

## TP Organización de Datos

Generado por Doxygen 1.7.6.1

Lunes, 3 de Junio de 2013 01:13:20



# Índice general

<b>1</b>	<b>Índice de clases</b>	<b>1</b>
1.1	Jerarquía de la clase . . . . .	1
<b>2</b>	<b>Índice de clases</b>	<b>3</b>
2.1	Lista de clases . . . . .	3
<b>3</b>	<b>Documentación de las clases</b>	<b>5</b>
3.1	Referencia de la Estructura ManejadorRegistrosVariables::_Header . . .	5
3.2	Referencia de la Clase Anio . . . . .	5
3.3	Referencia de la Clase ArbolBMas . . . . .	6
3.4	Referencia de la Clase ArchivoListas . . . . .	7
3.5	Referencia de la Clase ArchivoRegistrosLongitudFija . . . . .	7
3.6	Referencia de la Clase Bloque . . . . .	8
3.7	Referencia de la Clase Clave . . . . .	9
3.8	Referencia de la Clase ClaveNumerica . . . . .	10
3.9	Referencia de la Clase ClaveString . . . . .	11
3.10	Referencia de la Clase ClaveX . . . . .	13
3.11	Referencia de la Clase Compresor . . . . .	14
3.12	Referencia de la Clase CompresorPPMC . . . . .	15
3.13	Referencia de la Clase HashingExtensible . . . . .	16
3.14	Referencia de la Estructura header_arbol . . . . .	16
3.15	Referencia de la Estructura header_arlf . . . . .	16
3.16	Referencia de la Estructura ManejadorRegistrosVariables::Header- RegistroLibre . . . . .	17
3.17	Referencia de la Clase Heap . . . . .	17
3.18	Referencia de la Clase Idioma . . . . .	17

3.19 Referencia de la Clase Indexador . . . . .	17
3.20 Referencia de la Clase IndiceInvertido . . . . .	18
3.21 Referencia de la Clase IndiceInvertidoPorAutor . . . . .	19
3.21.1 Documentación del constructor y destructor . . . . .	19
3.21.1.1 IndiceInvertidoPorAutor . . . . .	19
3.22 Referencia de la Clase IndiceInvertidoPorTitulo . . . . .	19
3.23 Referencia de la Clase IterArbolBMas . . . . .	20
3.24 Referencia de la Clase IterNodoSecuencial . . . . .	20
3.25 Referencia de la plantilla de la Clase ManejadorArchivoDatosBasicos< TipoDato > . . . . .	20
3.26 Referencia de la Clase ManejadorArchivos . . . . .	22
3.27 Referencia de la Clase ManejadorArchivosTexto . . . . .	23
3.28 Referencia de la Clase ManejadorBloques . . . . .	24
3.29 Referencia de la Clase ManejadorRegistrosVariables . . . . .	25
3.30 Referencia de la Estructura mb_header . . . . .	27
3.31 Referencia de la Estructura mdb_header . . . . .	27
3.32 Referencia de la Clase NodoArbol . . . . .	28
3.33 Referencia de la Clase NodoInterno . . . . .	29
3.34 Referencia de la Clase NodoSecuencial . . . . .	31
3.35 Referencia de la Clase Normalizador . . . . .	32
3.36 Referencia de la Clase ParserCanciones . . . . .	33
3.37 Referencia de la Clase RegistroCancion . . . . .	33
3.38 Referencia de la Clase RegistroClave . . . . .	35
3.39 Referencia de la Clase RegistroFijo . . . . .	37
3.40 Referencia de la Clase RegistroVariable . . . . .	38
3.41 Referencia de la Clase ResolvedorConsultas . . . . .	39
3.42 Referencia de la Clase SortExterno . . . . .	39
3.43 Referencia de la Clase Tabla . . . . .	40
3.44 Referencia de la Clase Texto . . . . .	40

# Capítulo 1

## Índice de clases

### 1.1. Jerarquía de la clase

Esta lista de herencias esta ordenada aproximadamente por orden alfabético:

ManejadorRegistrosVariables::_Header . . . . .	5
Anio . . . . .	5
ArbolBMas . . . . .	6
ArchivoListas . . . . .	7
ArchivoRegistrosLongitudFija . . . . .	7
Bloque . . . . .	8
Clave . . . . .	9
ClaveNumerica . . . . .	10
ClaveString . . . . .	11
ClaveX . . . . .	13
Compresor . . . . .	14
CompresorPPMC . . . . .	15
HashingExtensible . . . . .	16
header_arbol . . . . .	16
header_arlf . . . . .	16
ManejadorRegistrosVariables::HeaderRegistroLibre . . . . .	17
Heap . . . . .	17
Idioma . . . . .	17
Indexador . . . . .	17
IndiceInvertido . . . . .	18
IndiceInvertidoPorAutor . . . . .	19
IndiceInvertidoPorTitulo . . . . .	19
IterArbolBMas . . . . .	20
IterNodoSecuencial . . . . .	20
ManejadorArchivos . . . . .	22
ManejadorArchivoDatosBasicos< TipoDato > . . . . .	20
ManejadorArchivosTexto . . . . .	23
ManejadorBloques . . . . .	24

ManejadorRegistrosVariables . . . . .	25
mb_header . . . . .	27
mdb_header . . . . .	27
NodoArbol . . . . .	28
NodoInterno . . . . .	29
NodoSecuencial . . . . .	31
Normalizador . . . . .	32
ParserCanciones . . . . .	33
RegistroFijo . . . . .	37
RegistroVariable . . . . .	38
RegistroCancion . . . . .	33
RegistroClave . . . . .	35
ResolvedorConsultas . . . . .	39
SortExterno . . . . .	39
Tabla . . . . .	40
Texto . . . . .	40

## Capítulo 2

# Índice de clases

### 2.1. Lista de clases

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

ManejadorRegistrosVariables::_Header	5
Anio	5
ArbolBMas	6
ArchivoListas	7
ArchivoRegistrosLongitudFija	7
Bloque	8
Clave	9
ClaveNumerica	10
ClaveString	11
ClaveX	13
Compresor	14
CompresorPPMC	15
HashingExtensible	16
header_arbol	16
header_arlf	16
ManejadorRegistrosVariables::HeaderRegistroLibre	17
Heap	17
Idioma	17
Indexador	17
IndiceInvertido	18
IndiceInvertidoPorAutor	19
IndiceInvertidoPorTitulo	19
IterArbolBMas	20
IterNodoSecuencial	20
ManejadorArchivoDatosBasicos< TipoDato >	20
ManejadorArchivos	22
ManejadorArchivosTexto	23
ManejadorBloques	24
ManejadorRegistrosVariables	25

mb_header	27
mdb_header	27
NodoArbol	28
NodoInterno	29
NodoSecuencial	31
Normalizador	32
ParserCanciones	33
RegistroCancion	33
RegistroClave	35
RegistroFijo	37
RegistroVariable	38
ResolvedorConsultas	39
SortExterno	39
Tabla	40
Texto	40



## Capítulo 3

# Documentación de las clases

### 3.1. Referencia de la Estructura `ManejadorRegistrosVariables::` - Header

#### Atributos públicos

- long **offsetPrimerRegistroLibre**
- unsigned int **tamanoArchivo**
- unsigned int **cantidadRegistros**
- unsigned int **cantidadRegistrosLibres**

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

- `ManejadorRegistrosVariables.h`

### 3.2. Referencia de la Clase `Anio`

#### Métodos públicos

- bool **validar** (string anio)
- bool **cargar** (string anioS)
- int **get\_anio** ()
- void **resetear** ()

#### Atributos públicos estáticos

- static int const **ANIO\_DEFAULT** = -1

### Atributos protegidos

- `int anio`

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- `Anio.h`
- `Anio.cpp`

## 3.3. Referencia de la Clase ArbolBMas

### Tipos públicos

- `typedef NodoInterno::TipoHijo TipoHijo`

### Métodos públicos

- **ArbolBMas** (const [ArbolBMas](#) &otro)
- `int crear` (std::string fileName, unsigned int tamBloque=BLOQUE\_TAM\_DEFAULT)
- `int eliminar` (std::string fileName)
- `int abrir` (std::string fileName, std::string mode)
- `int cerrar` ()
- `int agregar` ([RegistroClave](#) &reg)
- `int quitar` ([RegistroClave](#) &reg)
- `int buscar` ([RegistroClave](#) &reg)
- `int _buscar` ([RegistroClave](#) &reg, unsigned int &numBloque)
- `bool esta_vacio` ()
- `unsigned int get_cant_minima_nodo` ()
- `unsigned int get_cant_maxima_nodo` ()
- `int _set_header` ()
- `int _get_header` ()
- `int _obtener_nodo_secuencial` (int numNodoSecuencial, [NodoSecuencial](#) &nodoSecuencialActual)
- `int _hallar_hijo_correspondiente` ([RegistroClave](#) \*registro, [NodoInterno](#) \*nodo-Buscar, TipoHijo &numeroBloqueHijoCorrespondiente)
- `int _insertar_recurso` (unsigned int &numeroBloqueActual, [RegistroClave](#) \*registro, TipoHijo &hijoPromocionado, [ClaveX](#) \*clavePromocionada)
- `int _hallar_hoja` ([RegistroClave](#) \*registro, unsigned int &numeroBloque)
- `int _insertar_recurso_hoja` ([Bloque](#) \*bloqueActual, [RegistroClave](#) \*registro, - TipoHijo &hijoPromocionado)
- `int _split_hoja` ([NodoSecuencial](#) \*nodoActual, vector< [RegistroClave](#) > \*registrosOverflow, TipoHijo &hijoPromocionado, [ClaveX](#) \*clavePromocionada)
- `int _split_interno` ([NodoInterno](#) \*nodo, [ClaveX](#) \*clavePromocionada, TipoHijo &hijoPromocionado)
- `void imprimir` ()
- `void obtener_primer_nodo_secuencial` (TipoHijo &numeroPrimerNodo)

### Atributos públicos estáticos

- static const int **FACTOR\_CARGA** = 50

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- ArbolBMas.h
- ArbolBMas.cpp

## 3.4. Referencia de la Clase ArchivoListas

### Métodos públicos

- virtual int **crear** (std::string directorioSalida, std::string fileNamee)
- virtual int **abrir** (std::string directorioSalidaa, std::string fileNamee)
- virtual short **agregar** ([RegistroVariable](#) \*listaDeCanciones)
- virtual int **devolver** ([RegistroVariable](#) \*listaDeCanciones, unsigned short pos\_lista)
- virtual int **reconstruir\_listas** (unsigned short \*ref\_listas, unsigned short cant\_ref, int ID)
- virtual int **reconstruir\_listas** (unsigned short ref\_lista, [RegistroVariable](#) &lista-Modificada)
- virtual int **eliminar** (std::string directorioSalida, std::string fileNamee)
- virtual int **get\_cantidad\_listas** ()

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- ArchivoListas.h
- ArchivoListas.cpp

## 3.5. Referencia de la Clase ArchivoRegistrosLongitudFija

### Métodos públicos

- int **create** (std::string dir, std::string fileName, int recSize=REC\_SIZE, int max-Recs=MAX\_RECS)
- int **open** (std::string dir, std::string fileName, std::string mode)
- int **destroy** (std::string dir, std::string fileName)
- int **close** ()
- int **seek** (int numRec)
- int **read** (int numRec, [RegistroFijo](#) &rec)
- int **append** (const [RegistroFijo](#) &rec)
- int **write** (const int numRec, const [RegistroFijo](#) &rec)
- int **readNext** ([RegistroFijo](#) &rec)

- **int updateRecord** (int numRec, const [RegistroFijo](#) &rec)
- **int deleteRecord** (int numRec)
- **int getMaxRecs** ()
- **int getRecordSize** ()
- **int getFirstEmptyRecordNumber** ()

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- ArchivoRegistrosLongitudFija.h
- ArchivoRegistrosLongitudFija.cpp

### 3.6. Referencia de la Clase Bloque

#### Métodos públicos

- **Bloque** (unsigned int tamBloque=BLOQUE\_TAM\_DEFAULT)
- **Bloque** (const [Bloque](#) &otro)
- virtual unsigned short **get\_cantidad\_registros\_almacenados** () const throw ()
- virtual unsigned int **get\_espacio\_libre** () const throw ()
- unsigned int **get\_tamano\_bloque** () const throw ()
- virtual char \* **get\_buffer** () const throw ()
- virtual bool **esta\_vacio** () const throw ()
- virtual int **recuperar\_registro** ([RegistroVariable](#) \*registro, unsigned short numeroRegistro) const throw ()
- virtual int **eliminar\_registro** (unsigned short numeroRegistro) throw ()
- virtual int **agregar\_registro** ([RegistroVariable](#) \*registro) throw ()
- virtual void **desempaquetar** (const char \*datos) throw ()
- virtual void **empaquetar** (char \*copia) const throw ()
- virtual bool **fue\_eliminado** ()
- virtual int **actualizar\_ref\_prox\_bloque** (unsigned int primerBloque)
- virtual int **obtener\_ref\_prox\_bloque** ()
- virtual void **\_listar\_registros** () throw ()
- virtual unsigned int **\_calcular\_espacio\_usable** () const throw ()
- virtual unsigned int **\_calcular\_espacio\_ocupado** () const throw ()
- virtual unsigned int **\_calcular\_espacio\_libre** () const throw ()

#### Métodos protegidos

- virtual void **\_obtener\_espacio\_libre** () const throw ()
- virtual void **\_escribir\_espacio\_libre** () throw ()
- virtual unsigned int **\_obtener\_offset\_final** () const throw ()
- virtual unsigned int **\_obtener\_offset\_registro** (unsigned short numeroRegistro) const throw ()
- virtual void **\_limpiar\_buffer** () throw ()
- virtual int **\_agregar\_registro** (char \*dato, unsigned short tamanoDato) throw ()

### Atributos protegidos

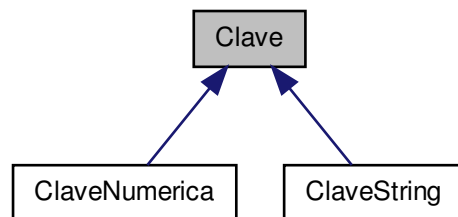
- char \* **bufferBloque**
- unsigned int **espacioLibre**
- unsigned int **tamanoBloque**
- unsigned int **espacioLibreOffset**

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- Bloque.h
- Bloque.cpp

## 3.7. Referencia de la Clase Clave

Diagrama de herencias de Clave



### Métodos públicos

- virtual void **operator=** (const Clave &)=0
- virtual bool **operator<** (const Clave &)=0
- virtual bool **operator>** (const Clave &)=0
- virtual bool **operator==** (const Clave &)=0
- virtual bool **operator!=** (const Clave &)=0
- virtual bool **operator<=** (const Clave &)=0
- virtual bool **operator>=** (const Clave &)=0
- virtual int **empaquetar** (char \*buffer)=0
- virtual void **desempaquetar** (char \*buffer, unsigned short tamanoBuffer)=0
- virtual int **get\_tamano** ()=0
- virtual void **resetear** ()=0

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- Clave.h
- Clave.cpp

### 3.8. Referencia de la Clase ClaveNumerica

Diagrama de herencias de ClaveNumerica

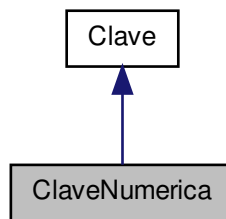
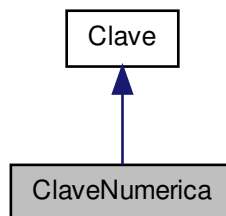


Diagrama de colaboración para ClaveNumerica:



#### Métodos públicos

- virtual bool **operator**< (const Clave &)
- virtual bool **operator**> (const Clave &)
- virtual bool **operator**== (const Clave &)
- virtual bool **operator**>= (const Clave &)
- virtual bool **operator**<= (const Clave &)

- virtual bool **operator!=** (const Clave &)
- virtual void **operator=** (const Clave &)
- virtual ostream & **operator<<** (ostream &os) const
- virtual void **desempaquetar** (char \*buffer, unsigned short tamanoBuffer=sizeof(int))
- virtual int **empaquetar** (char \*buffer)
- int **incrementar** ()
- void **set\_dato** (int dato)
- int **get\_dato** ()
- virtual void **imprimir\_dato** () const
- int **get\_tamano** ()
- void **resetear** ()
- **ClaveNumerica** (int dato)

#### Atributos protegidos

- int **dato**

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- ClaveNumerica.h
- ClaveNumerica.cpp

### 3.9. Referencia de la Clase ClaveString

Diagrama de herencias de ClaveString

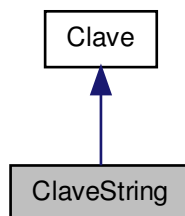
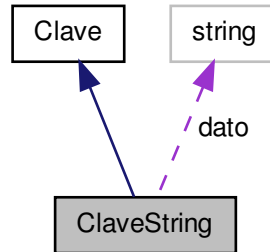


Diagrama de colaboración para ClaveString:



### Métodos públicos

- virtual void **desempaquetar** (char \*buffer, unsigned short tamanoBuffer=0)
- virtual int **empaquetar** (char \*buffer)
- virtual bool **operator>** (const Clave &otraClave)
- virtual bool **operator==** (const Clave &otraClave)
- virtual bool **operator<** (const Clave &otraClave)
- virtual bool **operator>=** (const Clave &otraClave)
- virtual bool **operator<=** (const Clave &otraClave)
- virtual bool **operator!=** (const Clave &otraClave)
- virtual void **operator=** (const Clave &otraClave)
- virtual void **operator=** (const string &unString)
- virtual std::ostream & **operator<<** (std::ostream &os) const
- void **remover\_espacios** ()
- virtual void **imprimir\_dato** () const
- void **set\_dato** (const string &)
- string **get\_dato** ()
- int **get\_tamano** ()
- void **resetear** ()
- **ClaveString** (const string &linea)
- **ClaveString** (const ClaveString &copia)

### Métodos públicos estáticos

- static string **remover\_espacios** (const std::string linea)



### Atributos públicos estáticos

- static const char **DATO\_DEFECTO** = '~'

### Atributos protegidos

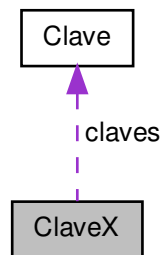
- string **dato**

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- ClaveString.h
- ClaveString.cpp

## 3.10. Referencia de la Clase ClaveX

Diagrama de colaboración para ClaveX:



### Métodos públicos

- **ClaveX** (const [ClaveX](#) &clavex)
- [ClaveX](#) & **operator=** (const [ClaveX](#) &otro)
- virtual int **empaquetar** (char \*buffer)
- virtual void **desempaquetar** (char \*buffer, unsigned short tamañoBuffer)
- virtual TipoClave **get\_tipo\_clave** () const
- virtual void **imprimir\_dato** () const
- virtual void **set\_clave** (string clave)
- virtual void **set\_clave** (int clave)
- virtual int **get\_tamaño\_empaquetado** () const
- virtual int **get\_tamaño\_clave** () const

- virtual int **get\_clave** (int &clave) const
- virtual int **get\_clave** (string &clave) const
- virtual bool **operator<** (const [ClaveX](#) &clavex) const
- virtual std::ostream & **operator<<** (std::ostream &os) const
- virtual bool **operator>** (const [ClaveX](#) &clavex) const
- virtual bool **operator>=** (const [ClaveX](#) &clavex) const
- virtual bool **operator<=** (const [ClaveX](#) &clavex) const
- virtual bool **operator==** (const [ClaveX](#) &clavex) const
- virtual bool **operator!=** (const [ClaveX](#) &clavex) const

### Métodos protegidos

- void **\_resetear\_datos** ()

### Atributos protegidos

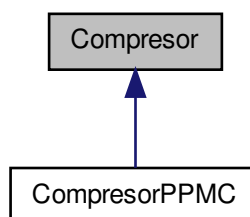
- TipoClave **tipoClave**
- [Clave](#) \* **claves** [CANTIDAD\_TIPOS\_CLAVE]

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- ClaveX.h
- ClaveX.cpp

## 3.11. Referencia de la Clase Compresor

Diagrama de herencias de Compresor



La documentación para esta clase fue generada a partir de los siguientes ficheros:

- Compresor.h
- Compresor.cpp

### 3.12. Referencia de la Clase CompresorPPMC

Diagrama de herencias de CompresorPPMC

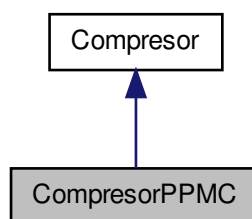
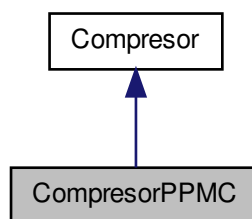


Diagrama de colaboración para CompresorPPMC:



La documentación para esta clase fue generada a partir de los siguientes ficheros:

- CompresorPPMC.h
- CompresorPPMC.cpp

### 3.13. Referencia de la Clase HashingExtensible

#### Métodos públicos

- virtual int **crear\_archivo** (std::string nombreArchivo)
- virtual int **eliminar\_archivo** ()
- virtual int **abrir\_archivo** (std::string nombreArchivo)
- virtual int **cerrar\_archivo** ()
- virtual int **agregar** ([RegistroClave](#) &reg)
- virtual int **devolver** ([ClaveX](#) &clave, [RegistroClave](#) \*reg)
- virtual int **modificar** ([RegistroClave](#) &elemN)
- virtual int **eliminar** ([ClaveX](#) &clave)

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- HashingExtensible.h
- HashingExtensible.cpp

### 3.14. Referencia de la Estructura header\_arbol

#### Atributos públicos

- unsigned int **minCantBytesClaves**
- unsigned int **maxCantBytesClaves**

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

- ArbolBMas.h

### 3.15. Referencia de la Estructura header\_arlf

#### Atributos públicos

- unsigned int **h\_recordSize**
- unsigned int **h\_maxRecords**

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

- ArchivoRegistrosLongitudFija.h

### 3.16. Referencia de la Estructura ManejadorRegistrosVariables::HeaderRegistroLibre

#### Atributos públicos

- unsigned short **espacioLibre**
- char **marcaBorrado**
- long **offsetProximoRegistroLibre**

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

- ManejadorRegistrosVariables.h

### 3.17. Referencia de la Clase Heap

#### Métodos públicos

- void **transformar\_en\_heap** ([RegistroVariable](#) vectorAOrdenar[], int tamaño)
- int **comparar\_registros\_variables** ([RegistroVariable](#) reg1, [RegistroVariable](#) reg2)

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- Heap.h
- Heap.cpp

### 3.18. Referencia de la Clase Idioma

#### Métodos públicos

- std::string **getIdioma** ()
- bool **cargar** (std::string idioma)

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- Idioma.h
- Idioma.cpp

### 3.19. Referencia de la Clase Indexador

#### Métodos públicos

- int **indexar** (std::string &directorioEntrada, std::string &directorioSalida)

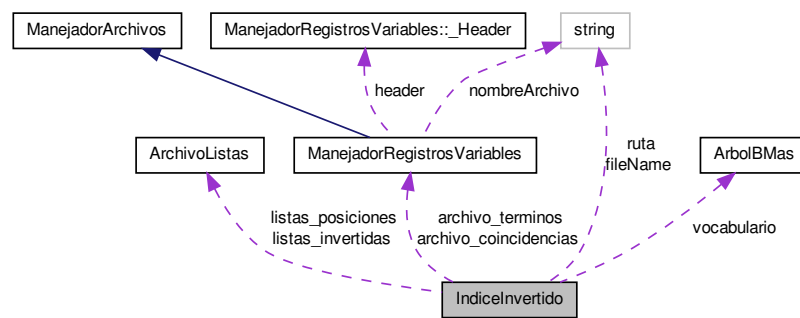
- int **consultar\_autor** (std::string &directorioSalida, std::string &unAutor)
- int **consultar\_titulo** (std::string &directorioSalida, std::string &titulo)
- int **consultar\_frase** (std::string &directorioSalida, std::string &frase)
- void **borrar\_cancion** (std::string &directorioSalida, int idCancion)

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- Indexador.h
- Indexador.cpp

### 3.20. Referencia de la Clase IndiceInvertido

Diagrama de colaboración para IndiceInvertido:



#### Métodos públicos

- virtual int **crear\_indice** (std::string directorioSalida, std::string fileNameee)
- virtual int **abrir\_indice** (std::string directorioSalida, std::string ileNameee)
- virtual int **cerrar\_indice** ()
- virtual int **agregar\_texto** (std::string texto, int IDcancion)
- virtual int **borrar\_indice** ()
- virtual int **buscar\_frase** (std::string frase, [RegistroVariable](#) &lista)

#### Atributos protegidos

- std::string **fileName**
- std::string **ruta**
- [ManejadorRegistrosVariables](#) **archivo\_terminos**
- [ArbolBMas](#) **vocabulario**

- [ArchivoListas](#) **listas\_invertidas**
- [ArchivoListas](#) **listas\_posiciones**
- [ManejadorRegistrosVariables](#) **archivo\_coincidencias**

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- IndiceInvertido.h
- IndiceInvertido.cpp

## 3.21. Referencia de la Clase IndiceInvertidoPorAutor

### Métodos públicos

- [IndiceInvertidoPorAutor](#) ()
- virtual int **crear\_indice** (std::string directorioSalida)
- virtual int **abrir\_indice** (std::string directorioSalida)
- virtual int **agregar\_cancion** ([RegistroCancion](#) cancion, int IDcancion)
- virtual long **buscar\_autor** (std::string autor, [RegistroVariable](#) &listaDeCanciones)
- virtual int **borrar\_indice** ()

### 3.21.1. Documentación del constructor y destructor

#### 3.21.1.1. [IndiceInvertidoPorAutor::IndiceInvertidoPorAutor](#) ( )

El archivo de listas se reconstruye cada vez que agregamos una nueva cancion al indice

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- IndiceInvertidoPorAutor.h
- IndiceInvertidoPorAutor.cpp

## 3.22. Referencia de la Clase IndiceInvertidoPorTitulo

### Métodos públicos

- virtual int **crear\_indice** (std::string directorioSalida)
- virtual int **abrir\_indice** (std::string directorioSalida)
- virtual int **agregar\_cancion** ([RegistroCancion](#) cancion, int IDcancion)
- virtual long **buscar\_titulo** (std::string titulo, [RegistroVariable](#) &listaDeCanciones)
- virtual int **borrar\_indice** ()

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- IndiceInvertidoPorTitulo.h
- IndiceInvertidoPorTitulo.cpp

### 3.23. Referencia de la Clase IterArbolBMas

#### Métodos públicos

- **IterArbolBMas** ([ArbolBMas](#) arbol)
- int **comienzo** (std::string operador, [ClaveX](#) clavelInicial)
- int **leer\_siguiente** ([RegistroClave](#) &regActual)

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- IterArbolBMas.h
- IterArbolBMas.cpp

### 3.24. Referencia de la Clase IterNodoSecuencial

#### Métodos públicos

- **IterNodoSecuencial** (const [NodoSecuencial](#) &nodoSec)
- int **get\_pos\_reg** (std::string unOperador, [ClaveX](#) clavelInicial)

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- IterNodoSecuencial.h
- IterNodoSecuencial.cpp

### 3.25. Referencia de la plantilla de la Clase ManejadorArchivoDatos-Basicos< TipoDato >

Diagrama de herencias de ManejadorArchivoDatosBasicos< TipoDato >

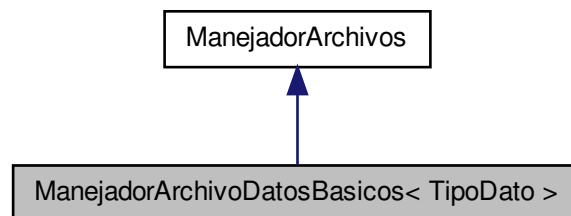
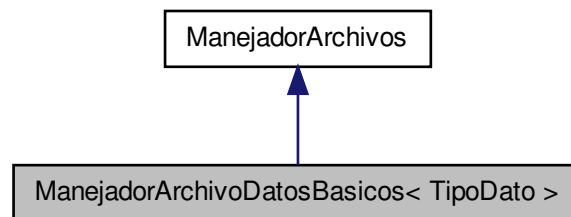




Diagrama de colaboración para `ManejadorArchivoDatosBasicos< TipoDato >`:



### Métodos públicos

- `int truncar` (long cantidadDatosDeseados)
- `int agregar` (const TipoDato dato)
- `int escribir` (const TipoDato dato, unsigned int pos)
- `int leer` (TipoDato \*dato, unsigned int pos)
- `int crear_archivo` (std::string p\_nombreArchivo)
- `int eliminar_archivo` (std::string p\_nombreArchivo)
- `int abrir_archivo` (std::string p\_nombreArchivo, std::string modo)
- `int cerrar_archivo` ()
- `int get_tamano_archivo` ()
- `int get_cantidad_datos_guardados` ()

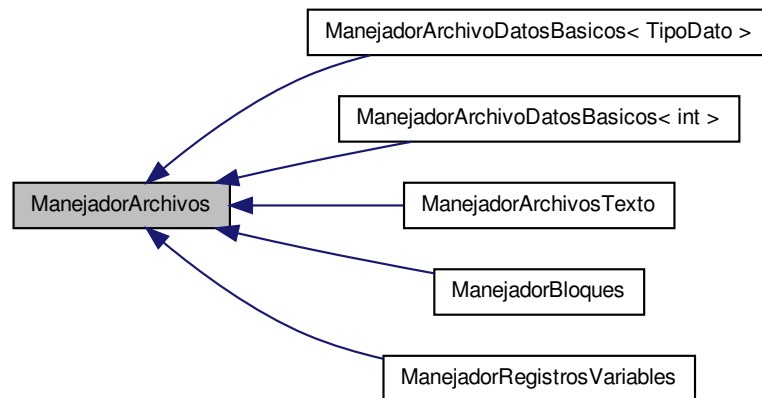
```
template<typename TipoDato> class ManejadorArchivoDatosBasicos< TipoDato >
```

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

- `ManejadorArchivoDatosBasicos.h`

### 3.26. Referencia de la Clase ManejadorArchivos

Diagrama de herencias de ManejadorArchivos



#### Métodos públicos

- virtual int **crear\_archivo** (std::string nombreArchivo)
- virtual bool **archivo\_existe** (std::string nombreArchivo) const throw ()
- virtual int **eliminar\_archivo** (std::string nombreArchivo) const throw ()

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- ManejadorArchivos.h
- ManejadorArchivos.cpp

### 3.27. Referencia de la Clase ManejadorArchivosTexto

Diagrama de herencias de ManejadorArchivosTexto

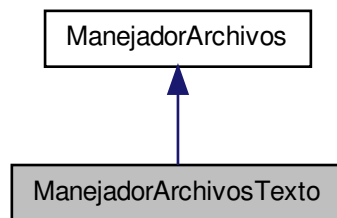
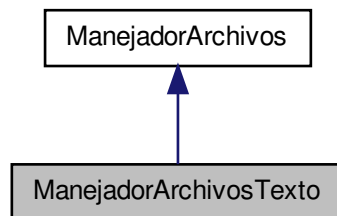


Diagrama de colaboración para ManejadorArchivosTexto:



#### Métodos públicos

- int **agregar\_linea** (std::string nombreArchivo, std::string linea) const throw ()
- int **mostrar\_archivo** (std::string nombreArchivo)
- int **hallar\_lineas** (std::string nombreArchivo, std::string linea) const throw ()

#### Atributos públicos estáticos

- static const int **LINEAS\_NO\_HALLADAS** = -1

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- `ManejadorArchivosTexto.h`
- `ManejadorArchivosTexto.cpp`

### 3.28. Referencia de la Clase ManejadorBloques

Diagrama de herencias de ManejadorBloques

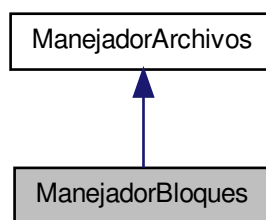
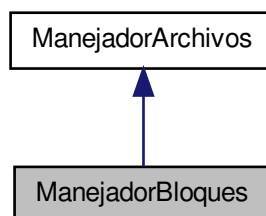


Diagrama de colaboración para ManejadorBloques:



#### Métodos públicos

- `unsigned int get_cantidad_bloques ()`
- `unsigned int get_tamano_bloque ()`
- `void set_ruta (std::string ruta)`
- `std::string get_ruta ()`
- `int crear_archivo (std::string nombreArchivo, unsigned int tamBloque)`

- int **eliminar\_archivo** (std::string nombreArchivo)
- int **abrir\_archivo** (std::string nombreArchivo, std::string modo)
- int **cerrar\_archivo** ()
- int **escribir\_bloque** (Bloque \*bloque)
- int **sobreescribir\_bloque** (Bloque \*bloque, unsigned int numBloque)
- Bloque \* **crear\_bloque** ()
- Bloque \* **obtener\_bloque** (unsigned int numBloque)

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- ManejadorBloques.h
- ManejadorBloques.cpp

### 3.29. Referencia de la Clase ManejadorRegistrosVariables

Diagrama de herencias de ManejadorRegistrosVariables

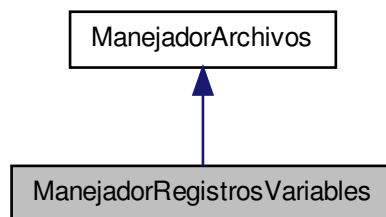
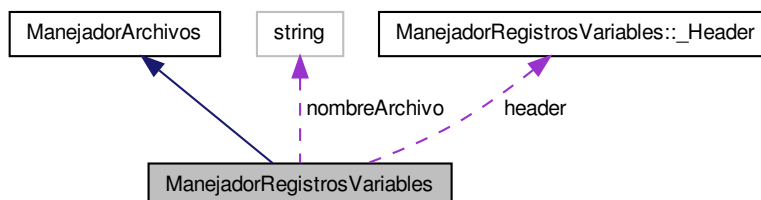


Diagrama de colaboración para ManejadorRegistrosVariables:



## Clases

- struct [\\_Header](#)
- struct [HeaderRegistroLibre](#)

## Tipos públicos

- typedef struct [ManejadorRegistrosVariables::\\_Header](#) **Header**

## Métodos públicos

- virtual int **crear\_archivo** (string nombreArchivo)
- virtual int **abrir\_archivo** (string nombreArchivo)
- virtual long **agregar\_registro** ([RegistroVariable](#) \*registro)
- virtual int **get\_cantidad\_registros** ()
- virtual int **get\_cantidad\_registros\_ocupados** ()
- virtual long **get\_tamano\_archivo** ()
- virtual long **get\_registro\_ocupado** ([RegistroVariable](#) \*registro, unsigned int numeroRegistro)
- virtual int **get\_registro\_por\_offset** ([RegistroVariable](#) \*registro, unsigned long offset)
- virtual long **eliminar\_registro\_ocupado** (unsigned int numeroRegistro)
- virtual int **eliminar\_registro\_por\_offset** (unsigned long offset)
- int **refactorizar** ()

## Atributos públicos estáticos

- static const unsigned int **OFFSET\_PRIMER\_REGISTRO** = sizeof(header)

## Métodos protegidos

- virtual void **\_leer\_header** ()
- virtual void **\_cerrar\_archivo** (fstream \*archivo)
- virtual void **\_resetear\_header** ()
- virtual void **\_guardar\_header** ()
- virtual void **\_desvincular** ()
- virtual long **\_append\_registro** ([RegistroVariable](#) \*registro)
- virtual bool **\_registro\_fue\_eliminado** (unsigned int numeroRegistro)
- virtual long **\_get\_offset\_registro** (unsigned int numeroRegistro)
- virtual long **\_buscar\_registro\_libre** (unsigned short espacioNecesario, long &offsetLibreAnterior, long &offsetLibreSiguiente)
- virtual void **\_pegar\_registro** ([RegistroVariable](#) \*registro, long offset)
- void **\_reemplazar** (string &nombreArchivoAuxiliar)
- void **\_cambiar\_header** ([Header](#) &headerCambiar)

#### Atributos protegidos

- `Header header`
- `string nombreArchivo`

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- `ManejadorRegistrosVariables.h`
- `ManejadorRegistrosVariables.cpp`

### 3.30. Referencia de la Estructura mb\_header

#### Atributos públicos

- `unsigned int tamanoBloque`
- `unsigned int cantidadBloques`
- `int proximoBloqueLibre`

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

- `ManejadorBloques.h`

### 3.31. Referencia de la Estructura mdb\_header

#### Atributos públicos

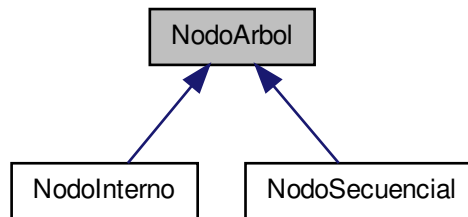
- `unsigned int cantidadDatosGuardados`

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

- `ManejadorArchivoDatosBasicos.h`

### 3.32. Referencia de la Clase NodoArbol

Diagrama de herencias de NodoArbol



#### Métodos públicos

- **NodoArbol** (char tipoNodo=TIPO\_HOJA)
- virtual int **desempaquetar** (const **Bloque** \*bloque)
- virtual int **empaquetar** (**Bloque** \*bloque)
- virtual bool **es\_hoja** ()
- virtual bool **es\_interno** ()

#### Atributos protegidos

- char **tipoNodo**

La documentación para esta clase fue generada a partir de los siguientes archivos:

- **NodoArbol.h**
- **NodoArbol.cpp**



### 3.33. Referencia de la Clase NodoInterno

Diagrama de herencias de NodoInterno

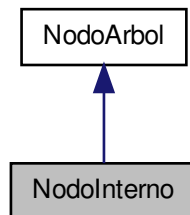
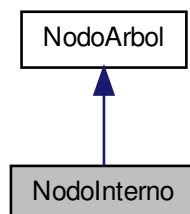


Diagrama de colaboración para NodoInterno:



#### Tipos públicos

- `typedef vector< ClaveX >::iterator IteradorVectorClaves`
- `typedef unsigned int TipoHijo`

#### Métodos públicos

- **NodoInterno** (unsigned int minCantidadBytes=MINIMA\_CANTIDAD\_BYTES\_DEFAULT, unsigned int maxCantidadBytes=MAXIMA\_CANTIDAD\_BYTES\_DEFAULT)

- `bool esta_vacio () const`
- `int get_tamano_ocupado ()`
- `unsigned short get_cantidad_claves ()`
- `unsigned short get_cantidad_hijos ()`
- `virtual int get_hijo_izquierdo (TipoHijo &hijo, const ClaveX &clave)`
- `virtual int get_hijo_derecho (TipoHijo &hijo, const ClaveX &clave)`
- `int get_clave (unsigned short numeroClave, ClaveX &clave)`
- `virtual int get_hijo (TipoHijo &hijo, unsigned short numeroHijo)`
- `int get_clave_mitad (ClaveX &clave)`
- `virtual int buscar_clave (const ClaveX &clave, unsigned short &posicionClave)`
- `int insertar_clave (const ClaveX &clave, unsigned short &posicionOcurrencia-Insercion)`
- `int insertar_hijo (TipoHijo hijo, unsigned short posicion=POSICION_FIN_VECTOR_HIJOS)`
- `int modificar_hijo (TipoHijo valor, unsigned short numeroHijo)`
- `int remover_clave (const ClaveX &clave, unsigned short &posicionEliminacion)`
- `int remover_clave (unsigned short numeroClave, ClaveX &clave)`
- `int remover_hijo (unsigned short numeroHijo)`
- `bool es_hoja ()`
- `bool es_interno ()`
- `virtual bool hay_overflow () const`
- `virtual bool hay_underflow () const`
- `void imprimir ()`
- `void imprimir_claves () const`
- `void imprimir_hijos () const`
- `int insertar_hijo_derecho (const ClaveX &clave, TipoHijo valor=HIJO_INVALIDO)`
- `int insertar_hijo_izquierdo (const ClaveX &clave, TipoHijo valor)`
- `int modificar_hijo_derecho (const ClaveX &, TipoHijo valor)`
- `int modificar_hijo_izquierdo (const ClaveX &, TipoHijo valor)`
- `void limpiar ()`
- `int buscar_hijo (TipoHijo valorHijo, unsigned short &posicionHijo)`
- `int obtener_hijo_siguiente_a (TipoHijo valorHijo, int &hijoSiguiente)`
- `vector< ClaveX > get_claves ()`
- `vector< TipoHijo > get_hijos ()`
- `int empaquetar (Bloque *bloque)`
- `int desempaquetar (const Bloque *bloque)`

### Atributos públicos estáticos

- `static const unsigned short POSICION_FIN_VECTOR_HIJOS = USHRT_MAX`
- `static const unsigned int MINIMA_CANTIDAD_BYTES_DEFAULT = 0`
- `static const unsigned int MAXIMA_CANTIDAD_BYTES_DEFAULT = 4096`

### Atributos protegidos

- unsigned int **maxCantidadBytes**
- unsigned int **minCantidadBytes**
- unsigned int **cantidadBytesOcupados**
- std::vector< [ClaveX](#) > **vectorClaves**
- std::vector< TipoHijo > **vectorHijos**

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- NodoInterno.h
- NodoInterno.cpp

## 3.34. Referencia de la Clase NodoSecuencial

Diagrama de herencias de NodoSecuencial

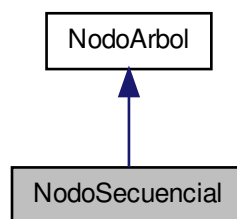
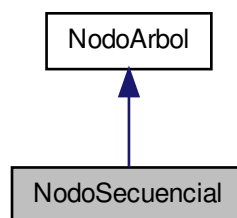


Diagrama de colaboración para NodoSecuencial:



### Métodos públicos

- **NodoSecuencial** (unsigned int minBytesOcupados, unsigned int maxBytesOcupados)
- **NodoSecuencial** (const [NodoSecuencial](#) &otro)
- **NodoSecuencial** & **operator=** (const [NodoSecuencial](#) &otro)
- void **imprimir** ()
- bool **tiene\_carga\_minima** ()
- void **set\_proximo\_nodo** (TipoPuntero p\_prox)
- int **insertar** (const [RegistroClave](#) &registro, std::vector< [RegistroClave](#) > &regs-Overflow)
- int **eliminar** (const [ClaveX](#) &clave, std::vector< [RegistroClave](#) > &regs-Underflow)
- int **buscar** (const [ClaveX](#) &claveBuscada, [RegistroClave](#) \*\*regDevuelto)
- int **empaquetar** ([Bloque](#) \*bloque)
- int **desempaquetar** (const [Bloque](#) \*)
- unsigned int **get\_bytes\_ocupados** () const
- TipoPuntero **get\_proximo\_nodo** () const
- unsigned int **get\_cantidad\_registros** () const
- std::vector< [RegistroClave](#) > **get\_registros** () const
- unsigned int **get\_max\_bytes** () const
- unsigned int **get\_min\_bytes** () const
- bool **esta\_vacio** ()

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- [NodoSecuencial.h](#)
- [NodoSecuencial.cpp](#)

## 3.35. Referencia de la Clase Normalizador

### Métodos públicos

- char **transformar\_char** (wchar\_t c)
- wstring **a\_minuscula** (wstring &linea)
- string **normalizar\_string** (wstring &linea)
- int **normalizar** (string pathArchivoEntrada, string pathArchivoSalida)
- string **normalizar\_input** (string linea)

### Métodos protegidos

- char **filtrar\_char** (wchar\_t c)

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- [Normalizador.h](#)
- [Normalizador.cpp](#)

### 3.36. Referencia de la Clase ParserCanciones

#### Métodos públicos

- int **crear** (std::string &dir)
- bool **fin\_directorio** () const
- int **obtener\_proxima\_cancion** ([RegistroCancion](#) &reg, std::string &nombre-Archivo)
- unsigned int **get\_cantidad\_archivos** () const

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- ParserCanciones.h
- ParserCanciones.cpp

### 3.37. Referencia de la Clase RegistroCancion

Diagrama de herencias de RegistroCancion

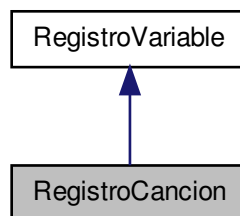
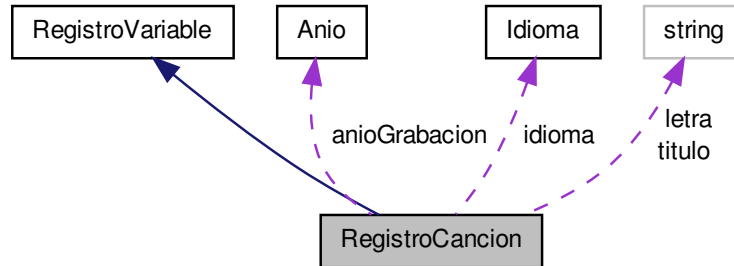


Diagrama de colaboración para RegistroCancion:



### Métodos públicos

- string **generar\_clave** () const throw ()
- int **cargar** (const char \*dato, unsigned short tamanoDato) throw ()
- unsigned short **get\_cantidad\_autores** () const throw ()
- string **get\_autor** (unsigned short numeroAutor) throw ()
- int **get\_anio** () throw ()
- string **get\_idioma** () throw ()
- string **get\_letra** () throw ()
- string **get\_titulo** () throw ()
- void **limpiar\_buffer** () throw ()
- unsigned short **get\_cantidad\_parametros** ()
- int **empaquetar** (char \*copia) throw ()
- int **desempaquetar** (const char \*copia) throw ()
- int **comprimir** ([Compresor](#) &compresor)

### Métodos protegidos

- virtual void **limpiar\_autores** () throw ()
- virtual unsigned short **obtener\_autores** () throw ()
- virtual unsigned short **contar\_parametros** () throw ()
- virtual bool **obtener\_parametro** (unsigned short numeroParametro, string &parametro) throw ()
- virtual int **obtener\_anio\_grabacion** () throw ()
- virtual bool **obtener\_titulo** () throw ()
- virtual bool **obtener\_idioma** () throw ()
- virtual bool **obtener\_letra** () throw ()

### Atributos protegidos

- `std::string * autores`
- `unsigned short cantidadAutores`
- `Idioma idioma`
- `Anio anioGrabacion`
- `string titulo`
- `string letra`
- `unsigned short cantidadParametros`

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- `RegistroCancion.h`
- `RegistroCancion.cpp`

## 3.38. Referencia de la Clase RegistroClave

Diagrama de herencias de RegistroClave

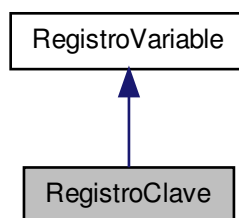
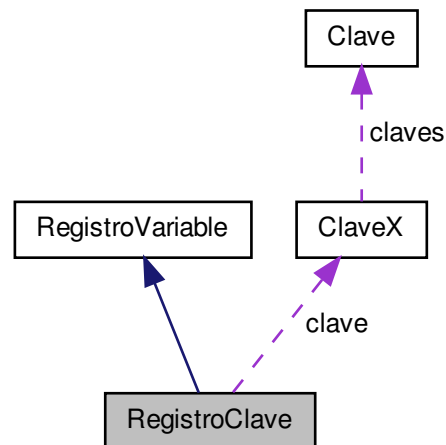


Diagrama de colaboración para RegistroClave:



### Métodos públicos

- **RegistroClave** (const [RegistroClave](#) &otro)
- [RegistroClave](#) & **operator=** (const [RegistroClave](#) &otro)
- virtual void **set\_clave** (const [ClaveX](#) &clave)
- virtual [ClaveX](#) **get\_clave** () const
- void **limpiar\_campos** () throw ()
- virtual int **desempaquetar** (const char \*copia) throw ()
- virtual bool **operator<** (const [RegistroClave](#) &rc) const
- virtual bool **operator>** (const [RegistroClave](#) &rc) const
- virtual bool **operator==** (const [RegistroClave](#) &rc) const
- virtual bool **operator<=** (const [RegistroClave](#) &rc) const
- virtual bool **operator>=** (const [RegistroClave](#) &rc) const

### Atributos públicos estáticos

- static const unsigned short **NUMERO\_CAMPO\_CLAVE** = 0

### Métodos protegidos

- void **\_agregar\_campo\_clave** ()



### Atributos protegidos

- [ClaveX](#) clave

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- RegistroClave.h
- RegistroClave.cpp

## 3.39. Referencia de la Clase RegistroFijo

### Métodos públicos

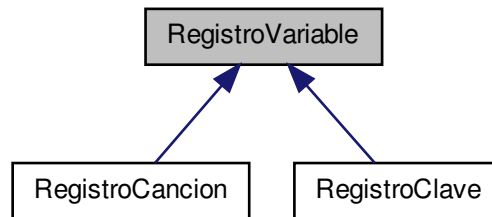
- **RegistroFijo** (int numFields, int fieldSizes[])
- **RegistroFijo** (int maxNumFields, int maxRecordSize)
- void **clear** ()
- bool **addFieldOfSize** (int fieldSize)
- int **getNumberOfFields** ()
- int **getRecordSize** ()
- int **getFieldNumber** (int number, std::string &field)
- bool **pack** (const char \*string)
- bool **unpack** (char \*string)
- int **write** (FILE \*file) const
- int **read** (FILE \*file)
- void **show** ()

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- RegistroFijo.h
- RegistroFijo.cpp

### 3.40. Referencia de la Clase RegistroVariable

Diagrama de herencias de RegistroVariable



#### Métodos públicos

- **RegistroVariable** & **operator=** (const [RegistroVariable](#) &otro)
- **RegistroVariable** (const [RegistroVariable](#) &otro)
- virtual unsigned short **get\_cantidad\_campos** () throw ()
- virtual char \* **get\_buffer** () throw ()
- virtual unsigned short **get\_tamano** () throw ()
- virtual int **eliminar** () throw ()
- virtual int **agregar\_campo** (const char \*campo, unsigned short tamanoCampo) throw ()
- virtual int **get\_tamano\_campo** (unsigned short numeroCampo)
- virtual int **recuperar\_campo** (char \*copia, unsigned short numeroCampo) throw ()
- virtual void **limpiar\_campos** () throw ()
- virtual int **empaquetar** (char \*copia) throw ()
- virtual int **get\_tamano\_empaquetado** () const
- virtual int **desempaquetar** (const char \*copia) throw ()
- virtual bool **fue\_eliminado** () throw ()
- virtual int **comprimir** ([Compresor](#) &compresor)
- virtual bool **esta\_limpio** ()
- virtual std::string **mostrar** () throw ()
- virtual int **agregar\_datos** (const char \*datos, unsigned short tamanoDatos) throw ()

#### Atributos públicos estáticos

- static const unsigned short **CAMPO\_ENCADENAMIENTO\_LIBRES** = 1

### Métodos protegidos

- virtual int **seek\_numero\_campo** (unsigned short numeroCampo)
- virtual std::string **mostrar\_campo** (unsigned short numeroCampo)
- void **\_inicializar\_buffer** ()

### Atributos protegidos

- unsigned short **tamano**
- char \* **buffer**

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- RegistroVariable.h
- RegistroVariable.cpp

## 3.41. Referencia de la Clase ResolvedorConsultas

### Métodos públicos

- **ResolvedorConsultas** (std::string &directorioSalida)
- std::vector< int > **get\_id\_canciones\_autor** (std::string &autor)
- std::string **get\_nombre\_archivo** (int id\_cancion)
- std::vector< int > **get\_ids\_canciones\_frases** (std::string &frase)
- [RegistroCancion](#) \* **get\_reg\_completo** (int id\_cancion)
- int **get\_id\_cancion\_titulo** (std::string &titulo)

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- ResolvedorConsultas.h
- ResolvedorConsultas.cpp

## 3.42. Referencia de la Clase SortExterno

### Métodos públicos

- **SortExterno** (std::string archAOrdenar)
- void **ordenar\_archivo** ()
- std::vector< string > **\_getVector** ()
- void **\_generar\_runs** ()
- void **\_merge** ()

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- SortExterno.h
- SortExterno.cpp

### 3.43. Referencia de la Clase Tabla

#### Métodos públicos

- int **crear** (string rutaArchivo)
- int **eliminar** ()
- void **set\_ruta** (string unaRuta)
- int **get\_tamano** ()
- int **obtener\_valor** (int posicion)
- int **cambiar\_valor** (int posicion, int nuevoValor)
- int **dividir** ()
- int **duplicar** ()

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- Tabla.h
- Tabla.cpp

### 3.44. Referencia de la Clase Texto

#### Métodos públicos

- int **parsear** (std::string unTexto)
- int **get\_proxima\_palabra** (std::string &palabra)

La documentación para esta clase fue generada a partir de los siguientes ficheros:

- Texto.h
- Texto.cpp