

Manual de referencia

RRRGzip

Generado por Doxygen 1.6.3

Fri Jun 11 22:04:09 2010

Índice general

1. Índice de clases	1
1.1. Lista de clases	1
2. Documentación de las clases	3
2.1. Referencia de la Estructura crc32	3
2.1.1. Descripción detallada	3
2.2. Referencia de la Clase Deflate	4
2.2.1. Descripción detallada	5
2.2.2. Documentación de las funciones miembro	5
2.2.2.1. AddUC	5
2.2.2.2. GetTexto	5
2.2.2.3. GetVC	5
2.2.2.4. GetVDComprimido	6
2.2.2.5. Huffman	6
2.2.2.6. SizeComprimido	6
2.2.2.7. SizeTexto	6
2.3. Referencia de la Estructura info	7
2.3.1. Descripción detallada	7
2.4. Referencia de la Clase LZ77	8
2.4.1. Descripción detallada	9
2.4.2. Documentación del constructor y destructor	9
2.4.2.1. LZ77	9
2.4.3. Documentación de las funciones miembro	9
2.4.3.1. AddUC	9
2.4.3.2. GetLZ	9
2.4.3.3. GetVC	10
2.4.3.4. SizeVC	10
2.5. Referencia de la Clase Tabla	11

2.5.1.	Descripción detallada	11
2.5.2.	Documentación del constructor y destructor	11
2.5.2.1.	Tabla	11
2.5.3.	Documentación de las funciones miembro	12
2.5.3.1.	GetCodigo	12
2.5.3.2.	GetVB	12
2.6.	Referencia de la Clase Vbits	13
2.6.1.	Descripción detallada	14
2.6.2.	Documentación del constructor y destructor	14
2.6.2.1.	Vbits	14
2.6.3.	Documentación de las funciones miembro	14
2.6.3.1.	Get	14
2.6.3.2.	GetVC	15
2.6.3.3.	GetVD	15
2.6.3.4.	Insertar	15
2.6.3.5.	operator!=	15
2.6.3.6.	operator+=	15
2.6.3.7.	operator+=	15
2.6.3.8.	operator=	16
2.6.3.9.	operator==	16
2.6.3.10.	Set	16
2.6.3.11.	SizeVC	16
2.6.4.	Documentación de las funciones relacionadas y clases amigas	16
2.6.4.1.	operator<<	16
2.6.4.2.	operator>>	17
2.7.	Referencia de la Clase VD_char	18
2.7.1.	Descripción detallada	19
2.7.2.	Documentación del constructor y destructor	19
2.7.2.1.	VD_char	19
2.7.2.2.	VD_char	19
2.7.3.	Documentación de las funciones miembro	19
2.7.3.1.	Add	19
2.7.3.2.	Get	20
2.7.3.3.	GetVD	20
2.7.3.4.	operator=	20
2.7.3.5.	Set	20

2.7.3.6. Size	20
2.7.4. Documentación de las funciones relacionadas y clases amigas	20
2.7.4.1. operator<<	20
2.7.4.2. operator>>	21
2.8. Referencia de la Clase VD_uint	22
2.8.1. Descripción detallada	22
2.8.2. Documentación del constructor y destructor	23
2.8.2.1. VD_uint	23
2.8.3. Documentación de las funciones miembro	23
2.8.3.1. Add	23
2.8.3.2. Get	23
2.8.3.3. operator=	23
2.8.3.4. Set	24
2.8.3.5. Size	24
2.8.4. Documentación de las funciones relacionadas y clases amigas	24
2.8.4.1. operator<<	24
2.8.4.2. operator>>	24

Capítulo 1

Índice de clases

1.1. Lista de clases

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

crc32 (Esta clase gestiona una tabla de 256 unsigned long y un unsigned long que sera el codigo crc32 de un texto dado)	3
Deflate (Esta clase gestiona una clase LZ77 y un vector de bits, que será el archivo comprimido)	4
info (Esta estructura contiene un entero y un Vbits asociado)	7
LZ77 (Esta clase gestiona un vector de dinamico de caracteres y lo codifica en un vector dinamico de unsigned ints)	8
Tabla (Esta clase gestiona un vector de infos)	11
Vbits (Esta clase gestiona un vector de bits)	13
VD_char (Esta clase gestiona un vector dinamico de unsigned chars)	18
VD_uint (Esta clase gestiona un vector dinamico de unsigned ints)	22

Capítulo 2

Documentación de las clases

2.1. Referencia de la Estructura `crc32`

esta clase gestiona una tabla de 256 unsigned long y un unsigned long que sera el codigo `crc32` de un texto dado

```
#include <crc32.h>
```

Atributos públicos

- unsigned long `tabla` [256]
- unsigned long `crc`

2.1.1. Descripción detallada

esta clase gestiona una tabla de 256 unsigned long y un unsigned long que sera el codigo `crc32` de un texto dado

Autor

Roberto Jose del Rio Rodriguez 75154244-L

Versión

Revision

0.1

Fecha

Date

11/06/2010

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

- `/home/roberto/Escritorio/MP2/compresor/final/programa/include/crc32.h`

2.2. Referencia de la Clase Deflate

esta clase gestiona una clase [LZ77](#) y un vector de bits, que será el archivo comprimido

```
#include <deflate.h>
```

Métodos públicos

- [Deflate](#) ()
Constructor por defecto.
- [~Deflate](#) ()
Destructor por defecto.
- void [LeerTexto](#) ()
Funcion que lee un vector de caracteres desde el teclado y lo almacena en lz.
- void [AddUC](#) (const unsigned char c)
Funcion que añade el caracter c al vector vc de LZ.
- [Vbits Huffman](#) (unsigned int ui) const
Funcion que devuelve el [Vbits](#) de Huffman asociado al unsigned int ui.
- void [Comprimir](#) ()
Funcion que comprime un archivo almacenado en lz y lo guarda en ziped.
- [Vbits GetComprimido](#) () const
Funcion que devuelve el [Vbits](#) ziped, miembro de la clase.
- unsigned char * [GetVDComprimido](#) () const
Funcion que devuelve el vector de unsigned char, miembro de la clase [VD_char](#), miembro de la clase [Vbits](#), asociado a ziped.
- [VD_char GetVC](#) () const
Funcion que devuelve el [VD_char](#), miembro de la clase [Vbits](#), asociado a ziped.
- [VD_char GetTexto](#) () const
Funcion que devuelve el [VD_char](#), miembro de [LZ77](#), asociado a lz.
- unsigned int [SizeTexto](#) ()
Funcion que devuelve el tamaño del [VD_char](#), miembro de la clase [Vbits](#), asociado a lz.
- unsigned int [SizeComprimido](#) () const
Funcion que devuelve el tamaño en bits de ziped.

2.2.1. Descripción detallada

esta clase gestiona una clase [LZ77](#) y un vector de bits, que será el archivo comprimido

Autor

Roberto Jose del Rio Rodriguez 75154244-L

Versión**Revision**

0.1

Fecha**Date**

11/06/2010

2.2.2. Documentación de las funciones miembro

2.2.2.1. void Deflate::AddUC (const unsigned char *c*) [inline]

Funcion que añade el caracter *c* al vector *vc* de LZ.

Parámetros

c caracter a almacenar.

2.2.2.2. VD_char Deflate::GetTexto () const [inline]

Funcion que devuelve el [VD_char](#), miembro de [LZ77](#), asociado a lz.

Devuelve

[VD_char](#) asociado a lz.

2.2.2.3. VD_char Deflate::GetVC () const [inline]

Funcion que devuelve el [VD_char](#), miembro de la clase [Vbits](#), asociado a ziped.

Devuelve

[VD_char](#) asociado a ziped.

2.2.2.4. unsigned char* Deflate::GetVDComprimido () const [inline]

Funcion que devuelve el vector de unsigned char, miembro de la clase [VD_char](#), miembro de la clase [Vbits](#), asociado a ziped.

Devuelve

vector de unsigned char asociado a ziped.

2.2.2.5. Vbits Deflate::Huffman (unsigned int ui) const

Funcion que devuelve el [Vbits](#) de Huffman asociado al unsigned int ui.

Parámetros

ui numero del cual queremos saber su [Vbits](#).

2.2.2.6. unsigned int Deflate::SizeComprimido () const

Funcion que devuelve el tamaño en bits de ziped.

Devuelve

tamaño en bits de ziped.

2.2.2.7. unsigned int Deflate::SizeTexto () [inline]

Funcion que devuelve el tamaño del [VD_char](#), miembro de la clase [Vbits](#), asociado a lz.

Devuelve

tamaño del [VD_char](#) asociado a lz.

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

- /home/roberto/Escritorio/MP2/compresor/final/programa/include/deflate.h

2.3. Referencia de la Estructura info

esta estructura contiene un entero y un [Vbits](#) asociado.

```
#include <tabla.h>
```

Atributos públicos

- unsigned int **codigo**
- [Vbits](#) **vb**

2.3.1. Descripción detallada

esta estructura contiene un entero y un [Vbits](#) asociado.

Autor

Roberto Jose del Rio Rodriguez 75154244-L

Versión

Revision

0.1

Fecha

Date

11/06/2010

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

- /home/roberto/Escritorio/MP2/compresor/final/programa/include/tabla.h

2.4. Referencia de la Clase LZ77

esta clase gestiona un vector de dinámico de caracteres y lo codifica en un vector dinámico de unsigned ints

```
#include <LZ77.h>
```

Métodos públicos

- [LZ77](#) ()
Constructor por defecto.
- [LZ77](#) (const [VD_char](#) &vdchar)
Constructor por parametros.
- [~LZ77](#) ()
Destructor por defecto.
- void [LeerVUI](#) ()
Funcion que lee un vector de unsigned ints por teclado y lo almacena en vi.
- void [EscribirVUI](#) () const
Funcion que saca por pantalla vi.
- unsigned int [GetLZ](#) (int i) const
Funcion que devuelve el elemento i-esimo del vector vi.
- unsigned int [SizeVC](#) ()
Funcion que devuelve el tamaño de vc.
- unsigned int [SizeUI](#) () const
Funcion que devuelve el tamaño de vi.
- void [LeerVUC](#) ()
Funcion que lee un vector de unsigned chars por teclado y lo almacena en vc.
- void [EscribirVUC](#) () const
Funcion que saca por pantalla vc.
- [VD_char](#) [GetVC](#) () const
Funcion que devuelve el el vector dinamico vc.
- void [AddUC](#) (const unsigned char c)
Funcion que añade el caracter "c" a vc.
- void [Codificar](#) ()
Funcion que codifica el texto del vector vc a codigo numerico, previo a aplicar la codificacion deflate.
- void [CodificarMejor](#) ()
Funcion que codifica el texto del vector vc a codigo numerico, previo a aplicar la codificacion deflate, pero con un algoritmo mas compresor.

2.4.1. Descripción detallada

esta clase gestiona un vector de dinámico de caracteres y lo codifica en un vector dinámico de unsigned ints

Autor

Roberto Jose del Rio Rodriguez 75154244-L

Versión**Revision**

0.1

Fecha**Date**

11/06/2010

2.4.2. Documentación del constructor y destructor

2.4.2.1. LZ77::LZ77 (const VD_char & vdchar)

Constructor por parametros.

Parámetros

← *vdchar* VD_char con el que inicializamos el LZ77.

2.4.3. Documentación de las funciones miembro

2.4.3.1. void LZ77::AddUC (const unsigned char *c*) [inline]

Funcion que añade el caracter "c" a vc.

Parámetros

c caracter a añadir al vector.

2.4.3.2. unsigned int LZ77::GetLZ (int *i*) const

Funcion que devuelve el elemento i-esimo del vector vi.

Parámetros

i numero del elemento a devolver.

2.4.3.3. VD_char LZ77::GetVC () const [inline]

Funcion que devuelve el el vector dinamico vc.

Devuelve

vector vc miembro.

2.4.3.4. unsigned int LZ77::SizeVC () [inline]

Funcion que devuelve el tamaño de vc.

Devuelve

tamaño de vc.

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

- /home/roberto/Escritorio/MP2/compresor/final/programa/include/LZ77.h

2.5. Referencia de la Clase Tabla

esta clase gestiona un vector de infos

```
#include <tabla.h>
```

Métodos públicos

- **Tabla** (unsigned int tipo)
Constructor por parametros.
- **~Tabla** ()
Destructor por defecto.
- unsigned int **GetCodigo** (unsigned int car) const
Esta funcion devuelve el codigo asociado a un entero dado.
- **Vbits GetVB** (unsigned int car) const
*Esta funcion devuelve el **Vbits** asociado a un entero dado.*
- void **EscribirTabla** () const
Esta funcion saca por pantalla la tabla actual.

2.5.1. Descripción detallada

esta clase gestiona un vector de infos

Autor

Roberto Jose del Rio Rodriguez 75154244-L

Versión

Revision

0.1

Fecha

Date

11/06/2010

2.5.2. Documentación del constructor y destructor

2.5.2.1. **Tabla::Tabla** (unsigned int *tipo*)

Constructor por parametros.

Parámetros

← *tipo* tipo de tabla que queremos crear.

2.5.3. Documentación de las funciones miembro**2.5.3.1. unsigned int Tabla::GetCodigo (unsigned int *car*) const**

Esta funcion devuelve el codigo asociado a un entero dado.

Parámetros

← *car* entero del que queremos saber el codigo asociado.

2.5.3.2. Vbits Tabla::GetVB (unsigned int *car*) const

Esta funcion devuelve el [Vbits](#) asociado a un entero dado.

Parámetros

← *car* entero del que queremos saber el [Vbits](#) asociado.

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

- /home/roberto/Escritorio/MP2/compresor/final/programa/include/tabla.h

2.6. Referencia de la Clase Vbits

esta clase gestiona un vector de bits

```
#include <vbits.h>
```

Métodos públicos

- **Vbits** ()
Constructor por defecto.
- **Vbits** (int num, int l)
Constructor por parametros.
- **~Vbits** ()
Destructor por defecto.
- void **Insertar** (unsigned char bit)
Funcion que agrega un bit al final del vector.
- int **Size** () const
Funcion que devuelve el numero de bits almacenados.
- unsigned int **Get** (int i) const
Funcion que devuelve el bit i-esimo.
- **VD_char GetVC** () const
Funcion que devuelve el vector dinamico miembro de la clase.
- unsigned char * **GetVD** () const
Funcion que devuelve el vector de unsigned char asociado al miembro vd.
- int **SizeVC** () const
Funcion que devuelve el numero de bytes reservados del vector dinamico.
- void **Set** (int i, unsigned char bit)
Funcion que asignar el bit "bit" en la posicion i-esima.
- **Vbits & operator=** (const **Vbits** &V)
Operador de asignacion.
- void **operator+=** (unsigned char bit)
Funcion que agrega un bit al final del vector.
- void **operator+=** (const **Vbits** &vbit)
Funcion que agrega un Vbit al final del vector.
- bool **operator==** (const **Vbits** &vbit) const
Operador de igualdad.

- `bool operator!= (const Vbits &vbit) const`
Operador de desigualdad.

Amigas

- `ostream & operator<< (ostream &os, const Vbits &vb)`
Operador de salida.
- `istream & operator>> (istream &is, Vbits &vb)`
Operador de entrada.

2.6.1. Descripción detallada

esta clase gestiona un vector de bits

Autor

Roberto Jose del Rio Rodriguez 75154244-L

Versión

Revision

0.1

Fecha

Date

11/06/2010

2.6.2. Documentación del constructor y destructor

2.6.2.1. Vbits::Vbits (int num, int l)

Constructor por parametros.

Parámetros

- ← *num* numero que queremos codificar en binario.
- ← *l* numero de bits con el que queremos codificar el numero num.

2.6.3. Documentación de las funciones miembro

2.6.3.1. unsigned int Vbits::Get (int i) const

Funcion que devuelve el bit i-esimo.

Parámetros

← *i* numero del bit que queremos consultar.

2.6.3.2. VD_char Vbits::GetVC () const [inline]

Funcion que devuelve el vector dinamico miembro de la clase.

Devuelve

vector dinamico asociado.

2.6.3.3. unsigned char* Vbits::GetVD () const [inline]

Funcion que devuelve el vector de unsigned char asociado al miembro vd.

Devuelve

vector unsigned char asociado a vd.

2.6.3.4. void Vbits::Insertar (unsigned char *bit*)

Funcion que agrega un bit al final del vector.

Parámetros

← *bit* bit que queremos agregar.

2.6.3.5. bool Vbits::operator!= (const Vbits & *vbit*) const

Operador de desigualdad.

Parámetros

← *vbit* Vbits que queremos comparar con el actual.

2.6.3.6. void Vbits::operator+= (const Vbits & *vbit*)

Funcion que agrega un Vbit al final del vector.

Parámetros

← *vbit* Vbit que queremos agregar.

2.6.3.7. void Vbits::operator+= (unsigned char *bit*)

Funcion que agrega un bit al final del vector.

Parámetros

← *bit* bit que queremos agregar.

2.6.3.8. Vbits& Vbits::operator= (const Vbits & V)

Operador de asignacion.

Parámetros

← *V* **Vbits** que le pasamos para copiarlo.

2.6.3.9. bool Vbits::operator== (const Vbits & vbit) const

Operador de igualdad.

Parámetros

← *vbit* **Vbits** que queremos comparar con el actual.

2.6.3.10. void Vbits::Set (int i, unsigned char bit)

Funcion que asignar el bit "bit" en la posicion i-esima.

Parámetros

← *i* numero del bit que queremos asignar.

← *bit* bit a asignar.

2.6.3.11. int Vbits::SizeVC () const [inline]

Funcion que devuelve el numero de bytes reservados del vector dinamico.

Devuelve

bytes reservados en el vector dinamico.

2.6.4. Documentación de las funciones relacionadas y clases amigas**2.6.4.1. ostream& operator<< (ostream & os, const Vbits & vb) [friend]**

Operador de salida.

Parámetros

← *os* flujo de salida.

← *vb* **Vbits** a sacar por el flujo

2.6.4.2. istream& operator>> (istream & *is*, Vbits & *vb*) [friend]

Operador de entrada.

Parámetros

- ← *is* flujo de entrada.
- ← *vb* Vbits a meter por el flujo

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

- /home/roberto/Escritorio/MP2/compresor/final/programa/include/vbits.h

2.7. Referencia de la Clase VD_char

esta clase gestiona un vector dinamico de unsigned chars

```
#include <vdchar.h>
```

Métodos públicos

- [VD_char \(\)](#)
Constructor por defecto. Crea un nuevo vector de unsigned char sin memoria reservada, e inicializa los elementos miembros de la clase a 0.
- [~VD_char \(\)](#)
Destructor por defecto.
- [VD_char \(const \[VD_char\]\(#\) &V\)](#)
Constructor de copia.
- [VD_char \(int n\)](#)
Constructor por parametros.
- [VD_char & operator= \(const \[VD_char\]\(#\) &V\)](#)
Operador de asignacion.
- [int \[Size\]\(#\) \(\) const](#)
Funcion que devuelve el numero de unsigned ints usados.
- [unsigned char \[Get\]\(#\) \(const int &i\) const](#)
Funcion que devuelve el elemento i-esimo del vector.
- [unsigned char * \[GetVD\]\(#\) \(\) const](#)
Funcion que devuelve el vector de unsigned char miembro de la clase.
- [void \[Set\]\(#\) \(unsigned char c, int i\)](#)
Funcion que asigna el elemento c al lugar i-esimo del vector.
- [void \[Add\]\(#\) \(unsigned char c\)](#)
Funcion que añade al final del vector el elemento c.

Amigas

- [ostream & operator<< \(ostream &os, const \[VD_char\]\(#\) &vd\)](#)
Operador de salida.
- [istream & operator>> \(istream &is, \[VD_char\]\(#\) &vd\)](#)
Operador de entrada.

2.7.1. Descripción detallada

esta clase gestiona un vector dinamico de unsigned chars

Autor

Roberto Jose del Rio Rodriguez 75154244-L

Versión**Revision**

0.1

Fecha**Date**

11/06/2010

2.7.2. Documentación del constructor y destructor

2.7.2.1. VD_char::VD_char (const VD_char & V)

Constructor de copia.

Parámetros

← *V* [VD_char](#) que le pasamos para copiarlo.

2.7.2.2. VD_char::VD_char (int *n*)

Constructor por parametros.

Parámetros

← *n* numero de elementos con que inicializamos el vector.

2.7.3. Documentación de las funciones miembro

2.7.3.1. void VD_char::Add (unsigned char *c*)

Funcion que añade al final del vector el elemento *c*.

Parámetros

← *c* elemento que queremos añadir.

2.7.3.2. unsigned char VD_char::Get (const int & i) const

Funcion que devuelve el elemento i-esimo del vector.

Parámetros

← *i* numero del elemento que queremos obtener.

2.7.3.3. unsigned char* VD_char::GetVD () const [inline]

Funcion que devuelve el vector de unsigned char miembro de la clase.

Devuelve

el vector de unsigned chars miembro de la clase.

2.7.3.4. VD_char& VD_char::operator= (const VD_char & V)

Operador de asignacion.

Parámetros

← *V* [VD_char](#) que le pasamos para copiarlo.

2.7.3.5. void VD_char::Set (unsigned char c, int i)

Funcion que asigna el elemento c al lugar i-esimo del vector.

Parámetros

← *c* elemento que queremos asignar.

← *i* lugar del vector que queremos asignar.

2.7.3.6. int VD_char::Size () const [inline]

Funcion que devuelve el numero de unsigned ints usados.

Devuelve

el numero de unsigned chars usados en el vector.

2.7.4. Documentación de las funciones relacionadas y clases amigas**2.7.4.1. ostream& operator<< (ostream & os, const VD_char & vd) [friend]**

Operador de salida.

Parámetros

← *os* flujo de salida.

← *vd* [VD_char](#) a sacar por el flujo

2.7.4.2. istream& operator>> (istream & *is*, VD_char & *vd*) [friend]

Operador de entrada.

Parámetros

← *is* flujo de entrada.

← *vd* [VD_char](#) a meter por el flujo

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

- /home/roberto/Escritorio/MP2/compresor/final/programa/include/vdchar.h

2.8. Referencia de la Clase VD_uint

esta clase gestiona un vector dinamico de unsigned ints

```
#include <vduint.h>
```

Métodos públicos

- **VD_uint** ()
Constructor por defecto. Crea un nuevo vector de unsigned int sin memoria reservada, e inicializa los elementos miembros de la clase a 0.
- **~VD_uint** ()
Destructor por defecto.
- **VD_uint** (const **VD_uint** &V)
Constructor de copia.
- **VD_uint** & **operator=** (const **VD_uint** &V)
Operador de asignacion.
- int **Size** () const
Funcion que devuelve el numero de unsigned ints usados.
- unsigned int **Get** (const int &i) const
Funcion que devuelve el elemento i-esimo del vector.
- void **Set** (unsigned int c, int i)
Funcion que asigna el elemento c al lugar i-esimo del vector.
- void **Add** (unsigned int c)
Funcion que añade al final del vector el elemento c.

Amigas

- ostream & **operator<<** (ostream &os, const **VD_uint** &vd)
Operador de salida.
- istream & **operator>>** (istream &is, **VD_uint** &vd)
Operador de entrada.

2.8.1. Descripción detallada

esta clase gestiona un vector dinamico de unsigned ints

Autor

Roberto Jose del Rio Rodriguez 75154244-L

Versión**Revision**

0.1

Fecha**Date**

11/06/2010

2.8.2. Documentación del constructor y destructor**2.8.2.1. VD_uint::VD_uint (const VD_uint & V)**

Constructor de copia.

Parámetros

← *V* [VD_uint](#) que le pasamos para copiarlo.

2.8.3. Documentación de las funciones miembro**2.8.3.1. void VD_uint::Add (unsigned int c)**

Funcion que añade al final del vector el elemento *c*.

Parámetros

← *c* elemento que queremos añadir.

2.8.3.2. unsigned int VD_uint::Get (const int & i) const

Funcion que devuelve el elemento *i*-esimo del vector.

Parámetros

← *i* numero del elemento que queremos obtener.

2.8.3.3. VD_uint& VD_uint::operator= (const VD_uint & V)

Operador de asignacion.

Parámetros

← *V* [VD_uint](#) que le pasamos para copiarlo.

2.8.3.4. void VD_uint::Set (unsigned int *c*, int *i*)

Funcion que asigna el elemento *c* al lugar *i*-esimo del vector.

Parámetros

- ← *c* elemento que queremos asignar.
- ← *i* lugar del vector que queremos asignar.

2.8.3.5. int VD_uint::Size () const [inline]

Funcion que devuelve el numero de unsigned ints usados.

Devuelve

el numero de unsigned ints usados en el vector.

2.8.4. Documentación de las funciones relacionadas y clases amigas**2.8.4.1. ostream& operator<< (ostream & *os*, const VD_uint & *vd*) [friend]**

Operador de salida.

Parámetros

- ← *os* flujo de salida.
- ← *vd* [VD_uint](#) a sacar por el flujo

2.8.4.2. istream& operator>> (istream & *is*, VD_uint & *vd*) [friend]

Operador de entrada.

Parámetros

- ← *is* flujo de entrada.
- ← *vd* [VD_uint](#) a meter por el flujo

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

- /home/roberto/Escritorio/MP2/compresor/final/programa/include/vduint.h

Índice alfabético

Add
 VD_char, 19
 VD_uint, 23
AddUC
 Deflate, 5
 LZ77, 9
crc32, 3
Deflate, 4
 AddUC, 5
 GetTexto, 5
 GetVC, 5
 GetVDComprimido, 5
 Huffman, 6
 SizeComprimido, 6
 SizeTexto, 6
Get
 Vbits, 14
 VD_char, 19
 VD_uint, 23
GetCodigo
 Tabla, 12
GetLZ
 LZ77, 9
GetTexto
 Deflate, 5
GetVB
 Tabla, 12
GetVC
 Deflate, 5
 LZ77, 9
 Vbits, 15
GetVD
 Vbits, 15
 VD_char, 20
GetVDComprimido
 Deflate, 5
Huffman
 Deflate, 6
info, 7
Insertar
 Vbits, 15
LZ77, 8
 AddUC, 9
 GetLZ, 9
 GetVC, 9
 LZ77, 9
 SizeVC, 10
operator<<
 Vbits, 16
 VD_char, 20
 VD_uint, 24
operator>>
 Vbits, 16
 VD_char, 20
 VD_uint, 24
operator+=
 Vbits, 15
operator=
 Vbits, 15
 VD_char, 20
 VD_uint, 23
operator==
 Vbits, 16
Set
 Vbits, 16
 VD_char, 20
 VD_uint, 23
Size
 VD_char, 20
 VD_uint, 24
SizeComprimido
 Deflate, 6
SizeTexto
 Deflate, 6
SizeVC
 LZ77, 10
 Vbits, 16
Tabla, 11
 GetCodigo, 12
 GetVB, 12
 Tabla, 11

Vbits, [13](#)
 Get, [14](#)
 GetVC, [15](#)
 GetVD, [15](#)
 Insertar, [15](#)
 operator<<, [16](#)
 operator>>, [16](#)
 operator+&, [15](#)
 operator=, [15](#)
 operator==, [16](#)
 Set, [16](#)
 SizeVC, [16](#)
 Vbits, [14](#)
VD_char, [18](#)
 Add, [19](#)
 Get, [19](#)
 GetVD, [20](#)
 operator<<, [20](#)
 operator>>, [20](#)
 operator=, [20](#)
 Set, [20](#)
 Size, [20](#)
 VD_char, [19](#)
 VD_char, [19](#)
VD_uint, [22](#)
 Add, [23](#)
 Get, [23](#)
 operator<<, [24](#)
 operator>>, [24](#)
 operator=, [23](#)
 Set, [23](#)
 Size, [24](#)
 VD_uint, [23](#)
 VD_uint, [23](#)