

## Organización de Datos

Generado por Doxygen 1.7.6.1

Domingo, 26 de Mayo de 2013 15:34:33



# 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 Clase Anio . . . . .	5
3.2	Referencia de la Clase ArbolBMas . . . . .	5
3.3	Referencia de la Clase ArchivoListas . . . . .	7
3.3.1	Documentación del constructor y destructor . . . . .	7
3.3.1.1	ArchivoListas . . . . .	7
3.4	Referencia de la Clase ArchivoRegistrosLongitudFija . . . . .	7
3.5	Referencia de la Clase Bloque . . . . .	8
3.6	Referencia de la Clase Clave . . . . .	9
3.7	Referencia de la Clase ClaveNumerica . . . . .	10
3.8	Referencia de la Clase ClaveString . . . . .	11
3.9	Referencia de la Clase ClaveX . . . . .	13
3.10	Referencia de la Clase Compresor . . . . .	14
3.11	Referencia de la Clase CompresorPPMC . . . . .	15
3.12	Referencia de la Clase HashingExtensible . . . . .	16
3.13	Referencia de la Estructura ManejadorRegistrosVariables::Header . . . . .	16
3.14	Referencia de la Estructura header_arbol . . . . .	16
3.15	Referencia de la Estructura header_arlf . . . . .	17
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 . . . . .	18
3.19.1 Documentación de las funciones miembro . . . . .	18
3.19.1.1 indexar . . . . .	18
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 . . . . .	20
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 > . . . . .	21
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 . . . . .	27
3.33 Referencia de la Clase NodoInterno . . . . .	28
3.34 Referencia de la Clase NodoSecuencial . . . . .	31
3.35 Referencia de la Clase Normalizador . . . . .	32
3.36 Referencia de la Clase ParserCanciones . . . . .	32
3.37 Referencia de la Clase RegistroCancion . . . . .	33
3.38 Referencia de la Clase RegistroClave . . . . .	35
3.39 Referencia de la Clase RegistroFijo . . . . .	36
3.40 Referencia de la Clase RegistroVariable . . . . .	37
3.41 Referencia de la Clase SortExterno . . . . .	38
3.42 Referencia de la Clase Tabla . . . . .	38
3.43 Referencia de la Clase Texto . . . . .	39

# Capítulo 1

## Índice de clases

### 1.1. Jerarquía de la clase

Esta lista de herencias esta ordenada aproximadamente por orden alfabético:

Anio . . . . .	5
ArbolBMas . . . . .	5
ArchivoListas . . . . .	7
ArchivoRegistrosLongitudFija . . . . .	7
Bloque . . . . .	8
Clave . . . . .	9
ClaveNumerica . . . . .	10
ClaveString . . . . .	11
ClaveX . . . . .	13
Compresor . . . . .	14
CompresorPPMC . . . . .	15
HashingExtensible . . . . .	16
ManejadorRegistrosVariables::Header . . . . .	16
header_arbol . . . . .	16
header_arlf . . . . .	17
ManejadorRegistrosVariables::HeaderRegistroLibre . . . . .	17
Heap . . . . .	17
Idioma . . . . .	17
Indexador . . . . .	18
IndiceInvertido . . . . .	18
IndiceInvertidoPorAutor . . . . .	19
IndiceInvertidoPorTitulo . . . . .	20
IterArbolBMas . . . . .	20
IterNodoSecuencial . . . . .	20
ManejadorArchivos . . . . .	22
ManejadorArchivoDatosBasicos< TipoDato > . . . . .	21
ManejadorArchivosTexto . . . . .	23
ManejadorBloques . . . . .	24

ManejadorRegistrosVariables . . . . .	25
mb_header . . . . .	27
mdb_header . . . . .	27
NodoArbol . . . . .	27
NodoInterno . . . . .	28
NodoSecuencial . . . . .	31
Normalizador . . . . .	32
ParserCanciones . . . . .	32
RegistroFijo . . . . .	36
RegistroVariable . . . . .	37
RegistroCancion . . . . .	33
RegistroClave . . . . .	35
SortExterno . . . . .	38
Tabla . . . . .	38
Texto . . . . .	39

## Capítulo 2

# Índice de clases

### 2.1. Lista de clases

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

Anio	5
ArbolBMas	5
ArchivoListas	7
ArchivoRegistrosLongitudFija	7
Bloque	8
Clave	9
ClaveNumerica	10
ClaveString	11
ClaveX	13
Compresor	14
CompresorPPMC	15
HashingExtensible	16
ManejadorRegistrosVariables::Header	16
header_arbol	16
header_arlf	17
ManejadorRegistrosVariables::HeaderRegistroLibre	17
Heap	17
Idioma	17
Indexador	18
IndiceInvertido	18
IndiceInvertidoPorAutor	19
IndiceInvertidoPorTitulo	20
IterArbolBMas	20
IterNodoSecuencial	20
ManejadorArchivoDatosBasicos< TipoDato >	21
ManejadorArchivos	22
ManejadorArchivosTexto	23
ManejadorBloques	24
ManejadorRegistrosVariables	25

mb_header	27
mdb_header	27
NodoArbol	27
NodoInterno	28
NodoSecuencial	31
Normalizador	32
ParserCanciones	32
RegistroCancion	33
RegistroClave	35
RegistroFijo	36
RegistroVariable	37
SortExterno	38
Tabla	38
Texto	39



## Capítulo 3

# Documentación de las clases

### 3.1. 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:

- src/CapaLogica/ManejoArchivos/Anio.h
- src/CapaLogica/ManejoArchivos/Anio.cpp

### 3.2. 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:

- src/CapaLogica/ArbolBMas/ArbolBMas.h
- src/CapaLogica/ArbolBMas/ArbolBMas.cpp

### 3.3. Referencia de la Clase ArchivoListas

#### Métodos públicos

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

#### 3.3.1. Documentación del constructor y destructor

##### 3.3.1.1. ArchivoListas::ArchivoListas ( )

Suponemos que no existen registros vacios

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

- src/CapaLogica/Indices/ArchivoListas.h
- src/CapaLogica/Indices/ArchivoListas.cpp

### 3.4. 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:

- src/CapaFisica/ArchivoRegistrosLongitudFija.h
- src/CapaFisica/ArchivoRegistrosLongitudFija.cpp

### 3.5. 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

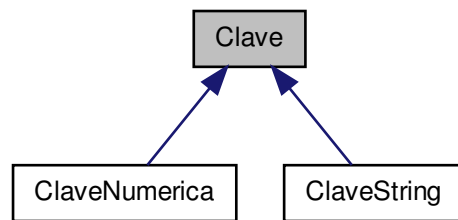
- char \* **bufferBloque**
- unsigned int **espacioLibre**
- unsigned int **tamanoBloque**
- unsigned int **espacioLibreOffset**

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

- src/CapaFisica/Bloque.h
- src/CapaFisica/Bloque.cpp

## 3.6. 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:

- src/CapaLogica/ManejoArchivos/Clave.h
- src/CapaLogica/ManejoArchivos/Clave.cpp

### 3.7. Referencia de la Clase ClaveNumerica

Diagrama de herencias de ClaveNumerica

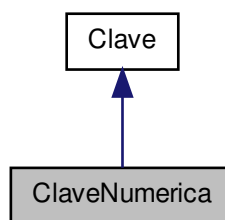
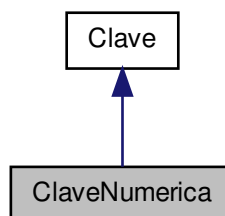


Diagrama de colaboración para ClaveNumerica:



#### 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:

- src/CapaLogica/ManejoArchivos/ClaveNumerica.h
- src/CapaLogica/ManejoArchivos/ClaveNumerica.cpp

### 3.8. Referencia de la Clase ClaveString

Diagrama de herencias de ClaveString

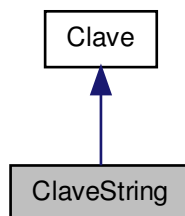
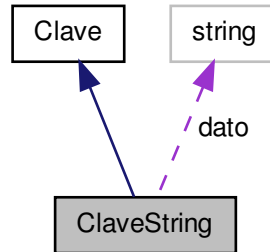


Diagrama de colaboración para ClaveString:



### Métodos públicos

- virtual void **desempaquetar** (char \*buffer, unsigned short tamañoBuffer=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\_tamaño** ()
- 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

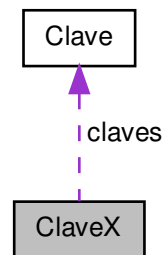
- string **dato**

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

- src/CapaLogica/ManejoArchivos/ClaveString.h
- src/CapaLogica/ManejoArchivos/ClaveString.cpp

## 3.9. 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

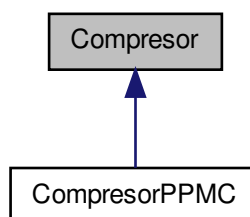
- TipoClave **tipoClave**
- [Clave](#) \* **claves** [CANTIDAD\_TIPOS\_CLAVE]

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

- src/CapaLogica/ManejoArchivos/ClaveX.h
- src/CapaLogica/ManejoArchivos/ClaveX.cpp

## 3.10. Referencia de la Clase Compresor

Diagrama de herencias de Compresor



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

- src/CapaLogica/Compresor/Compresor.h
- src/CapaLogica/Compresor/Compresor.cpp

### 3.11. Referencia de la Clase CompresorPPMC

Diagrama de herencias de CompresorPPMC

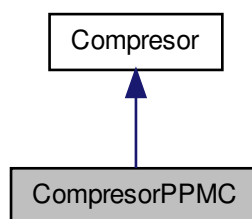
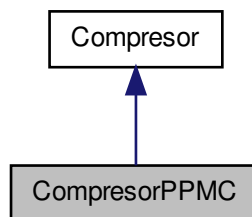


Diagrama de colaboración para CompresorPPMC:



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

- src/CapaLogica/Compresor/CompresorPPMC.h
- src/CapaLogica/Compresor/CompresorPPMC.cpp

### 3.12. 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:

- src/CapaLogica/HashingExtensible/HashingExtensible.h
- src/CapaLogica/HashingExtensible/HashingExtensible.cpp

### 3.13. 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:

- src/CapaFisica/ManejadorRegistrosVariables.h

### 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:

- src/CapaLogica/ArbolBMas/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:

- src/CapaFisica/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:

- src/CapaFisica/ManejadorRegistrosVariables.h

### 3.17. Referencia de la Clase Heap

#### Métodos públicos

- void **ordenar** ([RegistroClave](#) vectorAOrdenar[], int tamano)

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

- src/Utilitarios/Heap.h
- src/Utilitarios/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 archivos:

- src/CapaLogica/ManejoArchivos/Idioma.h
- src/CapaLogica/ManejoArchivos/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)

#### 3.19.1. Documentación de las funciones miembro

##### 3.19.1.1. int Indexador::indexar ( std::string & *directorioEntrada*, std::string & *directorioSalida* )

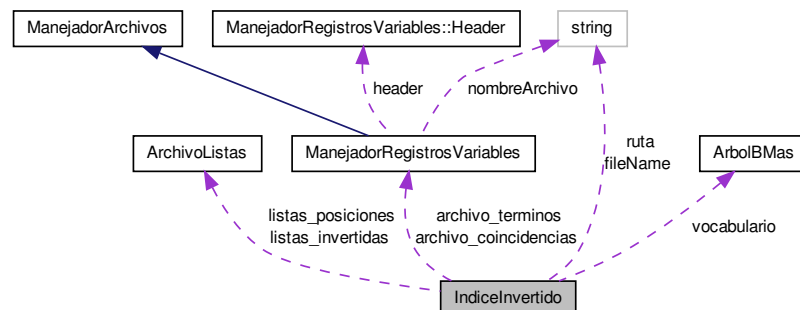
incrementamos el ID de la cancion

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

- src/CapaInterfaz/Indexador.h
- src/CapaInterfaz/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 fileNamee)
- virtual int **abrir\_indice** (std::string directorioSalida, std::string fileNamee)
- virtual int **agregar\_cancion** ([RegistroCancion](#) cancion, 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:

- src/CapaLogica/Indices/IndiceInvertido.h
- src/CapaLogica/Indices/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:

- src/CapaLogica/Indices/IndiceInvertidoPorAutor.h
- src/CapaLogica/Indices/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:

- src/CapaLogica/Indices/IndiceInvertidoPorTitulo.h
- src/CapaLogica/Indices/IndiceInvertidoPorTitulo.cpp

### 3.23. Referencia de la Clase `IterArbolBMas`

#### Métodos públicos

- **IterArbolBMas** ([ArbolBMas](#) arbol)
- int **start** (std::string operador, [ClaveX](#) claveInicial)
- int **readNext** ([RegistroClave](#) &regActual)

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

- src/CapaLogica/ArbolBMas/IterArbolBMas.h
- src/CapaLogica/ArbolBMas/IterArbolBMas.cpp

### 3.24. Referencia de la Clase `IterNodoSecuencial`

#### Métodos públicos

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

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

- src/CapaLogica/ArbolBMas/IterNodoSecuencial.h
- src/CapaLogica/ArbolBMas/IterNodoSecuencial.cpp



### 3.25. Referencia de la plantilla de la Clase `ManejadorArchivoDatosBasicos< TipoDato >`

Diagrama de herencias de `ManejadorArchivoDatosBasicos< TipoDato >`

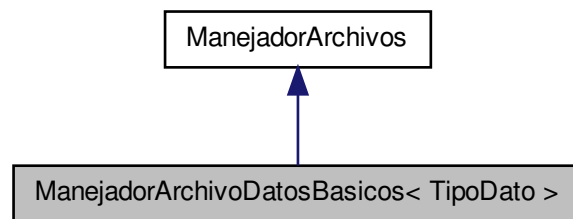
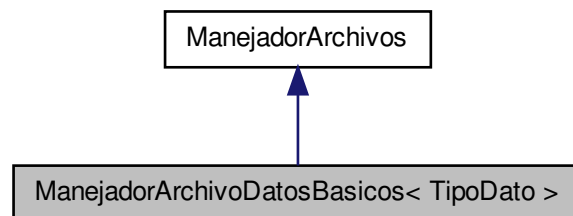


Diagrama de colaboración para `ManejadorArchivoDatosBasicos< TipoDato >`:



#### 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** ()

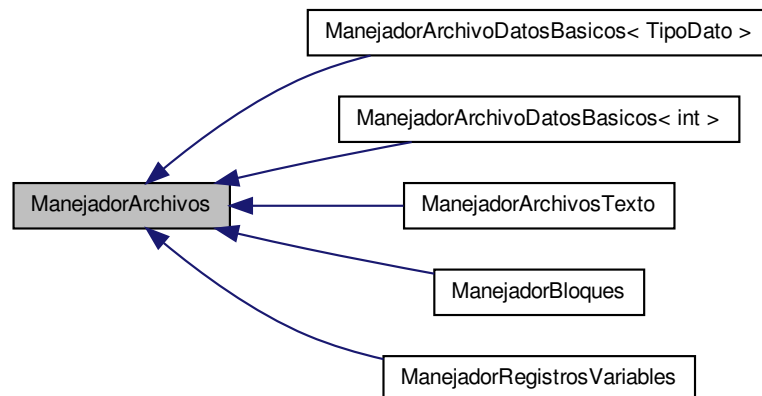
```
template<typename TipoDato> class ManejadorArchivoDatosBasicos< TipoDato >
```

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

- src/CapaFisica/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:

- src/CapaFisica/ManejadorArchivos.h
- src/CapaFisica/ManejadorArchivos.cpp

### 3.27. Referencia de la Clase ManejadorArchivosTexto

Diagrama de herencias de ManejadorArchivosTexto

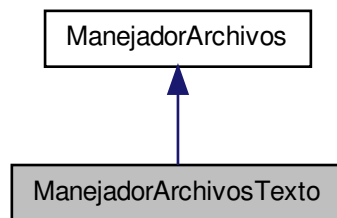
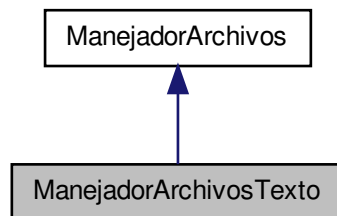


Diagrama de colaboración para ManejadorArchivosTexto:



#### 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:

- src/CapaFisica/ManejadorArchivosTexto.h
- src/CapaFisica/ManejadorArchivosTexto.cpp

### 3.28. Referencia de la Clase ManejadorBloques

Diagrama de herencias de ManejadorBloques

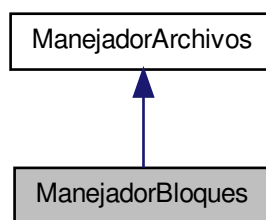
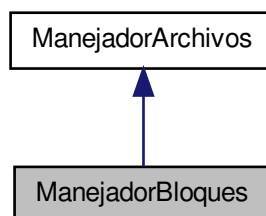


Diagrama de colaboración para ManejadorBloques:



#### 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:

- src/CapaFisica/ManejadorBloques.h
- src/CapaFisica/ManejadorBloques.cpp

### 3.29. Referencia de la Clase ManejadorRegistrosVariables

Diagrama de herencias de ManejadorRegistrosVariables

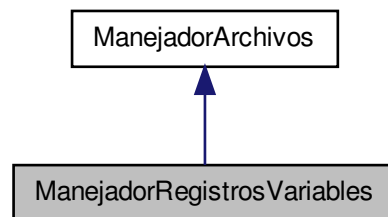
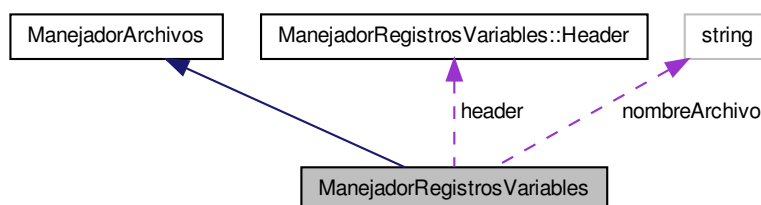


Diagrama de colaboración para ManejadorRegistrosVariables:



## Clases

- struct [Header](#)
- struct [HeaderRegistroLibre](#)

## 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 short numeroRegistro)
- virtual int **get\_registro\_por\_offset** ([RegistroVariable](#) \*registro, unsigned long offset)
- virtual long **eliminar\_registro\_ocupado** (unsigned short numeroRegistro)
- virtual int **eliminar\_registro\_por\_offset** (unsigned long offset)

## 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 short numeroRegistro)
- virtual long **\_get\_offset\_registro** (unsigned short numeroRegistro)
- virtual long **\_buscar\_registro\_libre** (unsigned short espacioNecesario, long &offsetLibreAnterior, long &offsetLibreSiguiente)
- virtual void **\_pegar\_registro** ([RegistroVariable](#) \*registro, long offset)

## Atributos protegidos

- [Header](#) header
- string nombreArchivo

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

- src/CapaFisica/ManejadorRegistrosVariables.h
- src/CapaFisica/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:

- src/CapaFisica/ManejadorBloques.h

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

#### Atributos públicos

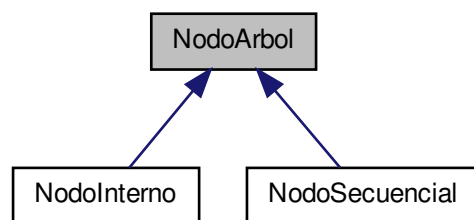
- unsigned int **cantidadDatosGuardados**

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

- src/CapaFisica/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:

- src/CapaLogica/ArbolBMas/NodoArbol.h
- src/CapaLogica/ArbolBMas/NodoArbol.cpp

### 3.33. Referencia de la Clase NodoInterno

Diagrama de herencias de NodoInterno

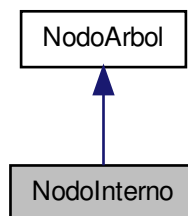
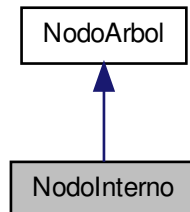




Diagrama de colaboración para NodoInterno:



### 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** ()
- int **set\_hijo\_izquierdo** (const [ClaveX](#) &clave, TipoHijo valor)
- int **set\_hijo\_derecho** (const [ClaveX](#) &clave, TipoHijo valor)
- 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:

- `src/CapaLogica/ArbolBMas/NodoInterno.h`
- `src/CapaLogica/ArbolBMas/NodoInterno.cpp`

### 3.34. Referencia de la Clase NodoSecuencial

Diagrama de herencias de NodoSecuencial

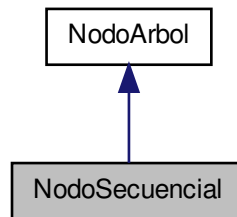
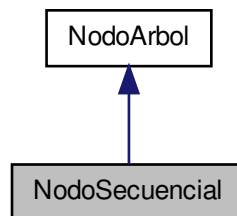


Diagrama de colaboración para NodoSecuencial:



#### 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:

- src/CapaLogica/ArbolBMas/NodoSecuencial.h
- src/CapaLogica/ArbolBMas/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)

#### Métodos protegidos

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

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

- src/Utilitarios/Normalizador.h
- src/Utilitarios/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:

- src/CapaLogica/Parser/ParserCanciones.h
- src/CapaLogica/Parser/ParserCanciones.cpp

### 3.37. Referencia de la Clase RegistroCancion

Diagrama de herencias de RegistroCancion

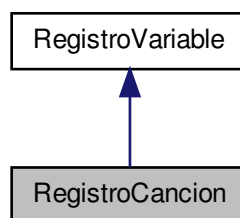
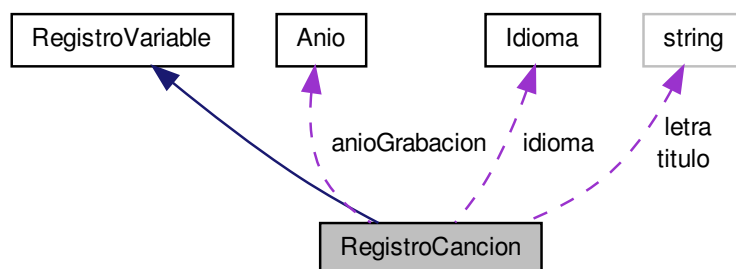


Diagrama de colaboración para RegistroCancion:



### 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 ()
- 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 ()
- int **comprimir** ([Compresor](#) &compresor)

### Métodos protegidos

- virtual void **limpiar\_autores** () throw ()
- virtual unsigned short **obtener\_autores** () throw ()

### Atributos protegidos

- std::string \* **autores**
- unsigned short **cantidadAutores**
- [Idioma](#) **idioma**
- [Anio](#) **anioGrabacion**
- string **titulo**
- string **letra**

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

- src/CapaLogica/ManejoArchivos/RegistroCancion.h
- src/CapaLogica/ManejoArchivos/RegistroCancion.cpp

### 3.38. Referencia de la Clase RegistroClave

Diagrama de herencias de RegistroClave

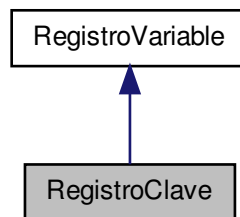
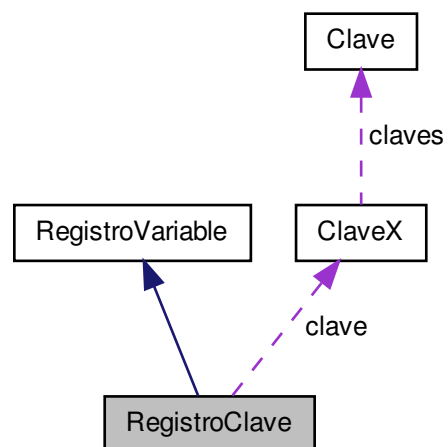


Diagrama de colaboración para RegistroClave:



#### 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:

- src/CapaLogica/ManejoArchivos/RegistroClave.h
- src/CapaLogica/ManejoArchivos/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** ()

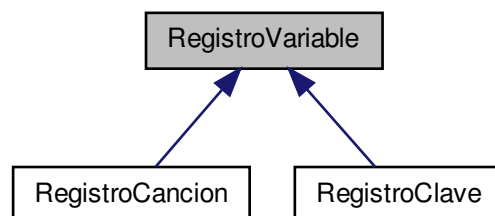


La documentación para esta clase fue generada a partir de los siguientes archivos:

- src/CapaFisica/RegistroFijo.h
- src/CapaFisica/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:

- src/CapaFisica/RegistroVariable.h
- src/CapaFisica/RegistroVariable.cpp

## 3.41. Referencia de la Clase SortExterno

### Métodos públicos

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

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

- src/Utilitarios/SortExterno.h
- src/Utilitarios/SortExterno.cpp

## 3.42. 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:

- src/CapaLogica/HashingExtensible/Tabla.h
- src/CapaLogica/HashingExtensible/Tabla.cpp

### 3.43. 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:

- src/CapaLogica/ManejoArchivos/Texto.h
- src/CapaLogica/ManejoArchivos/Texto.cpp