Universidad de Guadalajara.
Centro Universitario de Ciencias Exactas e Ingenierías.

DISEÑO DE INTERFACES.

Josué Daniel Alonso Jaquez 210422086.

Punto Extra.

Guardar en archivo sin sobrescribir.

En este reporte se muestra un programa capaz de escribir en un archivo de texto, que al cerrarse guarda la información y al hacer una nueva entrada de datos, no se sobre escribe y se guarda en disco.

Al ejecutar el programa tenemos esta pantalla, en la que se muestra un LineEdit para ingresar el texto que se desea escribir en el archivo, al presionar guardar se efectuará dicha acción.

Figura 1 Pantalla Inicio

Ahora al escribir en el lineEdit tenemos:

Figura 2 Ingreso de información

Al presionar guardar, la información se guardará en el archivo que se acaba de crear.

Si vamos a la carpeta del programa ejecutable podemos ver que se creó el archivo , con el nombre asignado implícitamente.

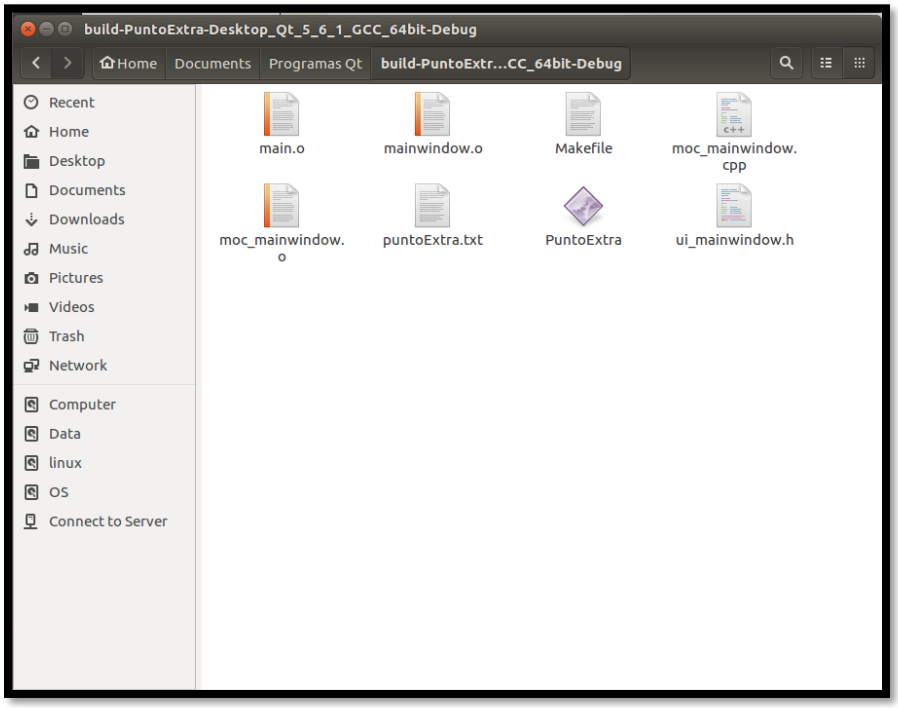

Figura 3 Archivo creado.

Ahora al abrir el archivo, tenemos que se guardó el texto ingresado.

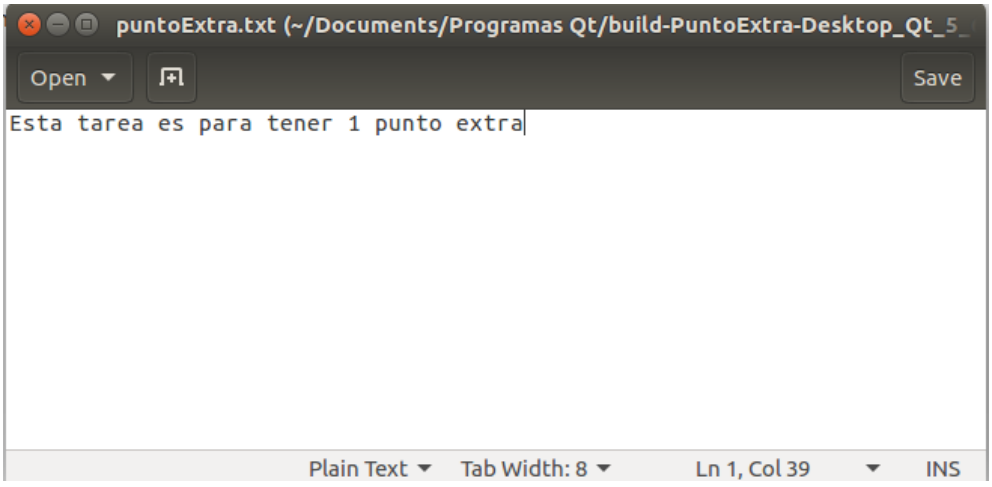

Figura 4: Abriendo el archive.

Repetimos el proceso de escribir en el LineEdit para comprobar el funcionamiento:

Figura 5 Nuevo Ingreso de datos

Al abrir el archivo tenemos que la información se ha guardado sin sobrescribirse y haciendo un salto de línea.

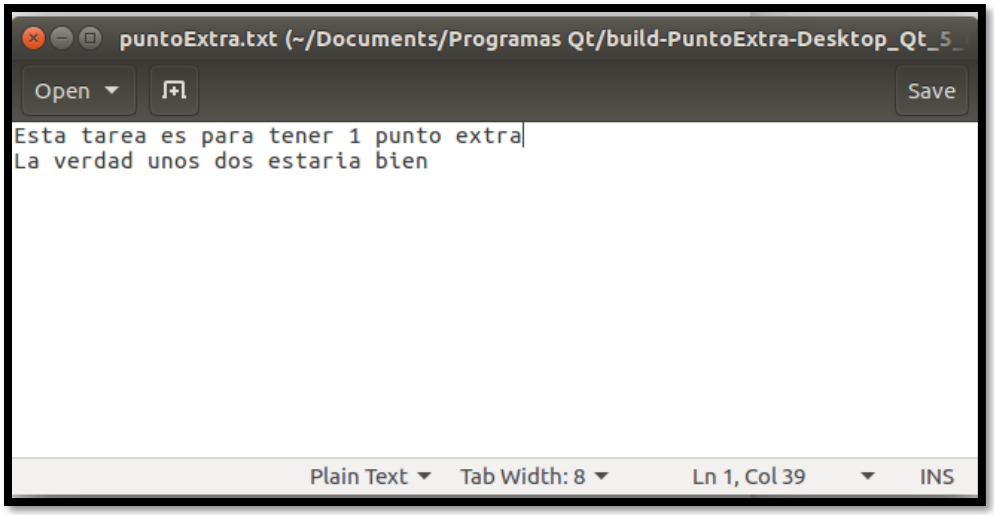

Figura 6 Escritura exitosa

Finalmente, en caso de no ingresar datos y presionar “Guardar” se muestra este mensaje de error.

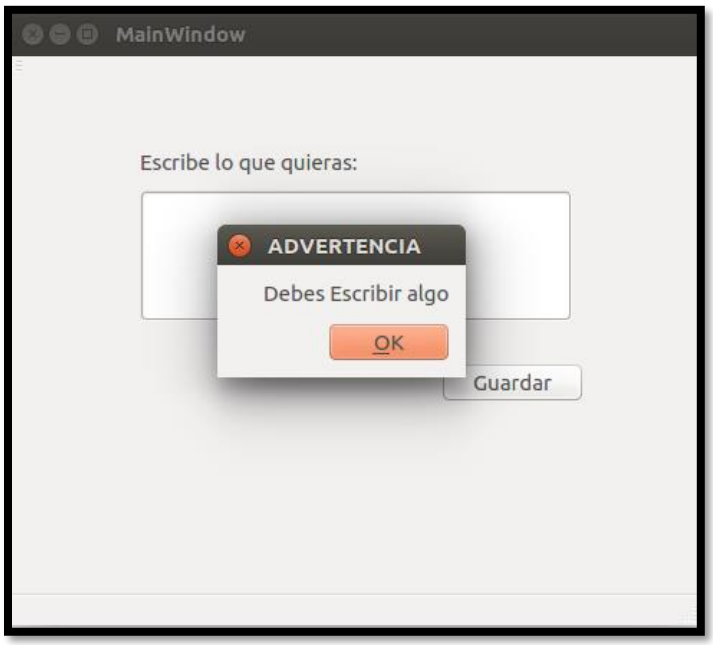

Figura 7 Mensaje Advertencia

En las siguientes páginas se anexa el código del programa.

Mainwindow.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QDebug>
#include <QFileInfo>
#include <qmessagebox.h>

MainWindow::MainWindow(QWidget *parent) :
   QMainWindow(parent),
   ui(new Ui::MainWindow)
{
   ui->setupUi(this);                ///Despliegue de UI
}

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

void MainWindow::on_pushButton_clicked()  ///Funcion de boton presionado
{
   QString lectura,comando;            ///Declaracion Qstring
   QFile archivo("puntoExtra.txt");     ///Declaracion del objeto archivo
   QFileInfo existe("puntoExtra.txt");   ///Declaracion de Qfile info del archivo creado

   lectura=ui->lineEdit->text();        ///Asigna el texto del lineEdit al string
   qDebug()<<lectura;                  ///Muestra en terminal lo que se escribio

if(lectura.isEmpty()){                 /// Si el string esta vacio no se escribe
qDebug()<<"No se escribio";
QMessageBox::about(this,"ADVERTENCIA","Debes Escribir algo"); /// Notificacion del error
ui->lineEdit->clear();                  /// Limpia el LineEdit
}
else{
    /// Si se escribio , se crea el archivo

    if(!existe.exists()){              /// Si el archivo no existe , se crea
       archivo.open(QIODevice::WriteOnly);  ///Abre el archivo , modo solo escritura
            QTextStream out(&archivo); /// Inicializacion de objeto QtextStream recibe el archivo
            out <<lectura;           /// Escribe en el archivo que recibio , la informacion
            out<<"\n";              ///asegura salto de linea
            archivo.close();         /// cierra el archivo
  }
    else{ /// Si el archivo si existe , se crea uno nuevo como auxiliar
        QFile temporal("temporal.txt");      ///crea archivo temporal
        temporal.open(QIODevice::WriteOnly);  ///abre archivo modo escritura
        QTextStream out2(&temporal); /// Se hace la escritura de la informacion
          out2 <<lectura;
          out2<<"\n";
           temporal.close();               ///Se cierra el archivo
 comando ="cat temporal.txt >> puntoExtra.txt";  ///Se utiliza funcion de
                                  concatenacion de archivos
           system(comando.toUtf8().constData());         ///Se ejecuta
           comando ="rm temporal.txt";                 ///Se borra el archivo temporal
```

```
            system(comando.toUtf8().constData());
    }
ui->lineEdit->clear();                  /// Limpia el LineEdit

}
}
```

main.cpp

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

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;           /// Declaracion de la clase en main
    w.show();               /// Llama a la funcion show
    return a.exec();
}
```

mainwindow.h

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>

namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow    ///Inicializacion de la clase mainWindow
{
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = 0);
    ~MainWindow();

private slots:
    void on_pushButton_clicked();       ///La clase mainWindow tiene la funcion de boton

private:
    Ui::MainWindow *ui;
};

#endif // MAINWINDOW_H
```