

SAPEICore

Generado por Doxygen 1.9.4

Chapter 1

Indice jerárquico

1.1 Jerarquía de la clase

Esta lista de herencias esta ordenada aproximadamente por orden alfabético:

QDialog	
AddCardDialog	??
QMainWindow	
MainWindow	??
QObject	
SerialHandler	??

Chapter 2

Índice de clases

2.1 Lista de clases

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

AddCardDialog	Clase que representa un cuadro de diálogo para agregar información de una tarjeta	??
MainWindow	Clase principal que maneja la interfaz gráfica y la lógica de la aplicación	??
SerialHandler	Clase que maneja la comunicación serie con un dispositivo (por ejemplo, Arduino)	??

Chapter 3

Indice de archivos

3.1 Lista de archivos

Lista de todos los archivos documentados y con descripciones breves:

/home/gaston/Documents/info2/SAPEI 1.0/SAPEIClient/ addcarddialog.h	
Declaración de la clase AddCardDialog	??
/home/gaston/Documents/info2/SAPEI 1.0/SAPEIClient/ mainwindow.h	
Declaración de la clase MainWindow	??
/home/gaston/Documents/info2/SAPEI 1.0/SAPEIClient/ serialhandler.h	
Declaración de la clase SerialHandler	??

Chapter 4

Documentación de las clases

4.1 Referencia de la Clase AddCardDialog

Clase que representa un cuadro de diálogo para agregar información de una tarjeta.

```
#include <addcarddialog.h>
```

Diagrama de herencias de AddCardDialog

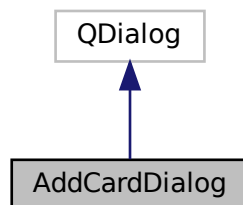
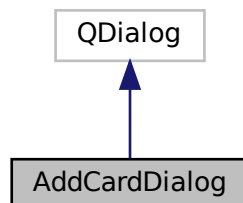


Diagrama de colaboración para AddCardDialog:



Métodos públicos

- [AddCardDialog](#) (QWidget *parent=nullptr)
Constructor de la clase [AddCardDialog](#).
- [~AddCardDialog](#) ()
Destructor de la clase [AddCardDialog](#).
- QString [getName](#) () const
Obtiene el nombre ingresado en el diálogo.
- QString [getLastName](#) () const
Obtiene el apellido ingresado en el diálogo.
- QString [getDNI](#) () const
Obtiene el DNI ingresado en el diálogo.
- QString [getEmail](#) () const
Obtiene el email ingresado en el diálogo.
- QString [getPhone](#) () const
Obtiene el teléfono ingresado en el diálogo.
- QString [getAddress](#) () const
Obtiene la dirección ingresada en el diálogo.
- int [getAge](#) () const
Obtiene la edad ingresada en el diálogo.
- QString [getLicense](#) () const
Obtiene la licencia ingresada en el diálogo.
- QString [getType](#) () const
Obtiene el tipo de vehículo ingresado en el diálogo.
- QString [getColor](#) () const
Obtiene el color del vehículo ingresado en el diálogo.
- QString [getBrand](#) () const
Obtiene la marca del vehículo ingresada en el diálogo.
- QString [getModel](#) () const
Obtiene el modelo del vehículo ingresado en el diálogo.

Atributos privados

- Ui::AddCardDialog * **ui**
Puntero a la interfaz de usuario del diálogo.

4.1.1 Descripción detallada

Clase que representa un cuadro de diálogo para agregar información de una tarjeta.

La clase [AddCardDialog](#) hereda de QDialog y proporciona un diálogo en la interfaz gráfica para ingresar detalles como nombre, apellido, DNI, email, teléfono, dirección, edad, información de licencia de vehículo y datos del vehículo (marca, modelo, color y tipo).

4.1.2 Documentación del constructor y destructor

4.1.2.1 AddCardDialog()

```
AddCardDialog::AddCardDialog (  
    QWidget * parent = nullptr ) [explicit]
```

Constructor de la clase [AddCardDialog](#).

Parámetros

<i>parent</i>	Puntero al widget padre. Si no se pasa, el valor predeterminado es nullptr.
---------------	---

4.1.3 Documentación de las funciones miembro

4.1.3.1 getAddress()

```
QString AddCardDialog::getAddress ( ) const
```

Obtiene la dirección ingresada en el diálogo.

Devuelve

Cadena de texto con la dirección.

4.1.3.2 getAge()

```
int AddCardDialog::getAge ( ) const
```

Obtiene la edad ingresada en el diálogo.

Devuelve

Número entero con la edad.

4.1.3.3 getBrand()

```
QString AddCardDialog::getBrand ( ) const
```

Obtiene la marca del vehículo ingresada en el diálogo.

Devuelve

Cadena de texto con la marca del vehículo.

4.1.3.4 getColor()

```
QString AddCardDialog::getColor ( ) const
```

Obtiene el color del vehículo ingresado en el diálogo.

Devuelve

Cadena de texto con el color del vehículo.

4.1.3.5 getDNI()

```
QString AddCardDialog::getDNI ( ) const
```

Obtiene el DNI ingresado en el diálogo.

Devuelve

Cadena de texto con el DNI.

4.1.3.6 getEmail()

```
QString AddCardDialog::getEmail ( ) const
```

Obtiene el email ingresado en el diálogo.

Devuelve

Cadena de texto con el email.

4.1.3.7 getLastName()

```
QString AddCardDialog::getLastName ( ) const
```

Obtiene el apellido ingresado en el diálogo.

Devuelve

Cadena de texto con el apellido.

4.1.3.8 getLicense()

```
QString AddCardDialog::getLicense ( ) const
```

Obtiene la licencia ingresada en el diálogo.

Devuelve

Cadena de texto con el número de licencia.

4.1.3.9 getModel()

```
QString AddCardDialog::getModel ( ) const
```

Obtiene el modelo del vehículo ingresado en el diálogo.

Devuelve

Cadena de texto con el modelo del vehículo.

4.1.3.10 getName()

```
QString AddCardDialog::getName ( ) const
```

Obtiene el nombre ingresado en el diálogo.

Devuelve

Cadena de texto con el nombre.

4.1.3.11 getPhone()

```
QString AddCardDialog::getPhone ( ) const
```

Obtiene el teléfono ingresado en el diálogo.

Devuelve

Cadena de texto con el número de teléfono.

4.1.3.12 getType()

```
QString AddCardDialog::getType ( ) const
```

Obtiene el tipo de vehículo ingresado en el diálogo.

Devuelve

Cadena de texto con el tipo de vehículo.

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

- /home/gaston/Documents/info2/SAPEI 1.0/SAPEIClient/addcarddialog.h
- /home/gaston/Documents/info2/SAPEI 1.0/SAPEIClient/addcarddialog.cpp

4.2 Referencia de la Clase MainWindow

Clase principal que maneja la interfaz gráfica y la lógica de la aplicación.

```
#include <mainwindow.h>
```

Diagrama de herencias de MainWindow

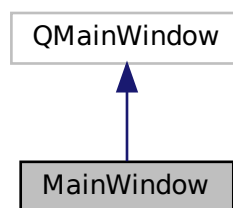
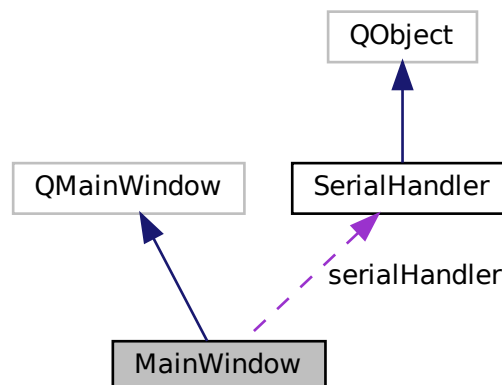


Diagrama de colaboración para MainWindow:



Métodos públicos

- [MainWindow](#) (QWidget *parent=nullptr)
Constructor de la clase [MainWindow](#).
- [~MainWindow](#) ()
Destructor de la clase [MainWindow](#).
- bool [validateId](#) (const QString &id)
Valida el ID recibido para determinar si es correcto.
- void [addCard](#) (const QString &id)
Agrega una tarjeta a la base de datos utilizando el ID proporcionado.

Slots privados

- void [on_addCardButton_clicked](#) ()
Slot que maneja el evento cuando se presiona el botón para agregar una tarjeta.
- void [onIdReceived](#) (const QString &id)
Slot que maneja el evento cuando se recibe un ID desde el puerto serie.
- void [onSelectSerialPortClicked](#) ()
Slot que maneja la selección del puerto serie.

Atributos privados

- [SerialHandler](#) * [serialHandler](#)
Manejador de comunicación serial.
- DataBase * [db](#)
Puntero a la base de datos utilizada para almacenar los clientes y tarjetas.
- bool [isAddingCardMode](#)
Indica si la aplicación está en modo de agregar tarjeta.
- bool [isChargingMode](#)
Indica si la aplicación está en modo de carga de saldo.
- QString [currentId](#)
Almacena temporalmente el ID de la tarjeta actual.
- Ui::MainWindow * [ui](#)
Puntero a la interfaz gráfica de la ventana principal.

4.2.1 Descripción detallada

Clase principal que maneja la interfaz gráfica y la lógica de la aplicación.

La clase [MainWindow](#) es responsable de interactuar con el usuario, manejar la recepción de IDs a través del puerto serie, y gestionar la base de datos de tarjetas. Además, permite agregar nuevas tarjetas y seleccionar el puerto serie.

4.2.2 Documentación del constructor y destructor

4.2.2.1 MainWindow()

```
MainWindow::MainWindow (  
    QWidget * parent = nullptr )
```

Constructor de la clase [MainWindow](#).

Parámetros

<i>parent</i>	Puntero al widget padre. Si no se pasa, el valor predeterminado es nullptr.
---------------	---

4.2.3 Documentación de las funciones miembro

4.2.3.1 addCard()

```
void MainWindow::addCard (
    const QString & id )
```

Agrega una tarjeta a la base de datos utilizando el ID proporcionado.

Parámetros

<i>id</i>	Cadena de texto con el ID de la tarjeta.
-----------	--

4.2.3.2 onIdReceived

```
void MainWindow::onIdReceived (
    const QString & id ) [private], [slot]
```

Slot que maneja el evento cuando se recibe un ID desde el puerto serie.

Parámetros

<i>id</i>	Cadena de texto con el ID recibido.
-----------	-------------------------------------

4.2.3.3 validateId()

```
bool MainWindow::validateId (
    const QString & id )
```

Valida el ID recibido para determinar si es correcto.

Parámetros

<i>id</i>	Cadena de texto con el ID a validar.
-----------	--------------------------------------

Devuelve

true si el ID es válido, false en caso contrario.

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

- /home/gaston/Documents/info2/SAPEI 1.0/SAPEIClient/[mainwindow.h](#)
- /home/gaston/Documents/info2/SAPEI 1.0/SAPEIClient/mainwindow.cpp

4.3 Referencia de la Clase SerialHandler

Clase que maneja la comunicación serie con un dispositivo (por ejemplo, Arduino).

```
#include <serialhandler.h>
```

Diagrama de herencias de SerialHandler

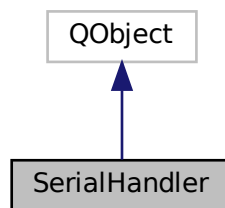
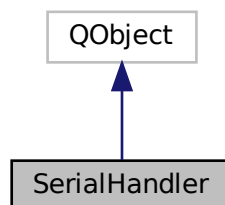


Diagrama de colaboración para SerialHandler:

**Señales**

- void [idReceived](#) (const QString &id)
Señal que se emite cuando se recibe un ID desde el dispositivo serie.

Métodos públicos

- `SerialHandler` (`QObject *parent=nullptr`)
Constructor de la clase `SerialHandler`.
- void `selectSerialPort` ()
Selecciona el puerto serie para establecer la conexión.
- void `connectSerialPort` (const `QString &portName`)
Conecta al puerto serie especificado.
- void `sendToArduino` (const `QString &message`)
Envía un mensaje al dispositivo conectado por el puerto serie.
- bool `isConnected` () const
Verifica si hay una conexión activa con el puerto serie.

Slots privados

- void `handleReadyRead` ()
Slot privado que maneja la lectura de los datos entrantes desde el puerto serie.

Atributos privados

- `QSerialPort * serial`
Puntero al objeto `QSerialPort` que maneja la comunicación serie.

4.3.1 Descripción detallada

Clase que maneja la comunicación serie con un dispositivo (por ejemplo, Arduino).

La clase `SerialHandler` permite seleccionar un puerto serie, conectarse a él, enviar mensajes, y recibir datos desde el dispositivo conectado. También proporciona una señal que emite cuando se recibe un ID.

4.3.2 Documentación del constructor y destructor

4.3.2.1 `SerialHandler()`

```
SerialHandler::SerialHandler (
    QObject * parent = nullptr )
```

Constructor de la clase `SerialHandler`.

Parámetros

<code>parent</code>	Puntero al objeto padre. Si no se pasa, el valor predeterminado es <code>nullptr</code> .
---------------------	---

4.3.3 Documentación de las funciones miembro

4.3.3.1 connectSerialPort()

```
void SerialHandler::connectSerialPort (
    const QString & portName )
```

Conecta al puerto serie especificado.

Parámetros

<i>portName</i>	Nombre del puerto serie al que se conectará.
-----------------	--

4.3.3.2 handleReadyRead

```
void SerialHandler::handleReadyRead ( ) [private], [slot]
```

Slot privado que maneja la lectura de los datos entrantes desde el puerto serie.

Este método es llamado automáticamente cuando hay datos disponibles para leer en el puerto serie.

4.3.3.3 idReceived

```
void SerialHandler::idReceived (
    const QString & id ) [signal]
```

Señal que se emite cuando se recibe un ID desde el dispositivo serie.

Parámetros

<i>id</i>	Cadena de texto que contiene el ID recibido.
-----------	--

4.3.3.4 isConnected()

```
bool SerialHandler::isConnected ( ) const
```

Verifica si hay una conexión activa con el puerto serie.

Devuelve

true si está conectado, false en caso contrario.

4.3.3.5 selectSerialPort()

```
void SerialHandler::selectSerialPort ( )
```

Selecciona el puerto serie para establecer la conexión.

Este método abre un diálogo para que el usuario seleccione el puerto serie al que desea conectarse.

4.3.3.6 sendToArduino()

```
void SerialHandler::sendToArduino (
    const QString & message )
```

Envía un mensaje al dispositivo conectado por el puerto serie.

Parámetros

<i>message</i>	Cadena de texto que contiene el mensaje a enviar.
----------------	---

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

- /home/gaston/Documents/info2/SAPEI 1.0/SAPEIClient/[serialhandler.h](#)
- /home/gaston/Documents/info2/SAPEI 1.0/SAPEIClient/serialhandler.cpp

Chapter 5

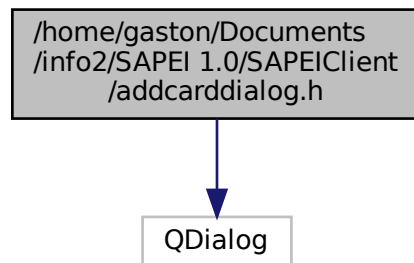
Documentación de archivos

5.1 Referencia del Archivo /home/gaston/Documents/info2/SAPEI 1.0/SAPEIClient/addcarddialog.h

Declaración de la clase [AddCardDialog](#).

```
#include <QDialog>
```

Dependencia gráfica adjunta para addcarddialog.h:



Clases

- class [AddCardDialog](#)

Clase que representa un cuadro de diálogo para agregar información de una tarjeta.

5.1.1 Descripción detallada

Declaración de la clase [AddCardDialog](#).

Este archivo contiene la declaración de la clase [AddCardDialog](#), que proporciona una interfaz de diálogo para agregar información de una tarjeta y sus detalles asociados.

5.2 /home/gaston/Documents/info2/SAPEI 1.0/SAPEIClient/addcarddialog.h

[Ir a la documentación de este archivo.](#)

```

1
9 #ifndef ADDCARDIALOG_H
10 #define ADDCARDIALOG_H
11
12 #include <QDialog>
13
14 namespace Ui {
15 class AddCardDialog;
16 }
17
26 class AddCardDialog : public QDialog {
27     Q_OBJECT
28
29 public:
34     explicit AddCardDialog(QWidget *parent = nullptr);
35
39     ~AddCardDialog();
40
45     QString getName() const;
46
51     QString getLastName() const;
52
57     QString getDNI() const;
58
63     QString getEmail() const;
64
69     QString getPhone() const;
70
75     QString getAddress() const;
76
81     int getAge() const;
82
87     QString getLicense() const;
88
93     QString getType() const;
94
99     QString getColor() const;
100
105     QString getBrand() const;
106
111     QString getModel() const;
112
113 private:
114     Ui::AddCardDialog *ui;
115 };
116
117 #endif // ADDCARDIALOG_H

```

5.3 Referencia del Archivo /home/gaston/Documents/info2/SAPEI 1.0/SAPEIClient/mainwindow.h

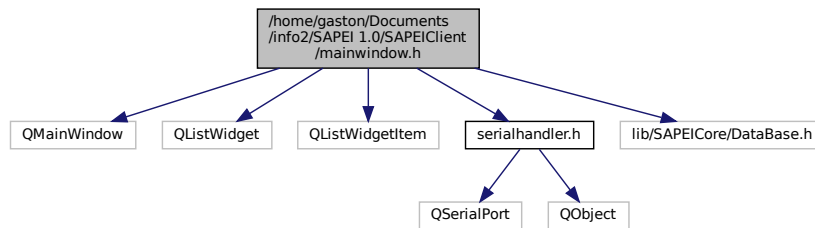
Declaración de la clase [MainWindow](#).

```

#include <QMainWindow>
#include <QListWidget>
#include <QListWidgetItem>
#include "serialhandler.h"
#include "lib/SAPEICore/DataBase.h"

```

Dependencia gráfica adjunta para mainwindow.h:



Clases

- class [MainWindow](#)

Clase principal que maneja la interfaz gráfica y la lógica de la aplicación.

5.3.1 Descripción detallada

Declaración de la clase [MainWindow](#).

Este archivo contiene la declaración de la clase [MainWindow](#), que es la ventana principal de la aplicación. Maneja la interacción con la interfaz gráfica de usuario, la comunicación serial, y la gestión de la base de datos.

5.4 /home/gaston/Documents/info2/SAPEI 1.0/SAPEIClient/mainwindow.h

[Ir a la documentación de este archivo.](#)

```

1
9 #ifndef MAINWINDOW_H
10 #define MAINWINDOW_H
11
12 #include <QMainWindow>
13 #include <QListWidget> // Añade soporte para la lista de widgets
14 #include <QListWidgetItem> // Soporte para los elementos de la lista
15 #include "serialhandler.h"
16 #include "lib/SAPEICore/DataBase.h"
17
18 QT_BEGIN_NAMESPACE
19 namespace Ui { class MainWindow; }
20 QT_END_NAMESPACE
21
22 class MainWindow : public QMainWindow {
23     Q_OBJECT
24
25 public:
26     MainWindow(QWidget *parent = nullptr);
27     ~MainWindow();
28
29     bool validateId(const QString &id);
30     void addCard(const QString &id);
31
32 private slots:
33     void on_addCardButton_clicked();
34
35     void onIdReceived(const QString &id);
36
37     void onSelectSerialPortClicked();
  
```

```
73
74 private:
75     SerialHandler *serialHandler;
76     DataBase *db;
77     bool isAddingCardMode;
78     bool isChargingMode;
79     QString currentId;
80     Ui::MainWindow *ui;
81 };
82
83 #endif // MAINWINDOW_H
```

5.5 Referencia del Archivo /home/gaston/Documents/info2/SAPEI 1.0/SAPEIClient/serialhandler.h

Declaración de la clase [SerialHandler](#).

```
#include <QSerialPort>
#include <QObject>
```

Dependencia gráfica adjunta para serialhandler.h:

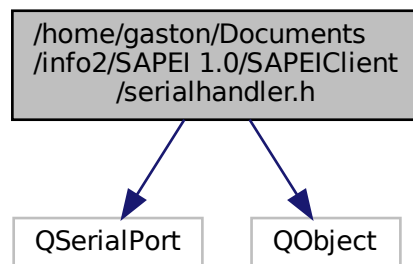
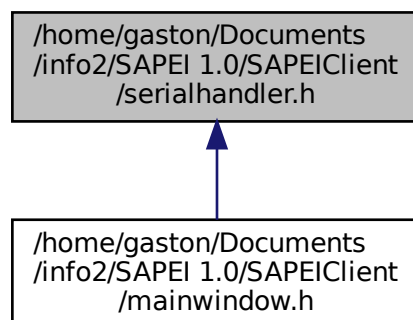


Gráfico de los archivos que directa o indirectamente incluyen a este archivo:



Clases

- class [SerialHandler](#)

Clase que maneja la comunicación serie con un dispositivo (por ejemplo, Arduino).

5.5.1 Descripción detallada

Declaración de la clase [SerialHandler](#).

Este archivo contiene la declaración de la clase [SerialHandler](#), responsable de manejar la comunicación con el puerto serie, enviar y recibir datos, y procesar los mensajes recibidos desde un dispositivo conectado por serial.

5.6 /home/gaston/Documents/info2/SAPEI 1.0/SAPEIClient/serialhandler.h

[Ir a la documentación de este archivo.](#)

```
1
9 #ifndef SERIALHANDLER_H
10 #define SERIALHANDLER_H
11
12 #include <QSerialPort>
13 #include <QObject>
14
22 class SerialHandler : public QObject {
23     Q_OBJECT
24
25 public:
30     SerialHandler(QObject *parent = nullptr);
31
37     void selectSerialPort();
38
43     void connectSerialPort(const QString &portName);
44
49     void sendToArduino(const QString &message);
50
55     bool isConnected() const;
56
57 signals:
62     void idReceived(const QString &id);
63
64 private slots:
70     void handleReadyRead();
71
72 private:
73     QSerialPort *serial;
74 };
75
76 #endif // SERIALHANDLER_H
```

