

Universidad de Guadalajara
Centro Universitario de Ciencias Exactas e Ingenierías
Ingeniería en Comunicaciones y Electrónica



Tarea puntos extras
Diseño de interfaces – D02
Pimienta Tabares Leonardo
Cod: 208629676
Fecha 06-10-2017
Dr: Rubén Estrada Marmolejo

Contenido

PRACTICA 1 1

 Objetivo..... 1

 Desarrollo..... 1

 Código..... 3

PRACTICA 1

Objetivo

Crear un archivo .txt de configuración y leerlo al inicio del programa para actualizarlas las variables importantes en este caso llevar un control de número de visitantes

Desarrollo

Se realiza el GUI con las siguientes puntos como se muestran en la figura



The screenshot shows a graphical user interface (GUI) window titled "MainWindow". The interface is designed for registering visitors. It features a digital display at the top left showing the number "0". To its right, a label "Ultimo numero de visitantes Registrados" is followed by a text box containing the number "26". Below the display is a camera feed showing a person's silhouette. To the right of the camera feed are several input fields: "Nombre completo", "A quien visita", "Empresa", "Fecha / Hora" (which is pre-filled with "vie. oct 6 2017 22:15:02"), and "Número". Below these fields are four radio buttons for selecting the type of identification: "INE" (selected), "Pasaporte", "Licencia manejo", and "Otro". A text box is located below the "Otro" option. At the bottom left, there is a button labeled "Ingresar".

Fig1 GUI con cama-web

Al llenar los campos se da click en ingresar es cuando registra un visitante más y empieza el contenido más los del día anterior en la parte superior de la foto ya cerrar la GUI se guarda el total de visitantes junto con los anteriores para el nuevo día tener la suma total



Fig2 GUI con cama-web

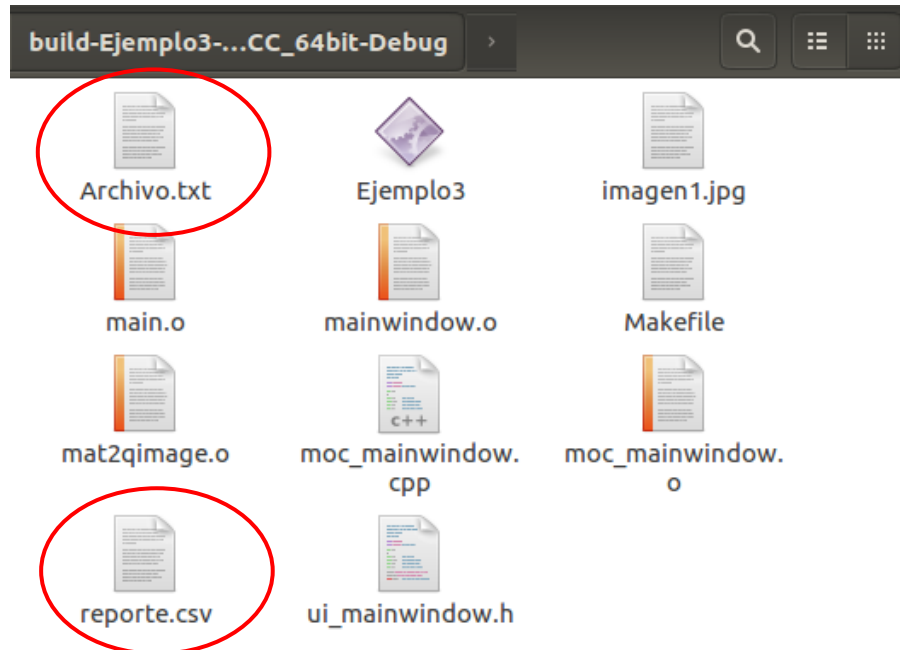


Fig3 Archivos creados

Código

```
#include "mainwindow.h"           // Bibliotecas utilizadas

#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>
// ..... Variables globales .....
QString empresa;
QString nombre;
QString quien;
QString tipoID;
QString numeroID;
QString fecha;
QString nombreFoto;
int contador;
QString nombreArchivo = "reporte.csv";
QString fechaActual;
QString linea;

using namespace cv;

cv::VideoCapture camara(0); // se utilizo la web-cam

// ..... Funcion Timmer
void MainWindow::fTimer(){
    QDate dia = QDate::currentDate(); // Se utiliza para actualizar la hora y fecha
    QTime hora = QTime::currentTime(); // mostrandola en un texlabel
    QString fecha = dia.toString() + " " + hora.toString();
    fechaActual = fecha;
    ui->label_7->setText(fecha);

    Mat IMAGEN;           //Se crea una matriz para procesar la imagen
    Mat IMAGEN2;
    if(camara.isOpened()){ //Se abre la Web-cam
        camara >> IMAGEN;
        cv::resize(IMAGEN,IMAGEN2,Size(400,300),0,0,0); //Se reduce el tamaño de 400,300 de la imagen
        QImage qImage = Mat2QImage(IMAGEN2); //se procesa para mostrarse en los texlabel
        QPixmap pixmap = QPixmap::fromImage(qImage);
```



```

    ui->label->clear();
    ui->label->setPixmap(pixmap);
}

//Abrir el Archivo para rescribirlo si existio una modificacion
QFile EscribirArchivo("Archivo.txt");
EscribirArchivo.open(QIODevice::WriteOnly | QIODevice::Truncate);
QTextStream TextoEscribir (&EscribirArchivo);
//Escribir el texto en el archivo
    TextoEscribir << contador;
//Cerra el archivo
    EscribirArchivo.close();

}

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    ui->radioButton->setChecked(true);
    // :::::::::: Se inicializa la funcion Timmer con una velocidad de 30 milisegundos
    QTimer *cronometro=new QTimer(this);
    connect(cronometro, SIGNAL(timeout()), this, SLOT(fTimer()));
    cronometro->start(30);

    if(!camara.isOpened()) camara.open(0); // se prebe que la camara se pueda abrir y si no mandar error
    else qDebug() << "No se abre la camara";

    // :::::::::: Iniciamos Abriendo el Archivo para leerlo ::::::::::
    QFile leerArchivo("Archivo.txt");
    if (leerArchivo.open(QIODevice::ReadOnly | QIODevice::Text)){ //se abre el archivo y se lee
        QTextStream stream(&leerArchivo); //Se ulizo QTextStream para extraer el texto en este caso numeros
        while (!stream.atEnd()){ // si se puede leer se extrae el texto
            linea.append(stream.readLine());
        }
        ui->lineEdit->setText(linea); //Se imprime en un lineEdit el valor leído en archivo txt
    }
    leerArchivo.close();

    contador =linea.toInt(); //convertimos una variable Qstrin en tipo entera para su manejo mas adelante

}

void MainWindow::on_pushButton_clicked() // por medio de la funcion del boton el GUI
{
    contador++; //contador de personas que ingresan
    // :::::: Se asignan las variables al los campos de escritura para su proceso
    ui->lcdNumber->display(contador);
    nombre = ui->lineEdit->text();
    quien = ui->lineEdit_2->text();
    empresa = ui->lineEdit_4->text();
}

```



```

numeroID = ui->lineEdit_5->text();
// ::::: diferentes campos de seleccion del radio_boton
if(ui->radioButton->isChecked()){
    tipoID = "INE";
}
else if(ui->radioButton_2->isChecked()){
    tipoID = "Pasaporte";
}
else if(ui->radioButton_3->isChecked()){
    tipoID = "Licencia de manejo";
}
else if(ui->radioButton_4->isChecked()){
    tipoID = ui->lineEdit_6->text();
}
// ::::: se crea un archivo con los valores asignados a cada campo
QFile reporte(nombreArchivo);
QFileInfo existe(nombreArchivo);
if(!existe.exists()){
    reporte.open(QIODevice::WriteOnly);
    QTextStream out(&reporte);
    out << "Nombre" << ",";
    out << "A quien se visita" << ",";
    out << "Empresa" << ",";
    out << "Tipo de ID" << ",";
    out << "Numero de ID" << ",";
    out << "Nombre de imagen" << ",";
    out << "Fecha y hora de ingreso \n";
    reporte.close();
}
if(existe.isFile() && existe.exists()){
    if(reporte.open(QIODevice::WriteOnly | QIODevice::Append)){
        QTextStream out(&reporte);
        out << nombre.toUtf8().constData() << ",";
        out << quien.toUtf8().constData() << ",";
        out << empresa.toUtf8().constData() << ",";
        out << tipoID.toUtf8().constData() << ",";
        out << numeroID.toUtf8().constData() << ",";
        out << "imagen" << contador << ".jpg" << ",";
        out << fechaActual.toUtf8().constData() << "\n";
        reporte.close();
        QString nombreImagen = "imagen" + QString::number(contador) + ".jpg";
        cv::imwrite(nombreImagen.toUtf8().constData(),IMAGENCHIDA);
    }
}
}

MainWindow::~MainWindow()
{
    delete ui;
}

```

