

(Gestión de intervalos)

0

Generado por Doxygen 1.8.11

Índice general

1	Índice de clases	1
1.1	Lista de clases	1
2	Documentación de las clases	3
2.1	Referencia de la Clase Intervalo	3
2.1.1	Descripción detallada	3
2.1.2	Documentación del constructor y destructor	3
2.1.2.1	Intervalo(double cotaInferior, double cotaSuperior)	3
2.1.2.2	Intervalo(double cotaInferior, double cotaSuperior, bool cerradoInferior, bool cerradoSuperior)	4
2.1.3	Documentación de las funciones miembro	4
2.1.3.1	dentroCotaInf() const	4
2.1.3.2	dentroCotaSup() const	5
2.1.3.3	estaDentro(double n) const	5
2.1.3.4	esVacio() const	6
2.1.3.5	getCotaInf() const	6
2.1.3.6	getCotaSup() const	6
2.1.3.7	setIntervalo(double cotaInferior, double cotaSuperior, bool cerradoInferior, bool cerradoSuperior)	6
	Índice	9

Capítulo 1

Índice de clases

1.1. Lista de clases

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

Intervalo	3
-------------------------------------	---

Capítulo 2

Documentación de las clases

2.1. Referencia de la Clase Intervalo

Métodos públicos

- **Intervalo** ()
Intervalo vacío por defecto $\text{valorInf} = \text{valorSup} \ \& \ \text{abiertoInf} + \text{abierto}$.
- **Intervalo** (double cotaInferior, double cotaSuperior)
Crea un Intervalo cerrado por defecto.
- **Intervalo** (double cotaInferior, double cotaSuperior, bool cerradoInferior, bool cerradoSuperior)
Crea Intervalo.
- double **getCotaInf** () const
Devuelve la cota inferior del intervalo.
- double **getCotaSup** () const
Devuelve la cota superior del intervalo.
- bool **dentroCotaInf** () const
Consulta si el intervalo es cerrado en su cota inferior.
- bool **dentroCotaSup** () const
Consulta si el intervalo es cerrado en su cota superior.
- void **setIntervalo** (double cotaInferior, double cotaSuperior, bool cerradoInferior, bool cerradoSuperior)
Define los valores del intervalo.
- bool **esVacio** () const
Consulta si el intervalo almacenado es vacío o no.
- bool **estaDentro** (double n) const
Consulta si un determinado valor está dentro del intervalo.

2.1.1. Descripción detallada

Definición en la línea 14 del archivo intervalo.cpp.

2.1.2. Documentación del constructor y destructor

2.1.2.1. Intervalo::Intervalo (double cotaInferior, double cotaSuperior)

Crea un **Intervalo** cerrado por defecto.

Parámetros

<i>cotaInferior</i>	
<i>cotaSuperior</i>	$cotaInferior \leq cotaSuperior$

Definición en la línea 121 del archivo intervalo.cpp.

```

121
122  assert (valido(cinf,csup, true, true));
123      cotaInf = cinf;
124      cotaSup = csup;
125      cerradoInf = true;
126      cerradoSup = true;
127 }
```

2.1.2.2. Intervalo::Intervalo (double *cotaInferior*, double *cotaSuperior*, bool *cerradoInferior*, bool *cerradoSuperior*)

Crea [Intervalo](#).

Parámetros

<i>cerradoInferior</i>	
<i>cerradoSuperior</i>	
<i>cotaInferior</i>	
<i>cotaSuperior</i>	$cotaInferior \leq cotaSuperior$

Definición en la línea 129 del archivo intervalo.cpp.

```

129
130  assert (valido(cinf, csup, cerrinf, cerrsup));
131      cotaInf = cinf;
132      cotaSup = csup;
133      cerradoInf = cerrinf;
134      cerradoSup = cerrsup;
135 }
```

2.1.3. Documentación de las funciones miembro

2.1.3.1. bool Intervalo::dentroCotaInf () const

Consulta si el intervalo es cerrado en su cota inferior.

Devuelve

Valores devueltos

<i>true</i>	si es cerrado
<i>false</i>	si es cerrado

Definición en la línea 145 del archivo intervalo.cpp.

```
145                                     {  
146     return cerradoInf;  
147 }
```

2.1.3.2. bool Intervalo::dentroCotaSup () const

Consulta si el intervalo es cerrado en su cota superior.

Devuelve

Valores devueltos

<i>true</i>	si es cerrado
<i>false</i>	si es cerrado

Definición en la línea 149 del archivo intervalo.cpp.

```
149                                     {  
150     return cerradoSup;  
151 }
```

2.1.3.3. bool Intervalo::estaDentro (double *n*) const

Consulta si un determinado valor está dentro del intervalo.

Parámetros

<i>n</i>	El valor consultado
----------	---------------------

Devuelve

Valores devueltos

<i>true</i>	si el valor <i>n</i> pertenece al intervalo,
<i>false</i>	en otro caso

Definición en la línea 165 del archivo intervalo.cpp.

```
165                                     {  
166     return ((p> cotaInf && p < cotaSup) || (p == cotaInf && cerradoInf) || (p == cotaSup && cerradoSup));  
167 }
```

2.1.3.4. `bool Intervalo::esVacio () const`

Consulta si el intervalo almacenado es vacío o no.

Devuelve

Valores devueltos

<i>true</i>	si es un intervalo vacío,
<i>false</i>	en otro caso

Definición en la línea 162 del archivo intervalo.cpp.

```
162         {  
163     return ((cotaInf == cotaSup) && !cerradoInf);  
164 }
```

2.1.3.5. `double Intervalo::getCotaInf () const`

Devuelve la cota inferior del intervalo.

Devuelve

El valor de la cota

Definición en la línea 137 del archivo intervalo.cpp.

```
137         {  
138     return cotaInf ;  
139 }
```

2.1.3.6. `double Intervalo::getCotaSup () const`

Devuelve la cota superior del intervalo.

Devuelve

El valor de la cota

Definición en la línea 141 del archivo intervalo.cpp.

```
141         {  
142     return cotaSup ;  
143 }
```

2.1.3.7. `void Intervalo::setIntervalo (double cotaInferior, double cotaSuperior, bool cerradoInferior, bool cerradoSuperior)`

Define los valores del intervalo.

Parámetros

<i>cerradoInferior</i>	
<i>cerradoSuperior</i>	
<i>cotaInferior</i>	
<i>cotaSuperior</i>	$cotaInferior \leq cotaSuperior$

Definición en la línea 153 del archivo intervalo.cpp.

```
153                                     {
154     if (valido(cinf, csup, cerrinf, cerrsup)) {
155         cotaInf = cinf;
156         cotaSup = csup;
157         cerradoInf = cerrinf;
158         cerradoSup = cerrsup;
159     }
160 }
```

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

- intervalo.cpp

Índice alfabético

- dentroCotaInf
 - Intervalo, [4](#)
- dentroCotaSup
 - Intervalo, [5](#)
- esVacio
 - Intervalo, [5](#)
- estaDentro
 - Intervalo, [5](#)
- getCotaInf
 - Intervalo, [6](#)
- getCotaSup
 - Intervalo, [6](#)
- Intervalo, [3](#)
 - dentroCotaInf, [4](#)
 - dentroCotaSup, [5](#)
 - esVacio, [5](#)
 - estaDentro, [5](#)
 - getCotaInf, [6](#)
 - getCotaSup, [6](#)
 - Intervalo, [3](#), [4](#)
 - setIntervalo, [6](#)
- setIntervalo
 - Intervalo, [6](#)