

My Project

Generado por Doxygen 1.12.0

1 Índice de estructuras de datos	1
1.1 Estructuras de datos	1
2 Índice de archivos	3
2.1 Lista de archivos	3
3 Documentación de estructuras de datos	5
3.1 Referencia de la clase alimento	5
3.1.1 Descripción detallada	5
3.1.2 Documentación de constructores y destructores	6
3.1.2.1 alimento()	6
3.2 Referencia de la clase electro	6
3.2.1 Descripción detallada	7
3.2.2 Documentación de constructores y destructores	7
3.2.2.1 electro()	7
3.3 Referencia de la clase fecha	7
3.3.1 Descripción detallada	8
3.3.2 Documentación de constructores y destructores	8
3.3.2.1 fecha()	8
3.4 Referencia de la clase producto	9
3.4.1 Descripción detallada	9
3.4.2 Documentación de constructores y destructores	9
3.4.2.1 producto()	9
4 Documentación de archivos	11
4.1 alimento.h	11
4.2 electro.h	11
4.3 fecha.h	11
4.4 producto.h	12
Índice alfabético	13

Capítulo 1

Índice de estructuras de datos

1.1. Estructuras de datos

Lista de estructuras con breves descripciones:

alimento	Clase que almacena un producto alimenticio. Almacena el nombre, precio y la fecha de caducidad	5
electro	Clase para almacenar datos de un producto electrodoméstico. Almacena el nombre del electrodoméstico, su precio y el voltaje con el que funciona	6
fecha	Clase para almacenar fechas ingresadas por el usuario. Almacena el día, mes y año	7
producto	Clase para manejar un producto. Guarda el nombre y el precio de un artículo del mercado . . .	9

Capítulo 2

Índice de archivos

2.1. Lista de archivos

Lista de todos los archivos documentados y con breves descripciones:

alimento.h	11
electro.h	11
fecha.h	11
producto.h	12

Capítulo 3

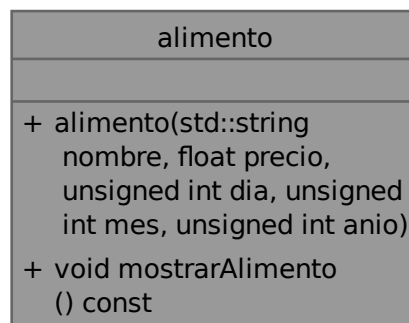
Documentación de estructuras de datos

3.1. Referencia de la clase alimento

clase que almacena un producto alimenticio. Almacena el nombre, precio y la fecha de caducidad.

```
#include <alimento.h>
```

Diagrama de colaboración de alimento:



Métodos públicos

- `alimento` (std::string nombre, float precio, unsigned int dia, unsigned int mes, unsigned int anio)
Constructor de la clase alimento.
- void `mostrarAlimento` () const
void `mostrarAlimento()` const Función constante que muestra datos del alimento en salida estándar.

3.1.1. Descripción detallada

clase que almacena un producto alimenticio. Almacena el nombre, precio y la fecha de caducidad.

3.1.2. Documentación de constructores y destructores

3.1.2.1. alimento()

```
alimento::alimento (
    std::string nombre,
    float precio,
    unsigned int dia,
    unsigned int mes,
    unsigned int anio)
```

Constructor de la clase alimento.

Parámetros

<i>std::string</i>	nombre: almacena el nombre en un string de la biblioteca estándar.
<i>float</i>	precio: almacena el precio en un flotante.
<i>unsigned</i>	int dia, mes y anio: almacena la fecha de caducidad.

La documentación de esta clase está generada del siguiente archivo:

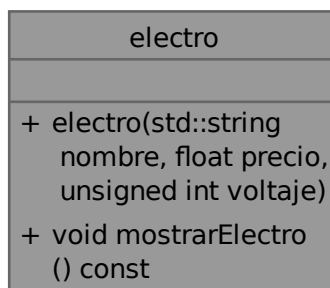
- alimento.h

3.2. Referencia de la clase electro

Clase para almacenar datos de un producto electrodoméstico. Almacena el nombre del electrodoméstico, su precio y el voltaje con el que funciona.

```
#include <electro.h>
```

Diagrama de colaboración de electro:



Métodos públicos

- **electro** (std::string nombre, float precio, unsigned int voltaje)

Constructor de la clase electro.

- void **mostrarElectro** () const

*void **mostrarElectro()** const: función contrante que muestra datos del electrodoméstico. Imprime en salida estándar datos del voltaje y el producto.*

3.2.1. Descripción detallada

Clase para almacenar datos de un producto electrodoméstico. Almacena el nombre del electrodoméstico, su precio y el voltaje con el que funciona.

3.2.2. Documentación de constructores y destructores

3.2.2.1. electro()

```
electro::electro (
    std::string nombre,
    float precio,
    unsigned int voltaje)
```

Constructor de la clase electro.

Parámetros

<i>std::string</i>	nombre: almacena el nombre del electrodoméstico en una variable string de la clase estándar.
<i>float</i>	precio: almaceena el precio en un flotante.
<i>unsigned</i>	int voltaje: almacena el voltaje de funcionamiento.

La documentación de esta clase está generada del siguiente archivo:

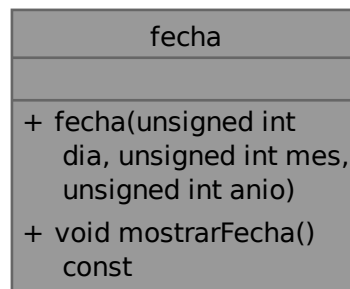
- electro.h

3.3. Referencia de la clase fecha

Clase para almacenar fechas ingresadas por el usuario. Almacena el día, mes y año.

```
#include <fecha.h>
```

Diagrama de colaboración de fecha:



Métodos públicos

- [fecha](#) (unsigned int dia, unsigned int mes, unsigned int anio)
Constructor de la clase fecha.
- void **mostrarFecha** () const
void [mostrarFecha\(\)](#) const Función constante que muestra la fecha en salida estándar.

3.3.1. Descripción detallada

Clase para almacenar fechas ingresadas por el usuario. Almacena el día, mes y año.

3.3.2. Documentación de constructores y destructores

3.3.2.1. fecha()

```
fecha::fecha (
    unsigned int dia,
    unsigned int mes,
    unsigned int anio)
```

Constructor de la clase fecha.

Parámetros

<i>unsigned</i>	int dia: toma el día.
<i>unsigned</i>	int mes: toma el mes.
<i>unsigned</i>	int anio: toma el año.

La documentación de esta clase está generada del siguiente archivo:

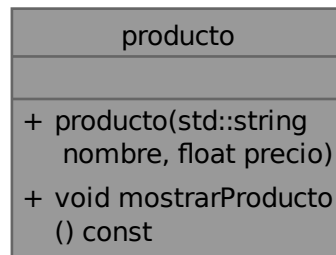
- fecha.h

3.4. Referencia de la clase producto

Clase para manejar un producto. Guarda el nombre y el precio de un artículo del mercado.

```
#include <producto.h>
```

Diagrama de colaboración de producto:



Métodos públicos

- `producto` (`std::string nombre`, `float precio`)
Constructor clase producto.
- `void mostrarProducto() const`
`void mostrarProducto() const`: función constante que muestra el producto almacenado (nombre y precio).

3.4.1. Descripción detallada

Clase para manejar un producto. Guarda el nombre y el precio de un artículo del mercado.

3.4.2. Documentación de constructores y destructores

3.4.2.1. producto()

```
producto::producto (
    std::string nombre,
    float precio)
```

Constructor clase producto.

Parámetros

<code>std::string</code>	Nombre del producto pasado como string de la biblioteca estándar.
<code>precio</code>	almacena el precio del producto como flotante.

La documentación de esta clase está generada del siguiente archivo:

- `producto.h`

Capítulo 4

Documentación de archivos

4.1. alimento.h

```
00001 #ifndef ALIMENTO_H
00002 #define ALIMENTO_H
00003
00004 #include "fecha.h"
00005 #include "producto.h"
00006 #include <string>
00007
00014 class alimento {
00015     public:
00016
00024         alimento(std::string nombre, float precio, unsigned int dia, unsigned int mes, unsigned int
00025         anio);
00026
00031         void mostrarAlimento() const;
00032     private:
00033         const fecha _caducidad;
00034         const producto _alimento;
00035 };
00036
00037 #endif
00038
```

4.2. electro.h

```
00001 #ifndef ELECTRO_H
00002 #define ELECTRO_H
00003
00004 #include "producto.h"
00005 #include <string>
00006
00013 class electro {
00014     public:
00015
00023         electro(std::string nombre, float precio, unsigned int voltaje);
00024
00030         void mostrarElectro() const;
00031
00032     private:
00033         const unsigned int _voltaje;
00034         const producto _electro;
00035 };
00036
00037 #endif
00038
00039
```

4.3. fecha.h

```
00001 #ifndef FECHA_H
```

```
00002 #define FECHA_H
00003
00004 #include <iostream>
00005
00012 class fecha {
00013     public:
00014
00022         fecha(unsigned int dia, unsigned int mes, unsigned int anio);
00023
00029         void mostrarFecha() const;
00030
00031
00032     private:
00033         const unsigned int _dia;
00034         const unsigned int _mes;
00035         const unsigned int _anio;
00036 };
00037
00038 #endif
00039
```

4.4. producto.h

```
00001 #ifndef PRODUCTO_H
00002 #define PRODUCTO_H
00003
00004 #include <string>
00005
00012 class producto {
00013     public:
00020         producto(std::string nombre, float precio);
00024         void mostrarProducto() const;
00025
00026     private:
00027         const std::string _nombre;
00028         const float _precio;
00029 };
00030
00031 #endif
00032
```


Índice alfabético

alimento, [5](#)
alimento, [6](#)

electro, [6](#)
electro, [7](#)

fecha, [7](#)
fecha, [8](#)

producto, [9](#)
producto, [9](#)