

My Project

Generated by Doxygen 1.8.12

Contents

1	Todo List	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	conjunto< T, CMP > Class Template Reference	7
4.1.1	Detailed Description	9
4.1.2	Constructor & Destructor Documentation	9
4.1.2.1	conjunto() [1/2]	9
4.1.2.2	conjunto() [2/2]	9
4.1.3	Member Function Documentation	10
4.1.3.1	begin()	10
4.1.3.2	cbegin()	10
4.1.3.3	cend()	11
4.1.3.4	clear()	11
4.1.3.5	count()	11
4.1.3.6	empty()	13
4.1.3.7	end()	13
4.1.3.8	erase() [1/2]	13
4.1.3.9	erase() [2/2]	14
4.1.3.10	find()	14

4.1.3.11	ibegin()	15
4.1.3.12	iend()	15
4.1.3.13	insert()	15
4.1.3.14	lower_bound()	16
4.1.3.15	operator=()	17
4.1.3.16	sbegin()	17
4.1.3.17	send()	18
4.1.3.18	size()	18
4.1.3.19	upper_bound()	18
4.2	conjunto< T, CMP >::const_impar_iterator Class Reference	19
4.3	conjunto< T, CMP >::const_iterator Class Reference	20
4.4	conjunto< T, CMP >::const_secure_iterator Class Reference	20
4.5	enfermedad Class Reference	21
4.5.1	Detailed Description	21
4.5.2	Constructor & Destructor Documentation	22
4.5.2.1	enfermedad() [1/2]	22
4.5.2.2	enfermedad() [2/2]	22
4.5.3	Member Function Documentation	22
4.5.3.1	getDatabase()	22
4.5.3.2	getID()	22
4.5.3.3	getName()	23
4.5.3.4	nameContains()	23
4.5.3.5	operator!=(())	23
4.5.3.6	operator<()	23
4.5.3.7	operator=()	24
4.5.3.8	operator==(())	24
4.5.3.9	setDatabase()	24
4.5.3.10	setID()	25
4.5.3.11	setName()	25
4.5.3.12	toString()	25

4.5.4	Friends And Related Function Documentation	25
4.5.4.1	operator<<	25
4.6	conjunto< T, CMP >::impar_iterator Class Reference	26
4.7	conjunto< T, CMP >::iterator Class Reference	26
4.8	mutacion Class Reference	27
4.8.1	Detailed Description	28
4.8.2	Constructor & Destructor Documentation	28
4.8.2.1	mutacion() [1/3]	28
4.8.2.2	mutacion() [2/3]	28
4.8.2.3	mutacion() [3/3]	29
4.8.3	Member Function Documentation	29
4.8.3.1	delSpaces()	29
4.8.3.2	getCaf()	29
4.8.3.3	getChr()	30
4.8.3.4	getCInsig()	30
4.8.3.5	getCommon()	30
4.8.3.6	getEnfermedades()	30
4.8.3.7	getGenes()	31
4.8.3.8	getID()	31
4.8.3.9	getPos()	31
4.8.3.10	getRef_alt()	31
4.8.3.11	operator<()	31
4.8.3.12	operator=()	32
4.8.3.13	operator==()	32
4.8.3.14	operator>()	32
4.8.3.15	setCaf()	33
4.8.3.16	setChr()	33
4.8.3.17	setCInsig()	33
4.8.3.18	setCommon()	33
4.8.3.19	setEnfermedades()	34
4.8.3.20	setGenes()	34
4.8.3.21	setID()	34
4.8.3.22	setPos()	34
4.8.3.23	setRef_alt()	35
4.8.4	Friends And Related Function Documentation	35
4.8.4.1	operator<<	35
4.9	conjunto< T, CMP >::secure_iterator Class Reference	35

5	File Documentation	37
5.1	enfermedad.h File Reference	37
5.1.1	Detailed Description	37
5.2	mutacion.h File Reference	37
5.2.1	Detailed Description	37
	Index	39

Chapter 1

Todo List

Class `conjunto< T, CMP >`

Implementa esta clase siguiendo la especificación asociada

Class `enfermedad`

Implementa esta clase, junto con su documentación asociada

Class `mutacion`

Implementa esta clase, junto con su documentación asociada

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<code>conjunto< T, CMP ></code>	
Clase conjunto	7
<code>conjunto< T, CMP >::const_impar_iterator</code>	19
<code>conjunto< T, CMP >::const_iterator</code>	20
<code>conjunto< T, CMP >::const_secure_iterator</code>	20
<code>enfermedad</code>	
Clase enfermedad, asociada al TDA enfermedad	21
<code>conjunto< T, CMP >::impar_iterator</code>	26
<code>conjunto< T, CMP >::iterator</code>	26
<code>mutacion</code>	
Clase mutacion, asociada a la definición de una mutación/SNP	27
<code>conjunto< T, CMP >::secure_iterator</code>	35

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

conjunto.h	??
conjunto.hxx	??
enfermedad.h	
Clase Enfermedad	37
enfermedad.hxx	??
mutacion.h	
Clase Mutacion	37
mutacion.hxx	??

Chapter 4

Class Documentation

4.1 conjunto< T, CMP > Class Template Reference

Clase conjunto.

```
#include <conjunto.h>
```

Classes

- class [const_impar_iterator](#)
- class [const_iterator](#)
- class [const_secure_iterator](#)
- class [impar_iterator](#)
- class [iterator](#)
- class [secure_iterator](#)

Public Types

- typedef T **value_type**
- typedef unsigned int **size_type**

Public Member Functions

- [conjunto](#) ()
constructor primitivo.
- [conjunto](#) (const [conjunto](#)< T, CMP > &d)
constructor de copia
- [iterator](#) [find](#) (const value_type &s)
busca una entrada en el conjunto
- [const_iterator](#) [find](#) (const value_type &s) const
- size_type [count](#) (const value_type &e) const
cuenta cuantas entradas coinciden con los parámetros dados.
- pair< [iterator](#), bool > [insert](#) (const value_type &val)
Inserta una entrada en el conjunto.

- **iterator erase** (const **iterator** position)
Borra una entrada en el conjunto . Busca la entrada y si la encuentra la borra.
- **size_type erase** (const value_type &val)
Borra una entrada en el conjunto . Busca la entrada y si la encuentra la borra.
- void **clear** ()
Borra todas las entradas del conjunto, dejandolo vacio.
- **size_type size** () const
numero de entradas en el conjunto
- bool **empty** () const
Chequea si el conjunto esta vacio (size()==0)
- **conjunto**< T, CMP > & **operator=** (const **conjunto** &org)
operador de asignación
- **conjunto::iterator begin** ()
begin del conjunto
- **conjunto::iterator end** ()
end del conjunto
- **conjunto::const_iterator cbegin** () const
begin del conjunto
- **conjunto::const_iterator cend** () const
end del conjunto
- **secure_iterator sbegin** ()
begin del conjunto
- **const_secure_iterator csbegin** () const
- **secure_iterator send** ()
end del conjunto
- **const_secure_iterator csend** () const
- **impar_iterator ibegin** ()
begin del conjunto
- **const_impar_iterator cibegin** () const
- **impar_iterator iend** ()
end del conjunto
- **const_impar_iterator ciend** () const
- **iterator lower_bound** (const value_type &val)
busca primer elemento por debajo ('antes', '<') de los parámetros dados.
- **const_iterator lower_bound** (const value_type &val) const
- **iterator upper_bound** (const value_type &val)
busca primer elemento por encima ('después', '>') de los parámetros dados.
- **const_iterator upper_bound** (const value_type &val) const

Friends

- class **impar_iterator**
- class **secure_iterator**
- class **iterator**
- class **const_impar_iterator**
- class **const_secure_iterator**
- class **const_iterator**

4.1.1 Detailed Description

```
template<typename T, typename CMP>
class conjunto< T, CMP >
```

Clase conjunto.

[conjunto<T,CMP>::conjunto](#), [find](#), [size](#), [insert](#), [erase](#),

Tipos [conjunto<T,CMP>::value_type](#), [conjunto<T,CMP>::size_type](#), [conjunto<T,CMP>::iterator](#), [conjunto<T,CMP>::const_iterator](#)

Descripción

Un conjunto es un contenedor que permite almacenar en orden creciente un conjunto de elementos no repetidos. En nuestro caso el conjunto va a tener un subconjunto restringido de métodos (inserción de elementos, consulta de un elemento, etc). Este conjunto "simulará" un conjunto de la stl, con algunas claras diferencias pues, entre otros, no estará dotado de la capacidad de iterar (recorrer) a través de sus elementos.

Asociado al conjunto, tendremos el tipo

```
conjunto::value_type
```

que permite hacer referencia al elemento almacenados en cada una de las posiciones del conjunto, en nuestro caso mutaciones (SNPs). Es requisito que el tipo `conjunto::value_type` tenga definida una relación de orden, `CMP`, y el operador de asignación, `operator=`.

El número de elementos en el conjunto puede variar dinámicamente; la gestión de la memoria es automática.

Todo Implementa esta clase siguiendo la especificación asociada

4.1.2 Constructor & Destructor Documentation

4.1.2.1 [conjunto\(\)](#) [1/2]

```
template<typename T , typename CMP >
conjunto< T, CMP >::conjunto ( )
```

constructor primitivo.

Constructor por defecto.

4.1.2.2 [conjunto\(\)](#) [2/2]

```
template<typename T, typename CMP>
conjunto< T, CMP >::conjunto (
    const conjunto< T, CMP > & d )
```

constructor de copia

Constructor de copia.

Parameters

in	<i>d</i>	conjunto a copiar
	<i>d</i>	conjunto para hacer la copia

4.1.3 Member Function Documentation**4.1.3.1 begin()**

```
template<typename T , typename CMP >  
conjunto< T, CMP >::iterator conjunto< T, CMP >::begin ( )
```

begin del conjunto

Devuelve begin del conjunto.

Returns

Devuelve un iterador al primer elemento del conjunto. Si no existe devuelve end

Postcondition

no modifica el conjunto.

4.1.3.2 cbegin()

```
template<typename T , typename CMP >  
conjunto< T, CMP >::const_iterator conjunto< T, CMP >::cbegin ( ) const
```

begin del conjunto

Devuelve const begin del conjunto.

Returns

Devuelve un iterador constante al primer elemento del conjunto. Si no existe devuelve cend

Postcondition

no modifica el conjunto.

4.1.3.3 cend()

```
template<typename T , typename CMP >
conjunto< T, CMP >::const_iterator conjunto< T, CMP >::cend ( ) const
```

end del conjunto

Devuelve const end del conjunto.

Returns

Devuelve un iterador constante al final del conjunto (posicion siguiente al ultimo).

Postcondition

no modifica el conjunto.

4.1.3.4 clear()

```
template<typename T , typename CMP >
void conjunto< T, CMP >::clear ( )
```

Borra todas las entradas del conjunto, dejandolo vacio.

Postcondition

El conjunto se modifica, quedando vacio.

4.1.3.5 count()

```
template<typename T , typename CMP >
conjunto< T, CMP >::size_type conjunto< T, CMP >::count (
    const value_type & e ) const
```

cuenta cuantas entradas coinciden con los parámetros dados.

Cuenta el numero de apariciones en vm de la mutacion e.

Parameters

in	e	entrada.
----	---	----------

Returns

Como el conjunto no puede tener entradas repetidas, devuelve 1 (si se encuentra la entrada) o 0 (si no se encuentra).

Postcondition

no modifica el conjunto.

Parameters

<i>e</i>	Mutacion a buscar
----------	-------------------

Returns

total contador con numero de apariciones

4.1.3.6 empty()

```
template<typename T , typename CMP >
bool conjunto< T, CMP >::empty ( ) const
```

Chequea si el conjunto esta vacio (`size()==0`)

Postcondition

No se modifica el conjunto.

4.1.3.7 end()

```
template<typename T , typename CMP >
conjunto< T, CMP >::iterator conjunto< T, CMP >::end ( )
```

end del conjunto

Devuelve end del conjunto.

Returns

Devuelve un iterador al final del conjunto (posicion siguiente al ultimo).

Postcondition

no modifica el conjunto.

4.1.3.8 erase() [1/2]

```
template<typename T , typename CMP >
conjunto< T, CMP >::iterator conjunto< T, CMP >::erase (
    const iterator posicion )
```

Borra una entrada en el conjunto . Busca la entrada y si la encuentra la borra.

Parameters

<i>in</i>	<i>val</i>	entrada a borrar.
<i>in</i>	<i>position</i>	iterador que apunta a la entrada que geremos borrar

Returns

devuelve la posicion siguiente al elemento borrado (para la version con iterador) o el numero de elementos borrados

Postcondition

Si esta en el conjunto su tamaño se decrementa en 1.

Parameters

in	<i>position</i>	iterador donde esta la entrada a borrar
----	-----------------	---

Postcondition

Si esta en el conjunto su tamaño se decrementa en 1.

4.1.3.9 erase() [2/2]

```
template<typename T , typename CMP >
conjunto< T, CMP >::size_type conjunto< T, CMP >::erase (
    const value_type & val )
```

Borra una entrada en el conjunto . Busca la entrada y si la encuentra la borra.

Parameters

in	<i>e</i>	entrada a borrar
----	----------	------------------

Postcondition

Si esta en el conjunto su tamaño se decrementa en 1.

4.1.3.10 find()

```
template<typename T , typename CMP >
conjunto< T, CMP >::iterator conjunto< T, CMP >::find (
    const value_type & s )
```

busca una entrada en el conjunto

Buscador de mutacion en vector de mutaciones.

Parameters

in	<i>s</i>	entrada a buscar.
----	----------	-------------------

Returns

Si existe una entrada en el conjunto con ese valor devuelve el iterador a su posicion, en caso contrario devuelve iterador al final de conjunto

Postcondition

no modifica el conjunto.

Parameters

s	Mutacion a buscar
---	-------------------

Returns

resul Par de datos con la mutacion y bool de si esta o no

4.1.3.11 ibegin()

```
template<typename T , typename CMP >
conjunto< T, CMP >::impar_iterator conjunto< T, CMP >::ibegin ( )
```

begin del conjunto

Returns

Devuelve un iterador impar (o iterador impar constante, respectivamente) al primer elemento (de posición impar) del conjunto. Si no existe devuelve end

4.1.3.12 iend()

```
template<typename T , typename CMP >
conjunto< T, CMP >::impar_iterator conjunto< T, CMP >::iend ( )
```

end del conjunto

Returns

Devuelve un iterador impar (o iterador impar constante, respectivamente) al final del conjunto (posicion siguiente al ultimo).

4.1.3.13 insert()

```
template<typename T , typename CMP >
pair< typename conjunto< T, CMP >::iterator, bool > conjunto< T, CMP >::insert (
    const value_type & e )
```

Inserta una entrada en el conjunto.

Parameters

<i>val</i>	entrada a insertar
------------	--------------------

Returns

un par donde el segundo campo vale true si la entrada se ha podido insertar con éxito, esto es, no existe una mutación con igual valor en el conjunto. False en caso contrario. El primer campo del par devuelve un iterador al elemento insertado, o `end()` si no fue posible la insercion

Postcondition

Si e no esta en el conjunto, el `size()` sera incrementado en 1.

Parameters

<i>e</i>	entrada a insertar
----------	--------------------

Returns

pair con true si la entrada se ha podido insertar con éxito, \ esto es, no existe una mutación con igual par chr/pos ni igual ID en el conjunto. False en caso contrario.

Postcondition

Si e no esta en el conjunto, el `size()` sera incrementado en 1.

4.1.3.14 lower_bound()

```
template<typename T , typename CMP >
conjunto< T, CMP >::iterator conjunto< T, CMP >::lower_bound (
    const value_type & val )
```

busca primer elemento por debajo ('antes', '<') de los parámetros dados.

cuenta cuantas entradas hay en el conjunto por debajo ('antes', '<') de los parámetros dados.

Parameters

<i>in</i>	<i>val</i>	entrada.
-----------	------------	----------

Returns

Devuelve un iterador al primer elemento que cumple que "elemento<e" es falso, esto es, el primer elemento que es mayor o igual que val

Si no existe devuelve end

Postcondition

no modifica el conjunto.

Parameters

in	e	entrada.
----	---	----------

Returns

Devuelve un iterador al primer elemento que cumple que "elemento<e" es falso, esto es, el primer elemento que es mayor o igual que e

Postcondition

no modifica el conjunto.

4.1.3.15 operator=()

```
template<typename T , typename CMP >
conjunto< T, CMP > & conjunto< T, CMP >::operator= (
    const conjunto< T, CMP > & org )
```

operador de asignación

Parameters

in	org	conjunto a copiar.
----	-----	--------------------

Returns

Crea y devuelve un conjunto duplicado exacto de org.

4.1.3.16 sbegin()

```
template<typename T , typename CMP >
conjunto< T, CMP >::secure_iterator conjunto< T, CMP >::sbegin ( )
```

begin del conjunto

Returns

Devuelve un iterador seguro (o iterador seguro constante, respectivamente) al primer elemento del conjunto.
Si no existe devuelve end

4.1.3.17 send()

```
template<typename T , typename CMP >
conjunto< T, CMP >::secure_iterator conjunto< T, CMP >::send ( )
```

end del conjunto

Returns

Devuelve un iterador seguro (o iterador seguro constante, respectivamente) al final del conjunto (posicion siguiente al ultimo).

Postcondition

no modifica el conjunto.

4.1.3.18 size()

```
template<typename T , typename CMP >
conjunto< T, CMP >::size_type conjunto< T, CMP >::size ( ) const
```

numero de entradas en el conjunto

Postcondition

No se modifica el conjunto.

Returns

numero de entradas en el conjunto

Postcondition

No se modifica el conjunto.

Returns

numero de entradas en el conjunto Devolver size_type

4.1.3.19 upper_bound()

```
template<typename T , typename CMP >
conjunto< T, CMP >::iterator conjunto< T, CMP >::upper_bound (
    const value_type & val )
```

busca primer elemento por encima ('después', '>') de los parámetros dados.

cuenta cuantas entradas hay en el conjunto por encima ('después', '>') de los parámetros dados.

Parameters

<i>in</i>	<i>val</i>	entrada. Devuelve un iterador al primer elemento que cumple que "elemento>e", esto es, el primer elemento ESTRICTAMENTE mayor que val
-----------	------------	---

Si no existe devuelve end

Postcondition

no modifica el conjunto.

Parameters

<i>in</i>	<i>e</i>	entrada. Devuelve un iterador al primer elemento que cumple que "elemento>e", esto es, el primer elemento ESTRICTAMENTE mayor que e
-----------	----------	---

Postcondition

no modifica el conjunto.

The documentation for this class was generated from the following files:

- conjunto.h
- conjunto.hxx

4.2 conjunto< T, CMP >::const_impar_iterator Class Reference

Public Member Functions

- **const_impar_iterator** (const [const_impar_iterator](#) &x)
- const T & **operator*** ()
- [const_impar_iterator](#) & **operator++** ()
- [const_impar_iterator](#) **operator++** (int i)
- bool **operator==** (const [const_impar_iterator](#) &x) const
- bool **operator!=** (const [const_impar_iterator](#) &x) const
- [const_impar_iterator](#) & **operator=** (const [const_impar_iterator](#) &x)

Friends

- class **conjunto**< T, CMP >

The documentation for this class was generated from the following files:

- conjunto.h
- conjunto.hxx

4.3 conjunto< T, CMP >::const_iterator Class Reference

Public Member Functions

- **const_iterator** (const [const_iterator](#) &x)
- **const_iterator** (const [secure_iterator](#) &x)
- const T & **operator*** ()
- [const_iterator](#) & **operator++** ()
- [const_iterator](#) **operator++** (int i)
- [const_iterator](#) & **operator--** ()
- [const_iterator](#) **operator--** (int i)
- bool **operator==** (const [const_iterator](#) &x) const
- bool **operator!=** (const [const_iterator](#) &x) const
- [const_iterator](#) & **operator=** (const [const_iterator](#) &x)

Friends

- class **conjunto**< T, CMP >

The documentation for this class was generated from the following files:

- conjunto.h
- conjunto.hxx

4.4 conjunto< T, CMP >::const_secure_iterator Class Reference

Public Member Functions

- **const_secure_iterator** (const [const_secure_iterator](#) &x)
- const T & **operator*** ()
- [const_secure_iterator](#) & **operator++** ()
- [const_secure_iterator](#) **operator++** (int i)
- [const_secure_iterator](#) & **operator--** ()
- [const_secure_iterator](#) **operator--** (int i)
- bool **operator==** (const [const_secure_iterator](#) &x) const
- bool **operator!=** (const [const_secure_iterator](#) &x) const
- [const_secure_iterator](#) & **operator=** (const [const_secure_iterator](#) &x)

Friends

- class **conjunto**< T, CMP >

The documentation for this class was generated from the following files:

- conjunto.h
- conjunto.hxx

4.5 enfermedad Class Reference

Clase enfermedad, asociada al TDA enfermedad.

```
#include <enfermedad.h>
```

Public Member Functions

- [enfermedad](#) ()
Construye una enfermedad vacia.
- [enfermedad](#) (const string &name, const string &ID, const string &database)
Constructor por parametros.
- void [setName](#) (const string &name)
Asignar nuevo nombre a la enfermedad.
- void [setID](#) (const string &ID)
Asignar nuevo ID a la enfermedad.
- void [setDatabase](#) (const string &database)
Asignar nuevo database a la enfermedad.
- string [getName](#) () const
Devuelve el nombre de la enfermedad.
- string [getID](#) () const
Devuelve el ID de la enfermedad.
- string [getDatabase](#) () const
Devuelve la database de la enfermedad.
- [enfermedad](#) & [operator=](#) (const [enfermedad](#) &e)
Operator =.
- string [toString](#) () const
Devuelve un string con los datos de la enfermedad.
- bool [operator==](#) (const [enfermedad](#) &e) const
Operator ==.
- bool [operator!=](#) (const [enfermedad](#) &e) const
Operator !=.
- bool [operator<](#) (const [enfermedad](#) &e) const
Operator <.
- bool [nameContains](#) (const string &str) const
Comprueba si el nombre de la enfermedad contiene el string.

Friends

- ostream & [operator<<](#) (ostream &os, const [enfermedad](#) &e)
Operator <<.

4.5.1 Detailed Description

Clase enfermedad, asociada al TDA enfermedad.

[enfermedad::enfermedad](#), Descripción contiene toda la información asociada a una enfermedad almacenada en la BD ClinVar-dbSNP (nombre de la enfermedad, id, BD que provee el id)

Todo Implementa esta clase, junto con su documentación asociada

Clase enfermedad con los atributos propios de la enfermedad

4.5.2 Constructor & Destructor Documentation

4.5.2.1 enfermedad() [1/2]

```
enfermedad::enfermedad ( )
```

Construye una enfermedad vacia.

fichero de implementacion de la clase enfermedad

4.5.2.2 enfermedad() [2/2]

```
enfermedad::enfermedad (
    const string & name,
    const string & ID,
    const string & database )
```

Constructor por parametros.

Parameters

<i>name</i>	nombre de la enfermedad
<i>ID</i>	de la enfermedad
<i>database</i>	de la enfermedad

4.5.3 Member Function Documentation

4.5.3.1 getDatabase()

```
string enfermedad::getDatabase ( ) const
```

Devuelve la database de la enfermedad.

Returns

database

4.5.3.2 getID()

```
string enfermedad::getID ( ) const
```

Devuelve el ID de la enfermedad.

Returns

ID

4.5.3.3 getName()

```
string enfermedad::getName ( ) const
```

Devuelve el nombre de la enfermedad.

Returns

name

4.5.3.4 nameContains()

```
bool enfermedad::nameContains (
    const string & str ) const
```

Comprueba si el nombre de la enfermedad contiene el string.

Parameters

<i>str</i>	string para buscar
------------	--------------------

Returns

true o false

Busca el string dentro del string de campo name

4.5.3.5 operator!=(())

```
bool enfermedad::operator!= (
    const enfermedad & e ) const
```

Operator !=.

Parameters

<i>e</i>	Enfermedad orig para comparar
----------	-------------------------------

Returns

true o false

Compara enfermedad this con la de origen para ver si son distintas

4.5.3.6 operator<()

```
bool enfermedad::operator< (
    const enfermedad & e ) const
```

Operator <.

Parameters

<i>e</i>	Enfermedad orig para comparar
----------	-------------------------------

Returns

true o false

Compara enfermedad this con la de origen para ver si this es menor que orig, orden alfabetico por campo name

4.5.3.7 operator=()

```
enfermedad & enfermedad::operator= (
    const enfermedad & e )
```

Operator =.

Parameters

<i>e</i>	enfermedad origen para hacer la asignacion
----------	--

Hace una asignacion de la enfermedad origen a this, realizando las comprobaciones necesarias

4.5.3.8 operator==()

```
bool enfermedad::operator== (
    const enfermedad & e ) const
```

Operator ==.

Parameters

<i>e</i>	Enfermedad orig para comparar
----------	-------------------------------

Returns

true o false

Compara enfermedad this con la de origen para ver si son iguales

4.5.3.9 setDatabase()

```
void enfermedad::setDatabase (
    const string & database )
```

Asignar nuevo database a la enfermedad.

Parameters

<i>database</i>	Nuevo database
-----------------	----------------

4.5.3.10 setID()

```
void enfermedad::setID (
    const string & ID )
```

Asignar nuevo ID a la enfermedad.

Parameters

<i>ID</i>	Nuevo ID
-----------	----------

4.5.3.11 setName()

```
void enfermedad::setName (
    const string & name )
```

Asignar nuevo nombre a la enfermedad.

Parameters

<i>name</i>	Nuevo nombre
-------------	--------------

4.5.3.12 toString()

```
string enfermedad::toString ( ) const
```

Devuelve un string con los datos de la enfermedad.

Returns

string con los datos de la enfermedad separados por espacios

4.5.4 Friends And Related Function Documentation

4.5.4.1 operator<<

```
ostream& operator<< (
    ostream & os,
    const enfermedad & e ) [friend]
```

Operator <<.

Parameters

<i>os</i>	Ostream
<i>e</i>	Enfermedad this para imprimir

Permite imprimir por pantalla una enfermedad

The documentation for this class was generated from the following files:

- [enfermedad.h](#)
- [enfermedad.hxx](#)

4.6 conjunto< T, CMP >::impar_iterator Class Reference

Public Member Functions

- **impar_iterator** (const [impar_iterator](#) &x)
- const T & **operator*** ()
- [impar_iterator](#) & **operator++** ()
- [impar_iterator](#) **operator++** (int i)
- bool **operator==** (const [impar_iterator](#) &x) const
- bool **operator!=** (const [impar_iterator](#) &x) const
- [impar_iterator](#) & **operator=** (const [impar_iterator](#) &x)

Friends

- class **conjunto**< T, CMP >

The documentation for this class was generated from the following files:

- [conjunto.h](#)
- [conjunto.hxx](#)

4.7 conjunto< T, CMP >::iterator Class Reference

Public Member Functions

- **iterator** (const [iterator](#) &x)
- **iterator** (const [secure_iterator](#) &x)
- const T & **operator*** ()
- [iterator](#) & **operator++** ()
- [iterator](#) **operator++** (int i)
- [iterator](#) & **operator--** ()
- [iterator](#) **operator--** (int i)
- bool **operator==** (const [iterator](#) &x) const
- bool **operator!=** (const [iterator](#) &x) const
- [iterator](#) & **operator=** (const [iterator](#) &x)

Friends

- class **conjunto**< **T**, **CMP** >

The documentation for this class was generated from the following files:

- conjunto.h
- conjunto.hxx

4.8 mutacion Class Reference

Clase mutacion, asociada a la definición de una mutación/SNP.

```
#include <mutacion.h>
```

Public Member Functions

- [mutacion](#) ()
Crea una mutacion vacia inicializando los atributos.
- [mutacion](#) (const [mutacion](#) &m)
Construye una mutacion como una copia de otra.
- [mutacion](#) (const string &str)
Construye una mutacion a partir de un string.
- string [delSpaces](#) (string &input)
Funcion auxiliar para eliminar espacios.
- void [setID](#) (const string &id)
Establecer un ID a la mutacion.
- void [setChr](#) (const string &chr)
Establecer un chr (Numero de cromosoma) a la mutacion.
- void [setPos](#) (const unsigned int &pos)
Establecer una nueva posicion.
- void [setRef_alt](#) (const std::vector< string > &ref_alt)
Establecer nuevas bases (ref_alt)
- void [setGenes](#) (const std::vector< string > &genes)
Establecer nuevos genes.
- void [setCommon](#) (const bool &common)
Dar nuevo valor a common.
- void [setCaf](#) (const std::vector< float > &caf)
Nuevo caf a establecer.
- void [setEnfermedades](#) (const std::vector< [enfermedad](#) > &enfermedades)
Nuevas enfermedades a establecer.
- void [setClnsig](#) (const std::vector< int > &clnsig)
Establecer un nuevo setClnsig.
- string [getID](#) () const
Devuelve el ID.
- string [getChr](#) () const
Devuelve los cromosomas.
- unsigned int [getPos](#) () const

- Devuelve la posicion.*
- `const std::vector< string > & getRef_alt () const`
- Devuelve las bases.*
- `const std::vector< string > & getGenes () const`
- Devuelve los genes.*
- `bool getCommon () const`
- Devuelve Common.*
- `const std::vector< float > & getCaf () const`
- Devuelve caf.*
- `const std::vector< enfermedad > & getEnfermedades () const`
- Devuelve las enfermedades.*
- `const std::vector< int > & getClnsig () const`
- Devuelve Clnsig.*
- `mutacion & operator= (const mutacion &m)`
- Operator =.*
- `bool operator== (const mutacion &m) const`
- Operator ==.*
- `bool operator< (const mutacion &m) const`
- Operator <.*
- `bool operator> (const mutacion &m) const`
- Operator >*

Friends

- `ostream & operator<< (ostream &os, const mutacion &m)`
- Operator <<.*

4.8.1 Detailed Description

Clase mutacion, asociada a la definici3n de una mutaci3n/SNP.

`mutacion::mutacion`,

Todo Implementa esta clase, junto con su documentaci3n asociada

Crea una mutacion con todos los datos identificadores de dicha mutacion

4.8.2 Constructor & Destructor Documentation

4.8.2.1 `mutacion()` [1/3]

```
mutacion::mutacion ( )
```

Crea una mutacion vacia inicializando los atributos.

Constructor por defecto.

4.8.2.2 `mutacion()` [2/3]

```
mutacion::mutacion (
    const mutacion & m )
```

Construye una mutacion como una copia de otra.

Constructor de copia.

Parameters

<i>m</i>	Mutacion a copiar
----------	-------------------

4.8.2.3 `mutacion()` [3/3]

```
mutacion::mutacion (
    const string & str )
```

Construye una mutacion a partir de un string.

Constructor por un string.

Parameters

<i>str</i>	String del que se extrae la informacion para construir la mutacion
------------	--

Se van extrayendo los valores avanzando un indice de lectura sobre el string, y leyendo en orden cada valor \

ORDEN DE APARICION: \ 1º: chr - numero de cromosoma (1-23) \ 2º: pos - posicion, numero largo, empieza en 1 \ 3º: ID - rsXXXX, X numeros \ 4º: ref_alt - Bases, desde que acaba ID hasta RS mayus \ 5º: genes - Genes, GENEINFO (buscar) hasta el final (;) \ 6º: common - COMMON, buscar, valores 0 o 1 o no aparecer (poner a 0) \ 7º: clnsig - CLNSIG, buscar (num) y hasta final (;) \ 8º: caf - CAF, buscar num y hasta (;), sino poner a 0 \ 9º: enfermedades - Buscar NOMBRE: name CLNDBN, ID: ID CLNDSDBID, DATABASE database CLNDSDB Con las 3 se tiene una enfermedad \

Parameters

<i>str</i>	string con la informacion para inicializar
------------	--

4.8.3 Member Function Documentation

4.8.3.1 `delSpaces()`

```
string mutacion::delSpaces (
    string & input )
```

Funcion auxiliar para eliminar espacios.

Parameters

<i>input</i>	String del que se van a eliminar los espacios
--------------	---

4.8.3.2 `getCaf()`

```
const std::vector< float > & mutacion::getCaf ( ) const
```

Devuelve caf.

Returns

caf

4.8.3.3 getChr()

```
string mutacion::getChr ( ) const
```

Devuelve los cromosomas.

Returns

chr

4.8.3.4 getClnsig()

```
const std::vector< int > & mutacion::getClnsig ( ) const
```

Devuelve Clnsig.

Returns

Clnsig

4.8.3.5 getCommon()

```
bool mutacion::getCommon ( ) const
```

Devuelve Common.

Returns

common

4.8.3.6 getEnfermedades()

```
const std::vector< enfermedad > & mutacion::getEnfermedades ( ) const
```

Devuelve las enfermedades.

Returns

enfermedades

4.8.3.7 getGenes()

```
const std::vector< string > & mutacion::getGenes ( ) const
```

Devuelve los genes.

Returns

genes

4.8.3.8 getID()

```
string mutacion::getID ( ) const
```

Devuelve el ID.

Returns

el ID

4.8.3.9 getPos()

```
unsigned int mutacion::getPos ( ) const
```

Devuelve la posicion.

Returns

posicion

4.8.3.10 getRef_alt()

```
const std::vector< string > & mutacion::getRef_alt ( ) const
```

Devuelve las bases.

Returns

bases (ref_alt)

4.8.3.11 operator<()

```
bool mutacion::operator< (
    const mutacion & m ) const
```

Operator <.

Parameters

<i>m</i>	mutacion origen para comparar con this
----------	--

Returns

bool true o false

Compara por numero de cromosoma que la mutacion orig sea mayor que this, en caso de igual numero de cromosoma comprueba la posicion

4.8.3.12 operator=()

```
mutacion & mutacion::operator= (
    const mutacion & m )
```

Operator =.

Parameters

<i>m</i>	mutacion para asignar a this
----------	------------------------------

Returns

*this

Asigna la mutacion origen a this, haciendo las comprobaciones necesarias

4.8.3.13 operator==()

```
bool mutacion::operator== (
    const mutacion & m ) const
```

Operator ==.

Parameters

<i>m</i>	mutacion para comparar con this
----------	---------------------------------

Returns

bool true o false

Compara this con la mutacion origen (m) para determinar si son iguales o no

4.8.3.14 operator>()

```
bool mutacion::operator> (
    const mutacion & m ) const
```

Operator >

Parameters

<i>m</i>	mutacion a comparar
----------	---------------------

Returns

Resultado de la comparacion Compara que sea menor y lo niega

4.8.3.15 setCaf()

```
void mutacion::setCaf (
    const std::vector< float > & caf )
```

Nuevo caf a establecer.

Parameters

<i>caf</i>	Vector de float con el nuevo caf
------------	----------------------------------

4.8.3.16 setChr()

```
void mutacion::setChr (
    const string & chr )
```

Establecer un chr (Numero de cromosoma) a la mutacion.

Parameters

<i>chr</i>	Nuevo numero de cromosoma a establecer
------------	--

4.8.3.17 setClnsig()

```
void mutacion::setClnsig (
    const std::vector< int > & clnsig )
```

Establecer un nuevo setClnsig.

Parameters

<i>clnsig</i>	Vector de enteros con los clnsig a establecer
---------------	---

4.8.3.18 setCommon()

```
void mutacion::setCommon (
```

```
const bool & common )
```

Dar nuevo valor a common.

Parameters

<i>common</i>	Con el nuevo valor true o false para common de la clase
---------------	---

4.8.3.19 setEnfermedades()

```
void mutacion::setEnfermedades (
    const std::vector< enfermedad > & enfermedades )
```

Nuevas enfermedades a establecer.

Parameters

<i>enfermedades</i>	Vector de enfermedades con las nnuevas enfermedades
---------------------	---

4.8.3.20 setGenes()

```
void mutacion::setGenes (
    const std::vector< string > & genes )
```

Establecer nuevos genes.

Parameters

<i>genes</i>	Vector string con los nuevos genes a establecer
--------------	---

4.8.3.21 setID()

```
void mutacion::setID (
    const string & id )
```

Establecer un ID a la mutacion.

Parameters

<i>id</i>	String con el nuevo ID a establecer
-----------	-------------------------------------

4.8.3.22 setPos()

```
void mutacion::setPos (
    const unsigned int & pos )
```

Establecer una nueva posicion.

Parameters

<i>pos</i>	Posicion a establecer
------------	-----------------------

4.8.3.23 setRef_alt()

```
void mutacion::setRef_alt (
    const std::vector< string > & ref_alt )
```

Establecer nuevas bases (ref_alt)

Parameters

<i>ref_alt</i>	Nuevas bases a establecer
----------------	---------------------------

4.8.4 Friends And Related Function Documentation

4.8.4.1 operator<<

```
ostream& operator<< (
    ostream & os,
    const mutacion & m ) [friend]
```

Operator <<.

Parameters

<i>os</i>	argumento ostream
<i>m</i>	mutacion this

Returns

os

Permite imprimir por pantalla una mutacion

The documentation for this class was generated from the following files:

- [mutacion.h](#)
- [mutacion.hxx](#)

4.9 conjunto< T, CMP >::secure_iterator Class Reference

Public Member Functions

- **secure_iterator** (const [secure_iterator](#) &x)

- `const T & operator* ()`
- `secure_iterator & operator++ ()`
- `secure_iterator operator++ (int i)`
- `secure_iterator & operator-- ()`
- `secure_iterator operator-- (int i)`
- `bool operator== (const secure_iterator &x) const`
- `bool operator!= (const secure_iterator &x) const`
- `secure_iterator & operator= (const secure_iterator &x)`

Friends

- `class conjunto< T, CMP >`

The documentation for this class was generated from the following files:

- `conjunto.h`
- `conjunto.hxx`

Chapter 5

File Documentation

5.1 enfermedad.h File Reference

Clase Enfermedad.

```
#include <string>
#include <iostream>
#include "enfermedad.hxx"
```

Classes

- class [enfermedad](#)

Clase enfermedad, asociada al TDA enfermedad.

5.1.1 Detailed Description

Clase Enfermedad.

5.2 mutacion.h File Reference

Clase Mutacion.

```
#include <string>
#include <iostream>
#include <vector>
#include <cctype>
#include "enfermedad.h"
#include "mutacion.hxx"
```

Classes

- class [mutacion](#)

Clase mutacion, asociada a la definición de una mutación/SNP.

5.2.1 Detailed Description

Clase Mutacion.

Index

- begin
 - conjunto, [10](#)
- cbegin
 - conjunto, [10](#)
- cend
 - conjunto, [10](#)
- clear
 - conjunto, [11](#)
- conjunto
 - begin, [10](#)
 - cbegin, [10](#)
 - cend, [10](#)
 - clear, [11](#)
 - conjunto, [9](#)
 - count, [11](#)
 - empty, [13](#)
 - end, [13](#)
 - erase, [13, 14](#)
 - find, [14](#)
 - ibegin, [15](#)
 - iend, [15](#)
 - insert, [15](#)
 - lower_bound, [16](#)
 - operator=, [17](#)
 - sbegin, [17](#)
 - send, [17](#)
 - size, [18](#)
 - upper_bound, [18](#)
- conjunto< T, CMP >, [7](#)
- conjunto< T, CMP >::const_impar_iterator, [19](#)
- conjunto< T, CMP >::const_iterator, [20](#)
- conjunto< T, CMP >::const_secure_iterator, [20](#)
- conjunto< T, CMP >::impar_iterator, [26](#)
- conjunto< T, CMP >::iterator, [26](#)
- conjunto< T, CMP >::secure_iterator, [35](#)
- count
 - conjunto, [11](#)
- delSpaces
 - mutacion, [29](#)
- empty
 - conjunto, [13](#)
- end
 - conjunto, [13](#)
- enfermedad, [21](#)
 - enfermedad, [22](#)
 - getDatabase, [22](#)
 - getID, [22](#)
 - getName, [22](#)
 - nameContains, [23](#)
 - operator!=, [23](#)
 - operator<, [23](#)
 - operator<<, [25](#)
 - operator=, [24](#)
 - operator==, [24](#)
 - setDatabase, [24](#)
 - setID, [25](#)
 - setName, [25](#)
 - toString, [25](#)
- enfermedad.h, [37](#)
- erase
 - conjunto, [13, 14](#)
- find
 - conjunto, [14](#)
- getCaf
 - mutacion, [29](#)
- getChr
 - mutacion, [30](#)
- getClnsig
 - mutacion, [30](#)
- getCommon
 - mutacion, [30](#)
- getDatabase
 - enfermedad, [22](#)
- getEnfermedades
 - mutacion, [30](#)
- getGenes
 - mutacion, [30](#)
- getID
 - enfermedad, [22](#)
 - mutacion, [31](#)
- getName
 - enfermedad, [22](#)
- getPos
 - mutacion, [31](#)
- getRef_alt
 - mutacion, [31](#)
- ibegin
 - conjunto, [15](#)
- iend
 - conjunto, [15](#)
- insert
 - conjunto, [15](#)
- lower_bound

- conjunto, 16
- mutacion, 27
 - delSpaces, 29
 - getCaf, 29
 - getChr, 30
 - getClnsig, 30
 - getCommon, 30
 - getEnfermedades, 30
 - getGenes, 30
 - getID, 31
 - getPos, 31
 - getRef_alt, 31
 - mutacion, 28, 29
 - operator<, 31
 - operator<<, 35
 - operator>, 32
 - operator=, 32
 - operator==, 32
 - setCaf, 33
 - setChr, 33
 - setClnsig, 33
 - setCommon, 33
 - setEnfermedades, 34
 - setGenes, 34
 - setID, 34
 - setPos, 34
 - setRef_alt, 35
- mutacion.h, 37
- nameContains
 - enfermedad, 23
- operator!=
 - enfermedad, 23
- operator<
 - enfermedad, 23
 - mutacion, 31
- operator<<
 - enfermedad, 25
 - mutacion, 35
- operator>
 - mutacion, 32
- operator=
 - conjunto, 17
 - enfermedad, 24
 - mutacion, 32
- operator==
 - enfermedad, 24
 - mutacion, 32
- sbegin
 - conjunto, 17
- send
 - conjunto, 17
- setCaf
 - mutacion, 33
- setChr
 - mutacion, 33
- setClnsig
 - mutacion, 33
- setCommon
 - mutacion, 33
- setDatabase
 - enfermedad, 24
- setEnfermedades
 - mutacion, 34
- setGenes
 - mutacion, 34
- setID
 - enfermedad, 25
 - mutacion, 34
- setName
 - enfermedad, 25
- setPos
 - mutacion, 34
- setRef_alt
 - mutacion, 35
- size
 - conjunto, 18
- toString
 - enfermedad, 25
- upper_bound
 - conjunto, 18