígitos convertido a su forma ordinal en árabe.

Tabla del método `ConvertirOrdinal` 4.5

Nombre	ConvertirOrdinal
Descripción	Convierte un número a su forma ordinal en árabe, tomando en cuenta el género.
Argumentos de entrada	<ul style="list-style-type: none"> ✓ numero (string): Número a convertir a ordinal. ✓ femenino (bool): Indica si el número debe ser femenino (opcional, valor por defecto es 'false').
Argumentos de salida	<ul style="list-style-type: none"> ✓ t_Numero (string): Número convertido a su forma ordinal en árabe.
Valor devuelto	<ul style="list-style-type: none"> ✓ string: Número ordinal en árabe, precedido de " " si es necesario.

Tabla 4.5. Tabla del Método ConvertirOrdinal de la clase ConversorOrdinal

Método ConvertirGrupoOrdinal 4.6

Nombre	ConvertirGrupoOrdinal
Descripción	Convierte un número dentro de un grupo (hasta 999) a su forma ordinal en árabe, considerando el género y la escala del número.
Argumentos de entrada	<ul style="list-style-type: none"> ✓ numero (int): Número a convertir dentro de un grupo (hasta 999). ✓ femenino (bool): Indica si el número debe ser femenino. ✓ escalaIndice (int): Índice de la escala para determinar el sufijo ordinal (miles, centenas, etc.).
Argumentos de salida	<ul style="list-style-type: none"> ✓ grupo_Texto (string): Número dentro del grupo convertido a su forma ordinal en árabe.
Valor devuelto	<ul style="list-style-type: none"> ✓ string: Número en su forma ordinal dentro de su grupo.

Tabla 4.6. Método ConvertirGrupoOrdinal de la clase ConversorOrdinal

4.2.4.4. Clase 'ConversorOriental'

La clase `ConversorOriental` se encarga de convertir números de su formato occidental (dígitos árabes comunes) a su formato oriental, utilizado en la mayoría de los países árabes.

- ✓ **Entrada:** La entrada es un número en formato de texto que contiene dígitos árabes occidentales (por ejemplo, 1, 2, 3, etc.).
- ✓ **Proceso:**
 - Si el número es negativo, se guarda esa información para agregar el signo "-" al resultado final.
 - Si el número contiene caracteres no numéricos o no es válido, se devuelve un mensaje de error.
 - Se define un array que mapea los dígitos árabes occidentales (0, 1, 2, ...) a sus equivalentes en árabe oriental (٠, ١, ٢, ...).
 - Cada carácter del número ingresado es recorrido, y si el carácter es un dígito, se reemplaza por su equivalente en árabe oriental.
 - Si el número es negativo, se agrega el signo "-" al principio del número convertido.
- ✓ **Salida:** La salida es el número convertido a su formato de dígitos árabes orientales.

Método 'ConvertirANumeroOriental'

Este es el principal método de la clase `ConvertorOriental`, y se encarga de convertir un número de su formato original a su formato oriental.

- ✓ **Entrada:**
 - **numero** (string): El número en formato occidental (con dígitos árabes comunes) que se va a convertir a formato oriental.
- ✓ **Proceso:**
 - Si el número es negativo, se marca el valor `esNegativo` como `true` y se elimina el signo negativo del número para poder convertirlo correctamente.
 - Si el número contiene caracteres no numéricos o es inválido (no es un número o no contiene dígitos), se devuelve el mensaje de error "Número no válido".
 - Se define un array llamado `arabicDigits`, que contiene los equivalentes en árabe oriental de los dígitos occidentales (0-9).
 - El número es recorrido carácter por carácter. Si el carácter es un dígito, se reemplaza por su equivalente en árabe oriental. Si es cualquier otro símbolo (por ejemplo, un punto decimal o un símbolo de moneda), se mantiene tal cual.
 - Si el número es negativo, se agrega el signo "-" al inicio del número convertido.
- ✓ **Salida:** El número convertido a su formato de dígitos árabes orientales.

Tabla del Método 'ConvertirANumeroOriental' 4.7

Nombre	ConvertirANumeroOriental
Descripción	Convierte un número de su forma occidental a su forma en dígitos árabes orientales.
Argumentos de entrada	✓ numero (string): Número en formato occidental a convertir.
Argumentos de salida	✓ (string): Número convertido a su forma oriental.
Valor devuelto	✓ string : Número en formato árabe oriental.

Tabla 4.7. Método ConvertirANumeroOriental de la clase ConversorOriental

4.2.4.5. Clase ‘ConversorRomano‘

La clase `ConversorRomano` se encarga de convertir un número entero o decimal a su equivalente en números romanos, un sistema de numeración utilizado en diversas aplicaciones como numeraciones de capítulos de libros, siglos, entre otros.

- ✓ **Entrada:** La clase toma como entrada un número en formato de texto (cadena de caracteres) que representa un número decimal. El número debe ser válido y no debe ser igual a cero.
- ✓ **Proceso:**
 - Primero, se verifica si el número es válido. Si el número no es un dígito, no puede ser convertido y se retorna un mensaje de error.
 - El número es procesado de tres en tres dígitos (por cada grupo de tres dígitos se genera su correspondiente valor romano).
 - Se recorre cada grupo de tres dígitos del número, y para cada grupo, se genera la correspondiente representación romana utilizando el método `ConvertirGrupoRomano`.
 - Si el número contiene más de tres dígitos, se aplica una barra encima de cada símbolo romano (usando el método `AplicarBarras`), que indica la multiplicación del valor romano por 1000.
- ✓ **Salida:** La salida es el número convertido a su formato en números romanos.

Método ‘ConvertirARomano‘

Este es el principal método de la clase `ConversorRomano`. Se encarga de convertir el número decimal a su formato romano, utilizando los valores estándar de los números romanos.

- ✓ **Entrada:**

- **numero** (string): El número a convertir a romano.

✓ **Proceso:**

- Si el número es inválido o igual a cero, se retorna un mensaje de error.
- El número es dividido en grupos de tres dígitos.
- Para cada grupo de tres dígitos, se llama al método `ConvertirGrupoRomano`, que devuelve la representación romana del grupo.
- Si hay más de tres dígitos, se aplica la barra (indicando multiplicación por 1000) usando el método `AplicarBarras`.

- ✓ **Salida:** El número convertido a su representación en romano, con barras sobre los valores romanos si es necesario.

Método ‘ConvertirGrupoRomano’

Este método convierte un grupo de tres dígitos a su forma romana.

✓ **Entrada:**

- **valor** (int): El valor del grupo de tres dígitos a convertir.

✓ **Proceso:**

- Se usa un conjunto de tuplas (**simbolos**) que asigna un valor romano a su correspondiente valor numérico (por ejemplo, 1000 = "M", 500 = "D", 100 = "C").
- El valor del grupo se va reduciendo y se va concatenando el símbolo romano correspondiente hasta que se alcanza 0.

- ✓ **Salida:** La forma romana correspondiente al grupo de tres dígitos.

Método ‘AplicarBarras’

Este método aplica barras sobre los símbolos romanos para representar números mayores de 1000.

✓ **Entrada:**

- **romano** (string): El valor romano.
- **nivel** (int): El nivel de barras que se debe aplicar.

✓ **Proceso:**

- Si el nivel es mayor que 0, se añaden barras sobre cada carácter del número romano usando el símbolo Unicode de la barra.

- ✓ **Salida:** El número romano con las barras aplicadas, si es necesario.

Tabla del Método ‘ConvertirARomano‘ 4.8

Nombre	ConvertirARomano
Descripción	Convierte un número decimal a su forma romana.
Argumentos de entrada	✓ numero (string): Número a convertir.
Argumentos de salida	✓ t_Numero (string): Número convertido a su forma romana.
Valor devuelto	✓ string : Número en formato romano.

Tabla 4.8. Método ConvertirARomano de la clase ConversorRomano

4.2.4.6. Clase ‘ConversorMultiplicativo‘

La clase `ConversorMultiplicativo` convierte números cardinales a su forma multiplicativa en árabe. Este formato se utiliza en contextos matemáticos o de frecuencia para expresar "veces" o "por". Por ejemplo, si queremos expresar una cantidad multiplicada por un número específico, como "tres veces", utilizamos esta forma multiplicativa.

- ✓ **Entrada:** La clase toma como entrada un número en formato de texto (cadena de caracteres) que se desea convertir a su forma multiplicativa en árabe.
- ✓ **Proceso:** La clase convierte el número cardinal a su representación en árabe, utilizando la lógica de la clase `ConversorNumerosArabes`. Luego, le agrega el sufijo adecuado para indicar la multiplicación. Por ejemplo, si el número es mayor que 1, se le añade el sufijo ". " "
- ✓ **Salida:** La salida es el número convertido en su forma multiplicativa en árabe, con el sufijo correspondiente dependiendo del número.

Método ‘ConvertirMultiplicativo‘

El método `ConvertirMultiplicativo` convierte un número cardinal a su representación multiplicativa en árabe. Este método es útil para expresar cantidades multiplicadas, como "veces" o "por" en situaciones matemáticas o de frecuencia.

- ✓ **Entrada:**
 - **numero** (string): El número a convertir a su forma multiplicativa en árabe.
- ✓ **Proceso:**
 - La clase convierte el número cardinal en su representación en árabe.
 - Si el número es 1, se utiliza el formato *مرة واحدة* (una vez).

- Si el número es 2, se utiliza el formato مَرَّتَان (dos veces).
- Para números mayores que 2, se utiliza el formato مَرَّات con el número cardinal convertido previamente.

✓ **Salida:**

- La salida es el número convertido a su forma multiplicativa en árabe, con el sufijo adecuado dependiendo del número.

Tabla del Método ‘ConvertirMultiplicativo’

Nombre	ConvertirMultiplicativo
Descripción	Convierte un número a su forma multiplicativa en árabe.
Argumentos de entrada	✓ numero (string): Número a convertir.
Argumentos de salida	✓ t_Numero (string): Número convertido a su forma multiplicativa.
Valor devuelto	✓ string : Número multiplicativo en árabe.

Tabla 4.9. Método ConvertirMultiplicativo de la clase ConversorMultiplicativo

4.2.5. Integración con el frontend

El servicio está diseñado para ser fácilmente integrable con el frontend, utilizando el cliente WCF (**ServicioNumerosArabesClient**) que facilita la comunicación entre el backend y el cliente web. Cuando el usuario ingresa un número y solicita la conversión, el cliente WCF se conecta al servicio, envía el número y recibe una lista de conversiones que luego se presentan en la página web utilizando un acordeón para una visualización clara y dinámica.

4.3. Frontend:

El frontend fue desarrollado con ASP.NET Web Forms, y sus 2 componentes principales son:

- ✓ **Entrada de número a convertir:** El usuario ingresa el número a convertir a través de un campo de texto en la página web como se puede ver en la figura 4.1. Este campo es un control de tipo `TextBox` de ASP.NET Web Forms definido como `txtNumero`. Cuando el usuario introduce el número y hace clic en el botón de conversión, el sistema valida la entrada y llama al servicio web correspondiente para realizar la conversión y muestra los resultados.

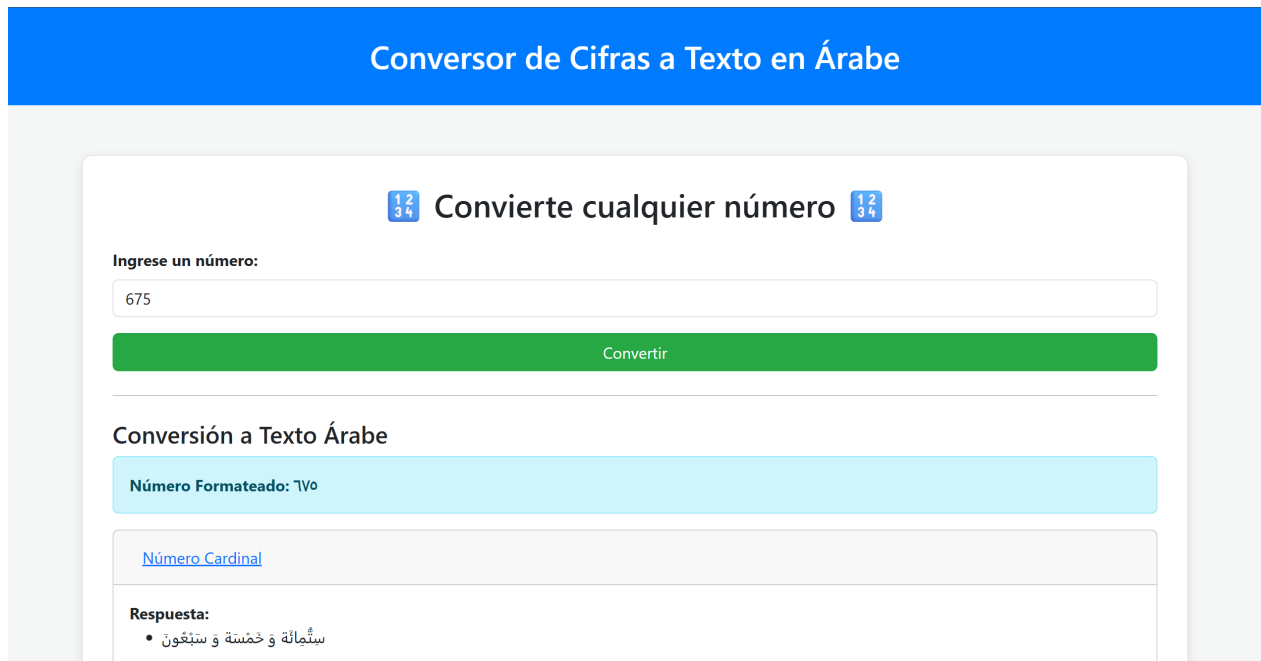


Figure 4.1: Vista de la interfaz gráfica de conversión

- ✓ **Acordeón para el resultado:** El acordeón [14] es un control que permite ocultar o mostrar contenido dentro de paneles expandibles. En este caso, se utiliza para mostrar las conversiones de manera organizada. Cuando el usuario solicita la conversión, el backend genera el HTML necesario para mostrar las conversiones en un acordeón. Este acordeón contiene varias secciones (cada una representando una conversión: Cardinal, Ordinal, Romano, etc.) que pueden expandirse o colapsarse al hacer clic sobre ellas. Los resultados se organizan en estos apartados, para el número 21, por ejemplo:

- **Número formateado:** Primero, el número se muestra en su forma oriental, como se ve a continuación en la figura 4.2:

Número formateado: ٢١

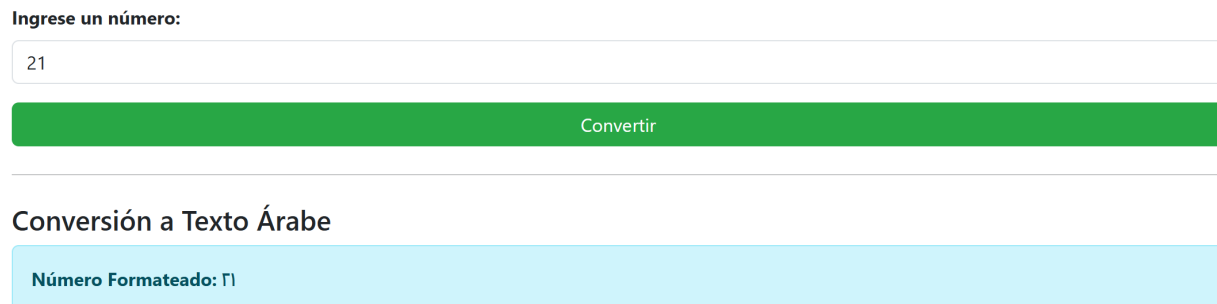


Figure 4.2: Vista: numero formateado

- **Conversión:** Después, se muestran las conversiones posibles para ese número. Por

ejemplo:

- Cardinal: واحد و عَشْرُونَ ("veintiuno").
 - Ordinal: الأول و العَشْرُونَ ("veintiún").
 - Romano: XXI.
 - Multiplicativo: واحد و عَشْرُونَ مَرَّات ("veintiuna veces").
- Ejemplos: También se proporcionan frases que emplean el número para cada tipo de conversión, como por ejemplo:
 - الرقم واحد و عَشْرُونَ يجلب الحظ السعيد. ("El número 21 trae buena suerte").
 - جاء الطالب في المرتبة الأول و العَشْرُونَ. ("El estudiante llegó en el puesto veintiuno").
 - Notas: Finalmente, se incluyen notas de interés que proporcionan más detalles sobre cómo funcionan estos números en árabe.

Esta es la vista de la conversión y cómo el sistema organiza la información en un acordeón interactivo. Si un tipo de conversión no tiene una forma disponible, simplemente no se muestra en el acordeón.

en la figura 4.3, se puede ver la vista del acordeón.

Conversión a Texto Árabe

Número Formateado: ٢١

Número Cardinal

Respuesta:

- واحد و عَشْرُونَ

Ejemplos:

- الرقم واحد و عَشْرُونَ يجلب الحظ السعيد.
- واحد و عَشْرُونَ هو رقم جميل جدًا.
- لقد كتبت واحد و عَشْرُونَ في الخانة العلوية.
- كم يورو لديك في حسابك؟ لدي واحد و عَشْرُونَ.
- كلفني واحد و عَشْرُونَ جنيهًا في إنجلترا.
- لدي واحد و عَشْرُونَ دولارًا مدخرًا.

Notas:

- En árabe, los números deben concordar en género con el sustantivo que acompañan. Por ejemplo, se usa 'اثنان' con palabras femeninas y 'اثنان' con palabras masculinas.
- El número 1 tiene formas especiales para mil y millón: 'ألف' (millón) y 'مليون' (millón), etc. Solo se dice 'ألف' no 'واحد ألف'.
- El número 2 tiene formas especiales para mil y millón: se dice 'ألفان' y 'مليونان', no 'اثنان ألف' o 'اثنان مليون'.
- A partir de 3 hasta 10, los sustantivos contados toman la forma plural, por ejemplo: 'ثلاثة آلاف' (tres miles), 'عشرة ملايين' (diez millones).
- El número 100 tiene formas especiales como 'مائة', 'مائتان' y combinaciones como 'ثلاثمائة'.
- Los números grandes como millones y miles son considerados sustantivos y no adjetivos, y se comportan como tales en la oración.

Figure 4.3: Vista de acordeón de resultados

4.3.1. Vistas de las conversiones

A continuación, en las figuras 4.4,4.5,4.6,4.7,4.8 se puede ver cómo se ve cada tipo de conversión:

Número Cardinal
Opciones y Notas

Respuesta:

- مائة و ثلاثة و عشرون

Ejemplos:

- لقد كتبت مائة و ثلاثة و عشرون في الحقل العلوي
- عدد الطلاب في الصف هو مائة و ثلاثة و عشرون
- هل يمكنني الحصول على مائة و ثلاثة و عشرون من هذا الرقم؟
- لقد رأيت مائة و ثلاثة و عشرون رقما في المستند
- رقم الهاتف الذي أبحث عنه هو مائة و ثلاثة و عشرون

Figure 4.4: Vista: conversión cardinal

Si se pulsa el botón "Opciones y Notas" que se ve en la figura 4.4, se añaden las opciones y notas a la vista, como se puede ver en la figura 4.5

Número Cardinal
Opciones y Notas

Respuesta:

- مائة و ثلاثة و عشرون

Ejemplos:

- لقد كتبت مائة و ثلاثة و عشرون في الحقل العلوي.
- عدد الطلاب في الصف هو مائة و ثلاثة و عشرون.
- هل يمكنني الحصول على مائة و ثلاثة و عشرون من هذا الرقم؟
- لقد رأيت مائة و ثلاثة و عشرون رقما في المستند.
- رقم الهاتف الذي أبحث عنه هو مائة و ثلاثة و عشرون.

Opciones:

- Femenino: مائة و ثلاث و عشرون

Notas:

- En árabe, los números deben concordar en género con el sustantivo que acompañan. Por ejemplo, se usa 'اثنان' con palabras femeninas y 'اثنان' con palabras masculinas.
- El número 'واحد' y 'واحدة' no se antepone cuando el número forma parte de mil (ألف), millón (مليون), etc. Solo se dice 'ألف', no 'واحد ألف'.
- El número 2 tiene formas especiales para mil y millón: se dice 'ألفان' y 'مليونان', no 'اثنان ألف' o 'اثنان مليون'.
- A partir de 3 hasta 10, los sustantivos contados toman la forma plural, por ejemplo: 'ثلاثة آلاف' (tres miles), 'عشرة' (diez millones).
- El número 100 tiene formas especiales como 'مائتان', 'مائة' y combinaciones como 'ثلاثمائة'.
- Los números grandes como millones y miles son considerados sustantivos y no adjetivos, y se comportan como tales en la oración.

Figure 4.5: Vista: conversión cardinal con opciones y notas

Número Ordinal
Opciones y Notas

Respuesta:

- المئة و الثالث و العشرون

Ejemplos:

- جاء الطالب في المرتبة المئة و الثالث و العشرون.
- حصلت على المركز المئة و الثالث و العشرون في المسابقة.
- كان هذا هو اليوم المئة و الثالث و العشرون من الرحلة.
- في المحاولة المئة و الثالث و العشرون نجحت أخيرًا.
- الكتاب المئة و الثالث و العشرون في السلسلة هو الأفضل.
- في المباراة المئة و الثالث و العشرون، سجل هدفًا رائعًا.

Figure 4.6: Vista: conversión ordinal

[Número Romano](#) [Opciones y Notas](#)

Respuesta:

- CXXIII

Ejemplos:

- CXXIII تم تأسيس المدينة في العام.
- كان معروفًا بشجاعته CXXIII الملك هنري.
- من الكتاب هذا العام CXXIII نُشرت الطبعة.
- حدثت تغييرات جذرية، CXXIII في القرن.
- من الفيلم CXXIII هذه هي النسخة.
- تم اتخاذ قرارات مهمة CXXIII في المؤتمر.

Figure 4.7: Vista: conversión a romano

[Número multiplicativo](#) [Opciones y Notas](#)

Respuesta:

- مائة و ثلاثون مَرَّات

Ejemplos:

- قام بمضاعفة الجهد مائة و ثلاثون مَرَّات .
- تكررت الحادثة مائة و ثلاثون مَرَّات .
- تضخم السعر بمعدل مائة و ثلاثون مَرَّات .
- تم النسخ مائة و ثلاثون مَرَّات .
- أُعيد التنفيذ مائة و ثلاثون مَرَّات خلال المشروع .

Figure 4.8: Vista: conversión multiplicativa

5. Desarrollo

5.1. Metodología

Este proyecto se ha desarrollado siguiendo la metodología en cascada [15], una de las metodologías más tradicionales y estructuradas para el desarrollo de software. En este enfoque, cada fase del proyecto depende de la entrega exitosa de la fase anterior, lo que permite un desarrollo secuencial y bien organizado. A continuación, se detallan las fases y tareas que se han seguido a lo largo del trabajo, y las decisiones tomadas durante el proceso.

5.1.1. Fases de desarrollo

Las fases del desarrollo del proyecto se dividen en cuatro partes principales: Cada fase ha sido diseñada para llevar a cabo tareas específicas.

Fase 1: Estudio previo / Análisis (40 horas)

Durante este tiempo, se realizaron varias tareas que permitieron identificar los requisitos necesarios para el desarrollo del servicio de conversión de números en árabe.

✓ **Tarea 1.1: Documentación y herramientas**

Se investigaron y estudiaron las tecnologías y las documentaciones correspondientes a las herramientas seleccionadas para iniciar el desarrollo del proyecto, así como la investigación de las fuentes necesarias para entender la conversión de números en árabe. Además, se recopiló la documentación relevante para comprender mejor las características de las herramientas y asegurarse de que fueran adecuadas para las necesidades del proyecto.

✓ **Tarea 1.2: Análisis de requisitos de usuario**

En esta tarea, se definieron las expectativas del usuario final. Se estudiaron los diferentes tipos de conversiones de números en árabe, como cardinales, ordinales, romanos y orientales, y se aseguraron de que las funcionalidades solicitadas estuvieran cubiertas adecuadamente.

✓ Tarea 1.3: Análisis de requisitos de software

Aquí, se analizaron las especificaciones técnicas y los requisitos del sistema, asegurándose de que el sistema tuviera la arquitectura correcta para que se pueda integrar en otros proyectos sin comprometer el funcionamiento.

Fase 2: Diseño / Desarrollo / Implementación / Despliegue (230 horas)

Con los requisitos bien establecidos, comenzó la fase de diseño y desarrollo.

✓ Tarea 2.1: Implementación del servicio web

Se implementó el servicio web, que se explicó en apartados anteriores.

✓ Tarea 2.2: Implementación y diseño de la aplicación

Se implementó la página web, que se explicó en apartados anteriores.

Fase 3: Evaluación / Validación / Prueba (20 horas)

En esta fase, se verificó que el sistema estuviera funcionando correctamente, realizando pruebas de todos los aspectos clave del sistema, incluyendo la conversión de números y el manejo de distintos formatos.

✓ Tarea 3: Pruebas de la aplicación

Se realizaron pruebas exhaustivas en la aplicación, cubriendo todos los casos posibles, como la conversión de diferentes tipos de números (positivos, negativos, con decimales, con monedas), comprobando las respuestas a los distintos tipos de entrada.

Fase 4: Documentación / Presentación (10 horas)

Finalmente, en esta fase, se elaboró la documentación técnica y se preparó la presentación del proyecto.

✓ Tarea 4.1: Elaboración de la documentación

Se preparó la memoria que explica todo el proceso de diseño y implementación del sistema,

✓ Tarea 4.2: Presentación

Se preparó la presentación final, que explica todo el proceso de desarrollo, desde el análisis inicial hasta la implementación final.

5.1.2. Control de versiones

Para facilitar el desarrollo y gestión del proyecto, se ha utilizado control de versiones [16]. Esto ha permitido mantener un registro detallado de todas las modificaciones realizadas, así

como la capacidad de revertir a versiones anteriores y verificar el historial de cambios cuando sea necesario.

El control de versiones es una práctica fundamental en el desarrollo de software que permite a los equipos gestionar y rastrear los cambios realizados en el código fuente de un proyecto a lo largo del tiempo. Este proceso no solo ayuda a mantener un registro detallado de las modificaciones, sino que también facilita la colaboración entre varios desarrolladores, permitiéndoles trabajar simultáneamente en diferentes partes del código sin interferir entre sí.

A través de sistemas de control de versiones, como Git, los desarrolladores pueden acceder a un historial completo de los cambios realizados, lo que les permite comparar diferentes versiones del código, identificar errores y restaurar versiones anteriores en caso de que se cometan fallos. Esto es crucial, ya que minimiza el riesgo de pérdida de trabajo y facilita la resolución de problemas cuando surgen conflictos o errores inesperados.

Además, el control de versiones también optimiza el proceso de integración continua y entrega continua (CI/CD), que son prácticas comunes en entornos de DevOps. Los sistemas de control de versiones permiten a los equipos de desarrollo realizar implementaciones más rápidas y seguras, al asegurarse de que cada cambio realizado en el código se pueda rastrear, probar y revisar antes de ser integrado al proyecto final. Esto mejora la eficiencia del desarrollo, reduce los tiempos de prueba y garantiza que las implementaciones sean más exitosas.

Otro aspecto clave del control de versiones es la protección del código fuente, que, en muchos casos, representa el conocimiento acumulado del equipo sobre el dominio del proyecto. Al proporcionar un sistema organizado para gestionar las versiones del código, se previene la pérdida de información valiosa debido a errores humanos o problemas imprevistos, y se asegura de que el código se mantenga accesible y respaldado a lo largo del tiempo.

6. Conclusiones y trabajos futuros

6.1. Conclusiones

Tras el desarrollo del proyecto y el análisis de los resultados obtenidos, se pueden extraer las siguientes conclusiones:

- ✓ La implementación del sistema de conversión de números a texto en árabe ha sido exitosa, permitiendo la conversión de números enteros, decimales, ordinales, romanos y multiplicativos en árabe, lo que facilita la utilización del sistema en diversas aplicaciones que requieren manipulación de números en este idioma.
- ✓ La solución desarrollada demuestra que la conversión numérica a texto, aunque aparentemente sencilla, implica un análisis detallado de las reglas gramaticales y lingüísticas del árabe, las cuales deben ser respetadas para garantizar la precisión de los resultados.
- ✓ La conversión de números a texto en árabe es especialmente relevante para su uso en áreas como la educación, la contabilidad y otros sistemas que manejan grandes volúmenes de datos numéricos.

6.2. Trabajos futuros

Con base en los avances y logros obtenidos en este proyecto, se proponen los siguientes trabajos futuros:

- ✓ **Extensión a otros idiomas:** El sistema de conversión podría ser ampliado para soportar otros idiomas, como francés, alemán o incluso chino, para una aplicación más global del proyecto.
- ✓ **Implementación de nuevos formatos numéricos:** El sistema podría ser extendido para permitir la conversión a otros sistemas numéricos, o a nuevos formatos como el fraccionario.
- ✓ **Integración con sistemas externos:** Integrar este sistema de conversión con plataformas existentes de gestión de datos.

Bibliography

- [1] AccountingW. *Number to Words Conversion*. Accedido: abril 2025. 2024. URL: <https://www.accountingw.com/2024/10/Number-to-Words-Conversion.html>.
- [2] numbers.un. *Numbers Converter*. Accedido: abril 2025. 2025. URL: <https://numbers.un-web.com/>.
- [3] ChatGPT. *ChatGPT*. Accedido: abril 2025. 2025. URL: <https://chatgpt.com/>.
- [4] DeepSeek. *DeepSeek*. Accedido: abril 2025. 2025. URL: <https://chat.deepseek.com/>.
- [5] Microsoft Copilot. *Microsoft Copilot*. Accedido: abril 2025. 2025. URL: <https://copilot.cloud.microsoft/>.
- [6] AM Salinas Copo. *Libro de Programación Orientada a Objetos*. Accedido: junio 2025. 2025. URL: <https://repositorio.cidecuador.org/bitstream/123456789/3182/3/LIBRO%20DE%20PROGRAMACI%C3%93N%20ORIENTADA%20A%20OBJETOS.pdf>.
- [7] Juan Gabriel Gomila. *Así es Visual Studio, el IDE Favorito de los Desarrolladores*. Accedido: junio 2025. 2024. URL: <https://cursos.frogamesformacion.com/pages/blog/visual-studio>.
- [8] Microsoft. *Visual Studio*. Accedido: junio 2025. 2019. URL: <https://visualstudio.microsoft.com/>.
- [9] Microsoft. *WCF Overview*. Accedido: junio 2025. 2016. URL: <https://learn.microsoft.com/es-es/dotnet/framework/wcf/whats-wcf>.
- [10] Right People Group. *Qué es .NET y por qué las empresas lo utilizan frente a otros marcos de trabajo*. Accedido: junio 2025. 2025. URL: <https://rightpeoplegroup.com/es/blog/que-es-net-y-por-que-las-empresas-lo-utilizan-frente-a-otros-marcos-de-trabajo>.
- [11] Microsoft. *ASP.NET Overview*. Accedido: junio 2025. 2021. URL: <https://learn.microsoft.com/es-es/aspnet/overview>.
- [12] certidevs. *curso-c-sharp*. Accedido: junio 2025. 2025. URL: <https://certidevs.com/curso-c-sharp>.
- [13] Programando a Medianoche. *Introducción a Task Parallel Library (TPL)*. Accedido: junio 2025. 2009. URL: <https://www.programandoamedianoche.com/2009/10/introduccion-a-task-parallel-library-tpl/>.
- [14] Microsoft. *usar el control extensor del panel contraíble de AJAX de ASP.NET*. Accedido: junio 2025. 2025. URL: <https://learn.microsoft.com/es-es/aspnet/web-forms/videos/ajax-control-toolkit/how-do-i-use-the-aspnet-ajax-collapsible-panel-extender>.

- [15] Strider Staff. *El modelo Waterfall en los proyectos de software: comprender esta metodología*. Accedido: junio 2025. 2024. URL: <https://www.onstrider.com/es/blog/metodologia-cascata-waterfall>.
- [16] Atlassian. *¿Qué es el control de versiones?* Accedido: junio 2025. 2025. URL: <https://www.atlassian.com/es/git/tutorials/what-is-version-control>.