

## Codificador - Decodificador

Generado por Doxygen 1.8.10

Viernes, 2 de Octubre de 2015 15:01:27



# Índice general

<b>1</b>	<b>Índice de namespaces</b>	<b>1</b>
1.1	Paquetes . . . . .	1
<b>2</b>	<b>Índice jerárquico</b>	<b>3</b>
2.1	Jerarquía de la clase . . . . .	3
<b>3</b>	<b>Índice de clases</b>	<b>5</b>
3.1	Lista de clases . . . . .	5
<b>4</b>	<b>Índice de archivos</b>	<b>7</b>
4.1	Lista de archivos . . . . .	7
<b>5</b>	<b>Documentación de namespaces</b>	<b>9</b>
5.1	Referencia del Namespace CodDecod . . . . .	9
<b>6</b>	<b>Documentación de las clases</b>	<b>11</b>
6.1	Referencia de la Clase CodDecod.CodificarDecodificar . . . . .	11
6.1.1	Descripción detallada . . . . .	12
6.1.2	Documentación del constructor y destructor . . . . .	12
6.1.2.1	CodificarDecodificar() . . . . .	12
6.1.3	Documentación de las funciones miembro . . . . .	12
6.1.3.1	button1_Click(object sender, EventArgs e) . . . . .	12
6.1.3.2	button2_Click(object sender, EventArgs e) . . . . .	13
6.1.3.3	codascii_Click(object sender, EventArgs e) . . . . .	14
6.1.3.4	codce_Click(object sender, EventArgs e) . . . . .	14
6.1.3.5	codmor_Click(object sender, EventArgs e) . . . . .	15
6.1.3.6	decodascii_Click(object sender, EventArgs e) . . . . .	16
6.1.3.7	decodce_Click(object sender, EventArgs e) . . . . .	17
6.1.3.8	decodmor_Click(object sender, EventArgs e) . . . . .	17
6.1.3.9	getPosABC(char character) . . . . .	18
6.1.3.10	salir_Click(object sender, EventArgs e) . . . . .	18
<b>7</b>	<b>Documentación de archivos</b>	<b>19</b>
7.1	Referencia del Archivo CodDecod/CodificarDecodificar.cs . . . . .	19

7.1.1	Descripción detallada . . . . .	19
<b>Índice</b>		<b>21</b>

# Capítulo 1

## Índice de namespaces

### 1.1. Paquetes

Aquí van los paquetes con una breve descripción (si está disponible):

<b>CodDecod</b> . . . . .	9
---------------------------	---



## Capítulo 2

# Índice jerárquico

### 2.1. Jerarquía de la clase

Esta lista de herencias esta ordenada aproximadamente por orden alfabético:

Form

CodDecod.CodificarDecodificar . . . . . 11





## Capítulo 3

# Índice de clases

### 3.1. Lista de clases

Lista de las clases, estructuras, uniones e interfaces con una breve descripción:

<b>CodDecod.CodificarDecodificar</b> . . . . .	11
--	----



## Capítulo 4

# Indice de archivos

### 4.1. Lista de archivos

Lista de todos los archivos con descripciones breves:

CodDecod/ <b>CodificarDecodificar.cs</b>	
Menú con distintos tipos de codificación y decodificación . . . . .	19



## Capítulo 5

# Documentación de namespaces

### 5.1. Referencia del Namespace CodDecod

#### Clases

- class **CodificarDecodificar**

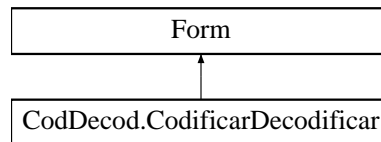


## Capítulo 6

# Documentación de las clases

### 6.1. Referencia de la Clase CodDecod.CodificarDecodificar

Diagrama de herencias de CodDecod.CodificarDecodificar



#### Métodos públicos

- **CodificarDecodificar ()**

#### Métodos privados

- void **button2\_Click** (object sender, EventArgs e)  
*Codificación para escícala.*
- void **button1\_Click** (object sender, EventArgs e)  
*Decodificación para escícala.*
- void **codascii\_Click** (object sender, EventArgs e)  
*Codificación ASCII.*
- void **decodascii\_Click** (object sender, EventArgs e)  
*Decodificación ASCII.*
- void **codmor\_Click** (object sender, EventArgs e)  
*Codificación morse.*
- void **decodmor\_Click** (object sender, EventArgs e)  
*Decodificación morse.*
- void **codce\_Click** (object sender, EventArgs e)  
*Codificación César.*
- void **decodce\_Click** (object sender, EventArgs e)  
*Decodificación César.*
- void **salir\_Click** (object sender, EventArgs e)  
*Salir de la aplicación.*

## Métodos privados estáticos

- static int **getPosABC** (char caracter)

*Se recorre el string que definimos con los valores para cifrar.*

### 6.1.1. Descripción detallada

Definición en la línea 18 del archivo CodificarDecodificar.cs.

### 6.1.2. Documentación del constructor y destructor

#### 6.1.2.1. CodDecod.CodificarDecodificar.CodificarDecodificar ( )

Definición en la línea 20 del archivo CodificarDecodificar.cs.

### 6.1.3. Documentación de las funciones miembro

#### 6.1.3.1. void CodDecod.CodificarDecodificar.button1\_Click ( object sender, EventArgs e ) [private]

Decodificación para escítala.

Abrimos el archivo de texto con la codificacion escítala

Parámetros

<i>open.Show↵ Dialog()</i>	Abrir un archivo de texto con OpenFileDialog
<i>unicodeString</i>	Guardamos el archivo en una cadena de texto

Mostramos un mensaje con la cadena del texto capturado

Pedimos el número original de caras para que el descifrado sea correcto

Parámetros

<i>caras</i>	Variable tipo entero, para guardar la primer dimensión de la escítala
--------------	---

Determinamos automaticamente la segunda dimensión de la escítala

Parámetros

<i>larcad</i>	Variable tipo entero, para determinar el largo de la cadena
<i>fi1</i>	Didivimos el largo de la cadena entre la primera dimensión del usuario

Creamos la matriz para descifrar la escítala

Parámetros

<i>char[,]mat1</i>	Matriz tipo caracter
<i>k</i>	Variable tipo entero, para recorrer la cadena de texto

Llenamos la matriz

Impresión de la matriz decodificada

Parámetros

<i>c</i>	Variable tipo string, para concatenar la matriz
----------	---

Guardamos la decodificación en un archivo de texto

Definición en la línea 198 del archivo CodificarDecodificar.cs.



6.1.3.2. void CodDecod.CodificarDecodificar.button2\_Click ( object sender, EventArgs e ) [private]

Codificación para escítala.

Abrir archivo de texto

Parámetros

<i>open.ShowDialog()</i>	Abrir un archivo de texto con OpenFileDialog
<i>unicodeString</i>	Guardamos el archivo en una cadena de texto

Mostramos un mensaje con la cadena del texto capturado

Parámetros

<i>caras</i>	Variable tipo entero, para guardar la primer dimensión de la escítala
--------------	---

Calcularemos la dimension minima que puede insertar el usuario

Parámetros

<i>larcad</i>	Variable tipo entero, para determinar el largo de la cadena
<i>fi1</i>	Dividimos el largo de la cadena entre la primera dimensión del usuario
<i>modd</i>	Variable tipo entero, para determinar el módulo del largo de la cadena entre la primera dimensión del usuario

Aumentamos una fila si el mod de el largo de la cadena y la primera dimension es mayor a 0

Le pedimos al usuario la segunda dimensión y le mostramos la dimension minima recomendada

Parámetros

<i>res1</i>	Dimension 2 ingresada por el usuario
-------------	--------------------------------------

Impresion de error si la segunda dimension es menor a la recomendada

Realizamos operaciones para determinar cuantos espacios sobran en la matriz

Parámetros

<i>mul</i>	Variable tipo entero, multiplicando las dos dimensiones para obtener el numero de celdas de la matriz
<i>res2</i>	Variable de tipo entero, que guarda el número de espacios en blanco que quedaran en la matriz

Llenamos los espacios con un espacio en blanco

Parámetros

<i>spa</i>	Variable tipo string, la cual nos guardara el número de espacios que bebemos agregarle
------------	--

Agregamos los espacios a nuestra cadena sumando la variable spa

Creamos la matriz para insertar la cadena de texto

Parámetros

<i>char[,]mat1</i>	Matriz tipo caracter
<i>k</i>	Variable tipo entero, para recorrer la cadena de texto

Llenamos la matriz con la cadena de texto

Imprimimos el arreglo

Parámetros

<i>c</i>	Variable tipo string, para concatenar la matriz
----------	---

Concatenamos la codificación escítala

**Parámetros**

<i>e1</i>	Variable tipo string, para concatenar
-----------	---------------------------------------

Mostramos la escítala en cuadro de dialogo

Guardamos la codificación en un archivo de texto

Definición en la línea 28 del archivo CodificarDecodificar.cs.

**6.1.3.3. void CodDecod.CodificarDecodificar.codascii\_Click ( object sender, EventArgs e ) [private]**

Codificación ASCII.

Creamos un nuevo objeto de la clase ASCIIEncoding

**Parámetros**

<i>rutas</i>	Variable tipo string en la cual se guardara los datos del archivo
--------------	---

Guardamos el archivo en un string

**Parámetros**

<i>string</i>	unicodeString Variable tipo string que nos permite guardar el archivo
---------------	---

Codificamos la cadena a ASCII

**Parámetros**

<i>Byte[] encoded↔ Bytes</i>	Matriz del tipo Byte donde se guarda la codificación
----------------------------------	--

Impresion de la codificación

**Parámetros**

<i>String</i>	c Variable tipo String nos sirve para concaquetar cadenas de texto
<i>Byte</i>	b Variable tipo Byte que nos sirve para recorrer una matriz tipo Byte uno por uno

Guardamos la codificación en un archivo de texto

**Parámetros**

<i>string</i>	guardar Variable que nos permite guardar en un archivo
---------------	--

Definición en la línea 289 del archivo CodificarDecodificar.cs.

**6.1.3.4. void CodDecod.CodificarDecodificar.codce\_Click ( object sender, EventArgs e ) [private]**

Codificación César.

Agregamos en una cadena los valores que va a detectar la codificación

**Parámetros**

<i>abc</i>	es una variable de tipo String
------------	--------------------------------

Abrimos un archivo de texto

**Parámetros**

<i>open.Show↔ Dialog()</i>	Abrir un archivo de texto con OpenFileDialog
--------------------------------	--

Lo guardamos en una cadena

Le pedimos al usuario que ingrese el desplazamiento

**Parámetros**

<i>desplazamiento</i>	es una variable de tipo int.
<i>cifrado</i>	es una variable de tipo String.

Recorremos el string para cifrarlo

Verificamos que el caracter existe en nuestro string abc definido

**Parámetros**

<i>posCaracter</i>	es una variable de tipo Int.
--------------------	------------------------------

SUMAMOS el desplazamiento al caracter

**Parámetros**

<i>pos</i>	es una variable de tipo Int.
------------	------------------------------

Asignamos el caracter segun el desplazamiento(cifrado)

Concatenamos el cifrado

Si no existe el caracter se pasa al string cifrado de manera igual

Mostramos el mensaje cifrado

Guardamos el mensaje de cifrado en un archivo de texto

Definición en la línea 720 del archivo CodificarDecodificar.cs.

**6.1.3.5. void CodDecod.CodificarDecodificar.codmor\_Click ( object sender, EventArgs e ) [private]**

Codificación morse.

Creamos un diccionario del tipo caracter-cadena

Asignamos a cada letra y numero un valor

Abrimos un archivo que contenga un texto

**Parámetros**

<i>open.ShowDialog()</i>	Abrir un archivo de texto con OpenFileDialog
--------------------------	--

Guardamos el archivo de texto en una cadena

**Parámetros**

<i>unicodeString</i>	Guardamos el archivo en una cadena de texto
----------------------	---

Convertimos la cadena a mayúsculas

Ciclo para eliminar el de nuestra cadena de texto

**Parámetros**

<i>j</i>	es una variable de tipo entero: contador para recorrer la cadena de texto
----------	---

Quitamos los return de la cadena de texto

se compara la cadena de texto con nuestro diccionario y a la vez sustituye el texto por el codigo morse

**Parámetros**

<i>codmorse</i>	es una variable de tipo String.
-----------------	---------------------------------

Obtenemos el largo de la cadena

**Parámetros**

<i>t</i>	es una variable de tipo int.
----------	------------------------------

Creamos nueva matriz para guardar la codificación

#### Parámetros

<i>string[]</i>	armorse es una variable de tipo String:
-----------------	---

For para llenar la matriz y codificar la cadena de texto

leemos la cadena en la posición i

Convertimos el caracter obtenido en el valor que definimos en el diccionario

Guardamos la codificación en la matriz y un string para imprimirlo

Impresión de la codificación

Guardamos la codificación en un archivo de texto

Definición en la línea 459 del archivo CodificarDecodificar.cs.

6.1.3.6. void CodDecod.CodificarDecodificar.decodascii\_Click ( object sender, EventArgs e ) [private]

Decodificación ASCII.

Creamos un nuevo objeto de la clase ASCIIEncoding

Abrimos un archivo con codificación ASCII

#### Parámetros

<i> rutas</i>	Variable tipo string en la cual se guardara los datos del archivo
---------------	---

Guardamos el archivo en un string

Creamos una matriz de tipo entero

#### Parámetros

<i>int[] matascii</i>	Una matriz de tipo entero que sera igual al tamaño de los datos del archivo
<i>b</i>	Variable de tipo string que nos sirve para guardar un caracter y convertirlo
<i>j</i>	Variable de tipo entero la cual nos sirve para recorrer la matriz

Llenaremos la matriz convirtiendo la cadena en enteros

Omitimos los espacios en blanco

Imprimimos la codificación

#### Parámetros

<i>inm</i>	Variable de tipo string que nos permite guardar los datos de la matriz
------------	--

Creamos una matriz de tipo byte

#### Parámetros

<i>Byte[] m1</i>	Matriz de tipo Byte donde sera igual a al tamaño de la matriz anterior
------------------	--

Llenamos la matriz tipo byte con la matriz tipo entero

#### Parámetros

<i>k</i>	Variable de tipo entero que nos permite recorrer
----------	--

Decodificamos la matriz

Guardamos la decodificación en un archivo de texto

#### Parámetros

<i>guardar</i>	Variable de tipo string que nos permite guardar nuestro texto decodificado en un archivo
----------------	--

Definición en la línea 346 del archivo CodificarDecodificar.cs.

6.1.3.7. `void CodDecod.CodificarDecodificar.decodce_Click ( object sender, EventArgs e ) [private]`

Decodificación César.

Agregamos en una cadena los valores que va a detectar la codificación

Parámetros

<i>abc</i>	es una variable de tipo String.
------------	---------------------------------

Abrimos un archivo de texto

Parámetros

<i>open.ShowDialog()</i>	Abrir un archivo de texto con OpenFileDialog
--------------------------	--

Lo guardamos en una cadena

Parámetros

<i>unicodeString</i>	es una variable de tipo String.
----------------------	---------------------------------

Le pedimos al usuario que ingrese el desplazamiento ORIGINAL

Parámetros

<i>desplazamiento</i>	es una variable de tipo Int.
-----------------------	------------------------------

Recorremos el string para descifrarlo

Verificamos que el caracter existe en nuestro string abc definido

RESTAMOS el desplazamiento al caracter

Concatenamos el cifrado

Si no existe el caracter se pasa al string descifrado de manera igual

Mostramos el mensaje descifrado

Guardamos el mensaje descifrado en un archivo de texto

Definición en la línea 843 del archivo CodificarDecodificar.cs.

6.1.3.8. `void CodDecod.CodificarDecodificar.decodmor_Click ( object sender, EventArgs e ) [private]`

Decodificación morse.

Creación de diccionario de datos de tipo string - char

Abrimos archivo con codificación morse

Parámetros

<i>open.ShowDialog()</i>	Abrir un archivo de texto con OpenFileDialog
--------------------------	--

Guardamos el archivo de texto en una cadena de texto

Parámetros

<i>unicodeString</i>	Guardamos el archivo en una cadena de texto
----------------------	---

Obtenemos el largo de la cadena de texto

**Parámetros**

<i>t</i>	es una variable de tipo entero: obtenemos el largo de la cadena de texto
----------	--

Definimos una matriz para guardar la cadena

Con este ciclo quitamos los espacios contenidos en la cadena

Guardamos el valor obtenido en la posición del arreglo indicado

Utilizamos este ciclo para verificar si existen null en la cadena

Decodificaremos la cadena

**Parámetros**

<i>decodmorse</i>	es una variable de tipo String.
-------------------	---------------------------------

Comparamos la posición del arreglo con nuestro diccionario

**Parámetros**

<i>b</i>	es una variable de tipo String.
----------	---------------------------------

Guardamos el valor decodificado en un string

Mostramos el texto decodificado

Con este algoritmo agregaremos el (return) para que detecte los saltos de linea

Guardamos el texto decodificado en un archivo de texto

Definición en la línea 585 del archivo CodificarDecodificar.cs.

**6.1.3.9. static int CodDecod.CodificarDecodificar.getPosABC ( char *character* ) [static],[private]**

Se recorre el string que definimos con los valores para cifrar.

Determinamos la posición del caracter para cifrar

**Devuelve**

i Regresa la posición obtenida del caracter detectado

-1 Regresamos este valor para indicar que el caracter no existe en nuestro diccionario

Definición en la línea 818 del archivo CodificarDecodificar.cs.

**6.1.3.10. void CodDecod.CodificarDecodificar.salir\_Click ( object *sender*, EventArgs *e* ) [private]**

Salir de la aplicación.

Definición en la línea 931 del archivo CodificarDecodificar.cs.

La documentación para esta clase fue generada a partir del siguiente fichero:

- CodDecod/**CodificarDecodificar.cs**

## Capítulo 7

# Documentación de archivos

### 7.1. Referencia del Archivo CodDecod/CodificarDecodificar.cs

Menú con distintos tipos de codificación y decodificación.

#### Clases

- class **CodDecod.CodificarDecodificar**

#### Namespaces

- namespace **CodDecod**

#### 7.1.1. Descripción detallada

Menú con distintos tipos de codificación y decodificación.

#### Versión

1.0





# Índice alfabético

- button1\_Click
  - CodDecod::CodificarDecodificar, 12
- button2\_Click
  - CodDecod::CodificarDecodificar, 12
- CodDecod, 9
- CodDecod.CodificarDecodificar, 11
- CodDecod/CodificarDecodificar.cs, 19
- CodDecod::CodificarDecodificar
  - button1\_Click, 12
  - button2\_Click, 12
  - codascii\_Click, 14
  - codce\_Click, 14
  - CodificarDecodificar, 12
  - codmor\_Click, 15
  - decodascii\_Click, 16
  - decodce\_Click, 17
  - decodmor\_Click, 17
  - getPosABC, 18
  - salir\_Click, 18
- codascii\_Click
  - CodDecod::CodificarDecodificar, 14
- codce\_Click
  - CodDecod::CodificarDecodificar, 14
- CodificarDecodificar
  - CodDecod::CodificarDecodificar, 12
- codmor\_Click
  - CodDecod::CodificarDecodificar, 15
- decodascii\_Click
  - CodDecod::CodificarDecodificar, 16
- decodce\_Click
  - CodDecod::CodificarDecodificar, 17
- decodmor\_Click
  - CodDecod::CodificarDecodificar, 17
- getPosABC
  - CodDecod::CodificarDecodificar, 18
- salir\_Click
  - CodDecod::CodificarDecodificar, 18