



# Universidad de Guadalajara

Centro Universitario de Ciencias Exactas e Ingenierías

Diseño de Interfaces

Sección: D02

**Puntos Extras 1: Crear un archivo .txt de configuración**



Alumno: Ramírez González Abel Yovanny

Código: 214287531

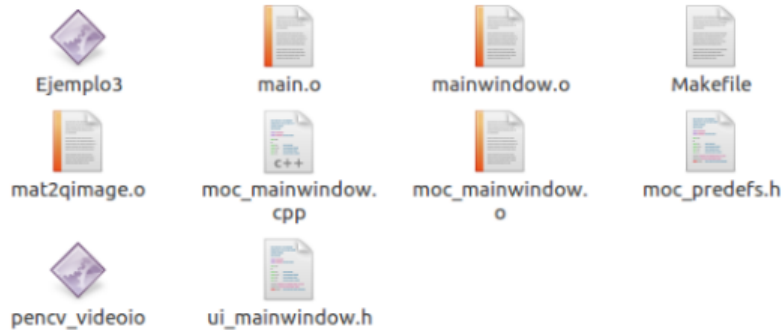
Profesor: Estrada Marmolejo Ruben

Fecha de Entrega: 07 de Septiembre de 2017

Carrera: Ingeniería en Comunicaciones y Electrónica.

A continuación, por medio de una serie de imágenes se muestra el correcto funcionamiento del programa (Crear un archivo txt de configuración y leerlo al inicio del programa para actualizar las variables importantes), y finalmente se muestran los códigos que lograron hacer posible lo anteriormente mencionado.

- 1) Trabajar por medio del ejemplo 17.



- 2) Lo primero es correr el programa llenando los campos propuestos

MainWindow

Ingresar

Nombre completo: Ramirez Glez Yovanny

A quién visita: Ramirez Carolina

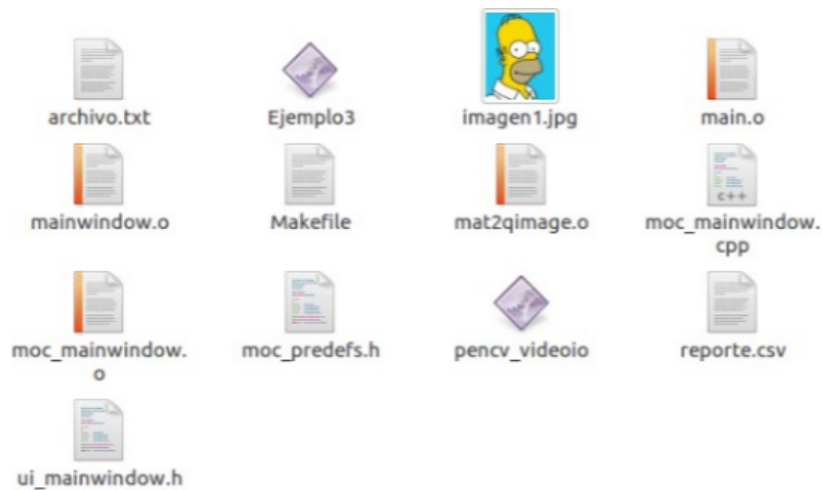
Empresa: Coca cola

Fecha/hora: sáb. oct 7 2017 15:47:47

☐ INE ☒ Pasaporte ☐ Licencia manejo ☐ Otros

Numero: 1234567

- 3) Enseguida se crean automáticamente los archivos: archivo.txt y reporte.csv, así mismo también se crea la imagen correspondiente, y esto se logra una vez que se corre el programa.



- 4) Se abren los archivos: archivo.txt y reporte.csv con la finalidad de corroborar que si tiene la información correcta.

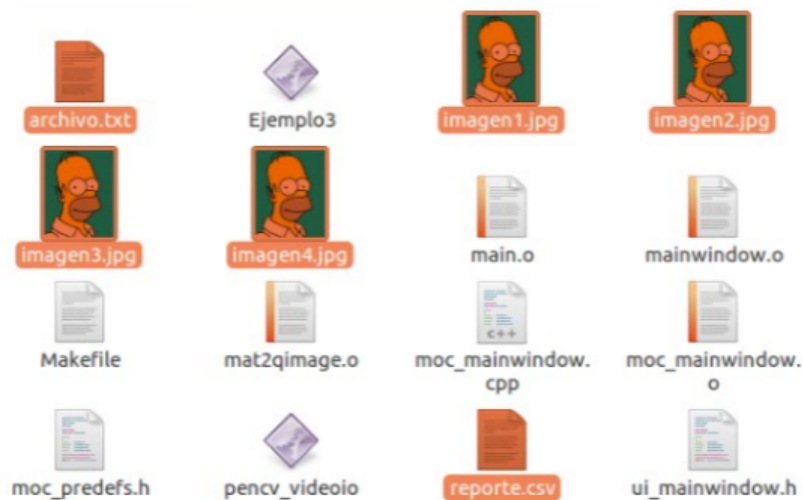
	A	B	C	D	E	F	G
1	Nombre	A quien visita	Empresa	Tipo de Identificacion	Numero de Identificacion	Nombre de imagen	Fecha y Hora de ingreso
2	Ramirez Glez Yovanny	Ramirez Carolina	Coca cola	PASAPORTE	1234567	imagen1.jpg	sáb, oct 7 2017 15:47:37
3							
4							



- 5) Cerramos el programa y lo corremos nuevamente para obtener la siguiente ventana (El numero del display tiene que ser el ultimo anteriormente obtenido: en este caso un 1).

- 6) Nuevamente llenamos los campos propuestos (hacerlos varias veces) en ingresamos los datos.

- 7) Nuevamente se crean automáticamente los archivos: archivo.txt y reporte.csv con los nuevos datos y así mismo también se crean las imágenes correspondiente.

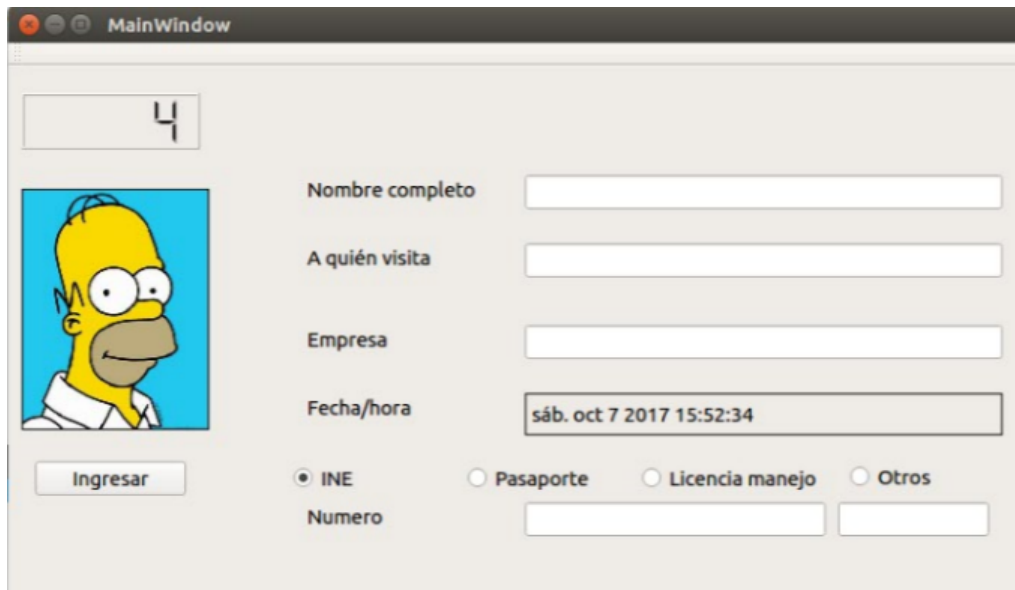


- 8) Se abren los archivos: archivo.txt y reporte.csv con la finalidad de corroborar que si tiene la información correcta.

	A	B	C	D	E	F	G
1	Nombre	A quien visita	Empresa	Tipo de Identificación	Numero de Identificación	Nombre de imagen	Fecha y Hora de ingreso
2	Ramirez Glez Yovanny	Ramirez Carolina	Coca cola	PASAPORTE	1234567	imagen1.jpg	sáb. oct 7 2017 15:47:37
3	yovanny	nadie	ninguna	INE	1234	imagen2.jpg	sáb. oct 7 2017 15:50:21
4	yovanny	julio	ninguna	LICENCIA	657868	imagen3.jpg	sáb. oct 7 2017 15:50:38
5	yovanny	Esteban	ninguna	credencial Cucei	657868	imagen4.jpg	sáb. oct 7 2017 15:51:15
6							



- 9) Finalmente cerramos el programa y lo corremos nuevamente para obtener la siguiente ventana (El numero del display tiene que ser el ultimo anteriormente obtenido: en este caso un 4).



Lo anterior se cumple para cualquier numeración.

## Código completo \*.pro:

```
# Project created by QtCreator 2017-08-18T16:20:48

QT      += core gui
greaterThan(QT_MAJOR_VERSION, 4): QT += widgets
TARGET = Ejemplo3
TEMPLATE = app
# The following define makes your compiler emit warnings if you use
# any feature of Qt which as been marked as deprecated (the exact warnings
# depend on your compiler). Please consult the documentation of the
# deprecated API in order to know how to port your code away from it.
DEFINES += QT_DEPRECATED_WARNINGS
# You can also make your code fail to compile if you use deprecated APIs.
# In order to do so, uncomment the following line.
# You can also select to disable deprecated APIs only up to a certain version of Qt.
#DEFINES += QT_DISABLE_DEPRECATED_BEFORE=0x060000
SOURCES += \
    main.cpp \
    mainwindow.cpp \
    mat2qimage.cpp
HEADERS += \
    mainwindow.h \
    mat2qimage.h
FORMS += \
    mainwindow.ui
INCLUDEPATH += /usr/local/include/opencv2
LIBS += -L/usr/local/lib -lopencv_core -lopencv_imgcodecs -lopencv_highgui -lopencv_videoio
CONFIG += link_pkgconfig
PKGCONFIG += opencv
```

## Código completo MainWindow.h:

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H
#include <QMainWindow>
namespace Ui {
class MainWindow;
}
class MainWindow : public QMainWindow
{
    Q_OBJECT
public:
    explicit MainWindow(QWidget *parent = 0);
    ~MainWindow();
public slots:
    void fTimer();
private slots:
    void on_pushButton_clicked();
private:
    Ui::MainWindow *ui;
};
#endif // MAINWINDOW_H
```

## Código completo MainWindow.cpp:

```
// Puntos Extras 1 : Guardar el valor del lcdNumber en un archivo.txt y leerlo
// Ramirez Gonzalez Abel Yovanny
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include "mat2qimage.h"
#include<opencv2/core/core.hpp>
#include<opencv2/ml/ml.hpp>
#include<opencv/cv.h>
#include<opencv2/imgproc/imgproc.hpp>
#include<opencv2/highgui/highgui.hpp>
#include<opencv2/video/background_segm.hpp>
#include<opencv2/videoio.hpp>
#include<opencv2/imgcodecs.hpp>
#include<qdebug.h>
#include<QTimer>
#include<QDateTime>
#include<QTime>
#include<QDate>
#include<QFile>
#include<QFileInfo>
using namespace cv;
QString nombre; // Declaramos todas las variables globales necesarias
QString empresa;
QString quien;
QString tiempoLocal;
QString tipoIdentificacion;
QString numeroIdentificacion;
QString nombreFoto;
```

```

int contador = 0;
Mat IMAGEN;
void MainWindow::fTimer(){
    //Actualizar la hora a la etiqueta correspondiente
    QDate fecha1 = QDate::currentDate();
    QTime tiempo = QTime::currentTime();
    QString fecha = fecha1.toString() + " " + tiempo.toString();
    tiempoLocal = fecha; // Campo a guardar en el excel
    ui->label_6->setText(fecha);
    //Se visualiza el contador guardado sin la necesidad de dar un clic
    ui->lcdNumber->display(contador);

    Mat IMAGEN2;
    IMAGEN = cv::imread("../img/homer.jpg");
    cv::resize(IMAGEN,IMAGEN2,Size(138,177),0,0,0);
    QImage qImage = Mat2QImage(IMAGEN2);
    QPixmap pixmap = QPixmap::fromImage(qImage);
    ui->label->clear();
    ui->label->setPixmap(pixmap);
}
MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    QString direccion="archivo.txt"; // Buscamos la direccion del archivo a leer
    QFile archivo(direccion);
    archivo.open(QIODevice::ReadOnly);
    QTextStream leerlo(&archivo); // Comenzamos a leer el archivo
    QString contenidoarchivo=(leerlo.readAll().toUtf8().constData()); // leemos el archivo
    // Indicamos lo que verdaderamente nos importa leer del archivo
    int a=contenidoarchivo.indexOf("<");
    int b=contenidoarchivo.indexOf(">");
    QString leido=contenidoarchivo.mid(a+1,(b-a-1));
    contador=leido.toInt(); // El contador toma su nuevo valor
    ui->radioButton->setChecked(true);
    QTimer *cronometro=new QTimer(this);
    connect(cronometro, SIGNAL(timeout()), this, SLOT(fTimer()));
    cronometro->start(30);
}
MainWindow::~MainWindow()
{
    delete ui;
}
void MainWindow::on_pushButton_clicked() // Todo esta dentro del boton Ingresar
{
    contador++; // Incrementar el contador
    ui->lcdNumber->display(contador); // Actualizar el display del registro
    // Obtener la informacion de los LineEdit en los siguientes LineEdit
    nombre = ui->lineEdit->text();
    quien = ui->lineEdit_2->text();
    empresa = ui->lineEdit_3->text();
    // Condicionales para saber el tipo de identificacion a utilizar

```

```

if(ui->radioButton->isChecked()) tipoIdentificacion = "INE";
else if(ui->radioButton_2->isChecked()) tipoIdentificacion = "PASAPORTE";
else if(ui->radioButton_3->isChecked()) tipoIdentificacion = "LICENCIA";
else if(ui->radioButton_4->isChecked()) tipoIdentificacion = ui->lineEdit_4->text();
numeroIdentificacion = ui->lineEdit_5->text();
// Creacion y verificacion de la existencia del archivo.txt
QFile archivo("archivo.txt");
QFileInfo existe2("archivo.txt");
if(existe2.isFile() && existe2.exists()){
// Crear el archivo con el contenido de las variables globales
if(archivo.open(QIODevice::WriteOnly | QIODevice::Append)){
    QTextStream out(&archivo);
    // Es fundamental para que obtener la numeracion del contador correcta
    archivo.resize(0);
    out << "<"; // Pasar la nueva numeracion correspondiente
    out << contador;
    out << ">";
    archivo.close();
}
}

// Creacion y verificacion de la existencia del archivo de reporte.csv
QFile reporte("reporte.csv");
QFileInfo existe("reporte.csv");
if(!existe.exists()){
    reporte.open(QIODevice::WriteOnly);
    // Agregamos los siguientes campos particulares a la base de datos (Nombre, Quien...)
    QTextStream out(&reporte);
    out << "Nombre" << ",";
    out << "A quien visita " << ",";
    out << "Empresa" << ",";
    out << "Tipo de Identificacion" << ",";
    out << "Numero de Identificacion" << ",";
    out << "Nombre de imagen" << ",";
    out << "Fecha y Hora de ingreso \n";
    reporte.close();
}
if(existe.isFile() && existe.exists()){
// Crear el archivo con el contenido de las variables globales
if(reporte.open(QIODevice::WriteOnly | QIODevice::Append)){
    QTextStream out(&reporte);
    // Pasar cada una de las variables hasta cerrar el archivo
    out << nombre.toUtf8().constData() << ",";
    out << quien.toUtf8().constData() << ",";
    out << empresa.toUtf8().constData() << ",";
    out << tipoIdentificacion.toUtf8().constData() << ",";
    out << numeroIdentificacion.toUtf8().constData() << ",";
    out << "imagen" << contador << ".jpg" << ",";
    out << tiempoLocal.toUtf8().constData() << "\n";
    reporte.close();
    QString nombreImagen = "imagen"+    QString::number(contador)+".jpg";
    cv::imwrite(nombreImagen.toUtf8().constData(), IMAGEN);
}
}
}

```