

## Actividad 5. QFileDialog

### Introducción

En esta actividad agregaras a tu proyecto de interfaz, de la actividad pasada (QPlainTextEdit), la característica de respaldar la información de las *Neuronas* en un archivo, usando la estructura de archivo que más te agrade.

También, será necesario que agregues la característica de recuperación de información, esto es, que tu interfaz sea capaz de elegir un archivo, el cual contiene un respaldo de *Neuronas*.

### Objetivo

El alumno complementara el desarrollo de la GUI:

- Haciendo uso de la clase QFileDialog para elegir la ubicación de un archivo a guardar o recuperar.
- Haciendo uso de un método creado para guardar una lista de *Neuronas* en un archivo con la estructura que más le convenga.
- Haciendo uso de un método creado para leer un archivo con la estructura que se eligió y convertirlo a una lista de *Neuronas*.

### Instrucciones

- Agrega al proyecto de la interfaz gráfica desarrollada en la actividad anterior (QPlainTextEdit), para que sea capaz de Guardar las *Neuronas* en archivo con la estructura que más te convenga. Así mismo, agregar la capacidad para abrir un archivo con la estructura que se eligió y recuperar las *Neuronas*. Los métodos anteriores deberán de estar conectadas a acciones (actions) dentro de tu interfaz gráfica.
- Haz pruebas a tu interfaz con las siguientes instrucciones:

- Agrega 3 *Neuronas* y crea un respaldoado.
- Después cierra tu interfaz y vuelva a correr la interfaz, con el fin de poder probar la opción de recuperar.
- Una vez recuperada la información desde el archivo, muestra las *Neuronas* en tu interfaz

## Desarrollo

Al realizar que la interfaz gráfica logre guardar y recuperar las neuronas de un archivo de la lista fue necesario utilizar los botones que fueron programados. Para se generó el código de los dos botones.

- Guardar

Este botón abre una ventana para guardar el archivo en alguna ubicación de la computadora.

```
void MainWindow::on_pushButton_4_clicked() {
    //RESPALDO
    string delimitador = "|";
    if(!lista.empty()){
        QString filename =
        QFileDialog::getSaveFileName(this, "Save
        file", "respaldo.txt");
        ofstream
        archivo(filename.toStdString(), ios::out);
        list<Neurona>::iterator aux = lista.begin();
        while(aux!=lista.end()){
            archivo<<aux->getId()<<delimitador;
            archivo<<aux->getVoltaje()<<delimitador;
            archivo<<aux->getPosX()<<delimitador;
            archivo<<aux->getPosY()<<delimitador;
            archivo<<aux->getRed()<<delimitador;
            archivo<<aux->getGreen()<<delimitador;
            archivo<<aux->getBlue()<<delimitador;
            aux++;
        }

        QMessageBox::information(this, "Guardado", "Guardado
        exitosamente...");
    }else{
        ui->plainTextEdit->clear();
        ui->plainTextEdit->insertPlainText("La lista
        esta vacia...");
    }
}
```

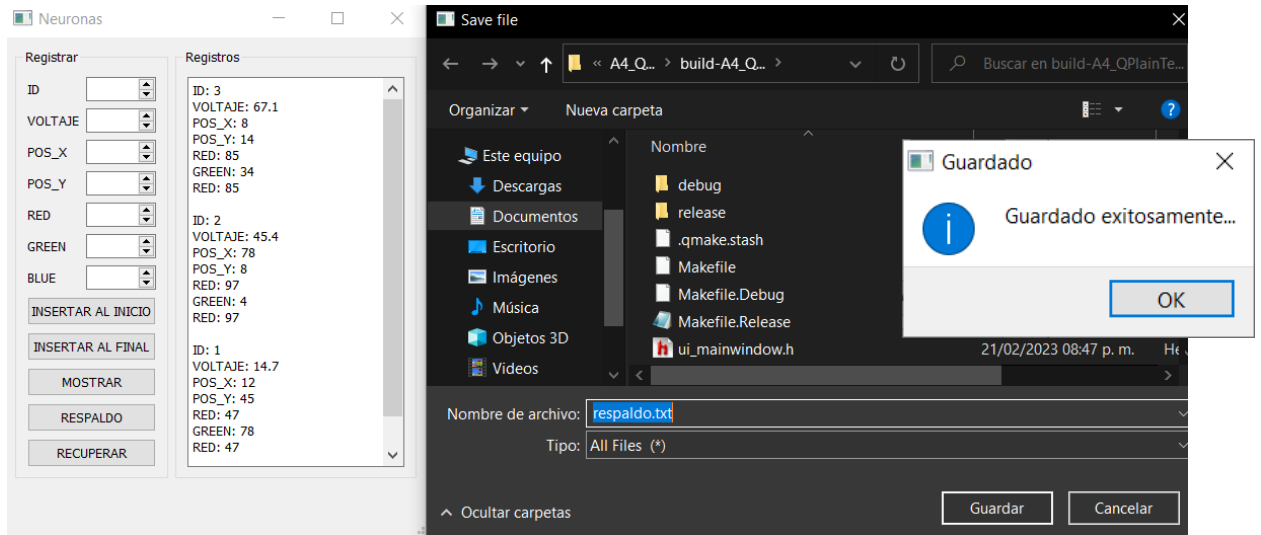
```
}  
}
```

- Recuperar

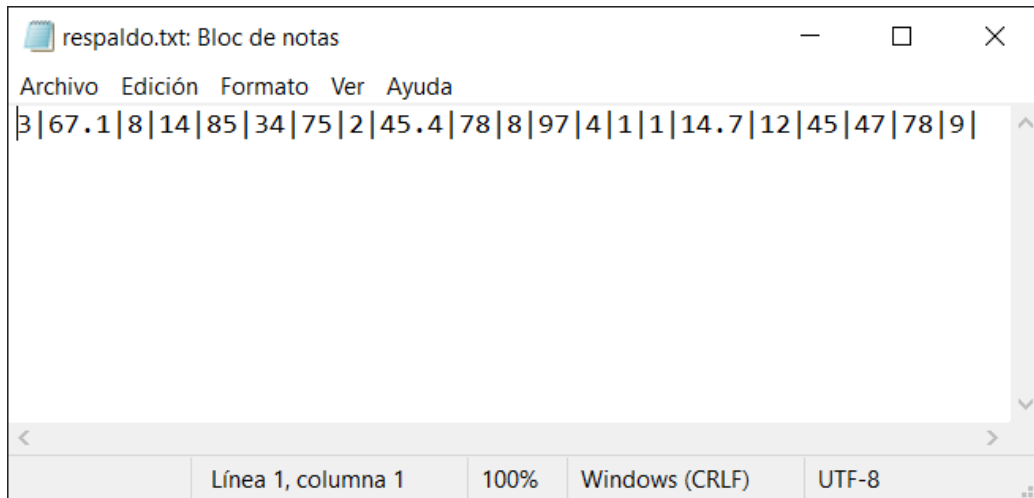
Para recuperar los datos, presionamos el botón y se abrirá una nueva ventana para buscar el archivo que vamos a recuperar.

```
void MainWindow::on_pushButton_5_clicked() {  
    //RECUPERAR  
    int id, posicion_x, posicion_y, red, green, blue;  
    double voltaje;  
    string registro;  
    QString filename =  
    QFileDialog::getOpenFileName(this, "Open file", "C://");  
    ifstream archivo(filename.toStdString(), ios::in);  
    if(!archivo.fail()) {  
        while(!archivo.eof()) {  
            getline(archivo, registro, '|');  
            id=atoi(registro.c_str());  
            if(id != NULL) {  
                getline(archivo, registro, '|');  
                voltaje=strtof(registro.c_str(),  
nullptr);  
                getline(archivo, registro, '|');  
                posicion_x=atoi(registro.c_str());  
                getline(archivo, registro, '|');  
                posicion_y=atoi(registro.c_str());  
                getline(archivo, registro, '|');  
                red=atoi(registro.c_str());  
                getline(archivo, registro, '|');  
                green=atoi(registro.c_str());  
                getline(archivo, registro, '|');  
                blue=atoi(registro.c_str());  
                Neurona *nuevo = new  
                Neurona(id, voltaje, posicion_x, posicion_y, red, green, blue)  
                ;  
                lista.push_back(*nuevo);  
            }  
        }  
        QMessageBox::information(this, "Recuperado", "Se  
recuperaron los archivos correctamente...");  
    } else {  
        QMessageBox::critical(this, "Error", "El archivo  
no se abrio correctamente...");  
    }  
}
```

## Resultados



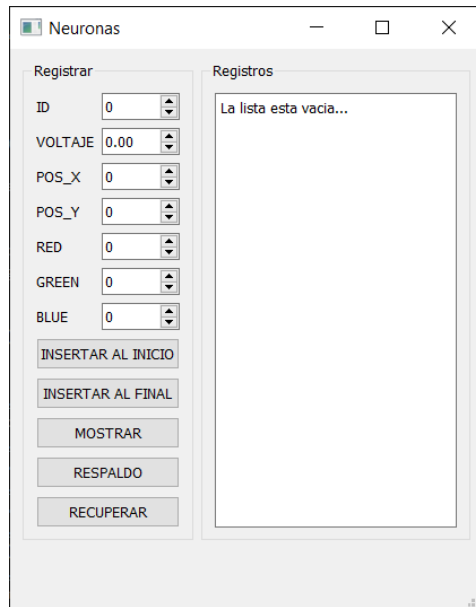
*Guardamos los tres registros en un archivo .txt*



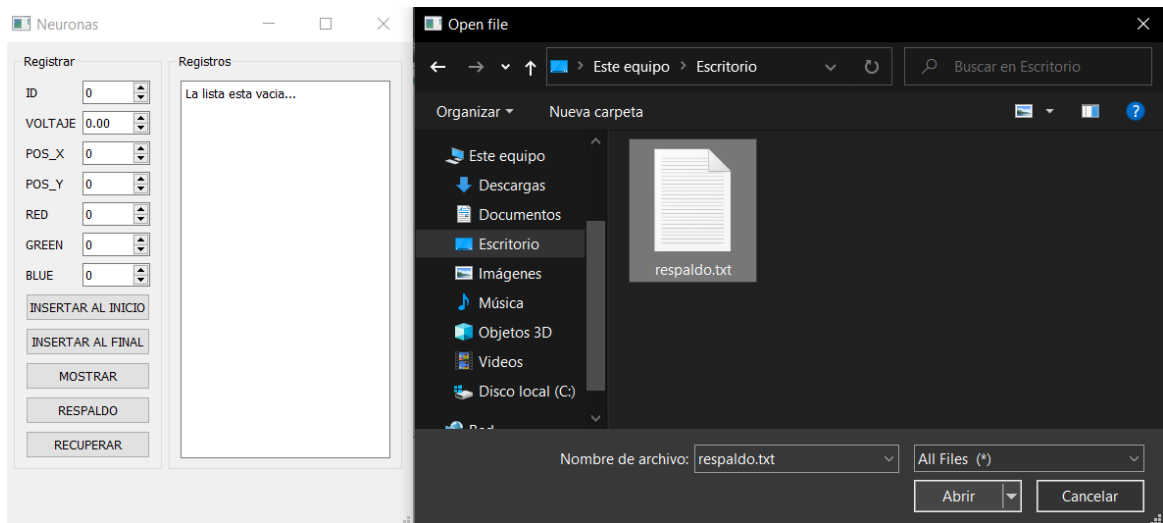
*Se guardaron correctamente las neuronas*

Cerramos el programa y volvemos abrir.

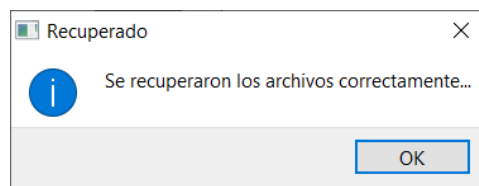
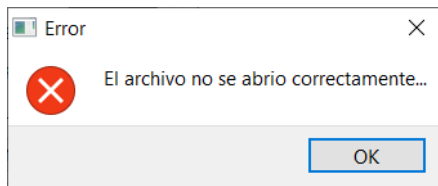
Seminario de Solución de Problemas de Algoritmia  
Becerra Ruan Juan Martin  
220286911



*Muestra la lista vacía.*



*Se abre la ventana para seleccionar el archivo para recuperar, cuando el archivo no se abre muestra una ventana de error y si se abre muestra un mensaje de recuperado.*



ID	VOLTAGE	POS_X	POS_Y	RED	GREEN	BLUE
3	67.1	8	14	85	34	85
2	45.4	78	8	97	4	97
1	14.7	12	45	47	78	47

*Se recuperaron correctamente*

## Conclusión

En esta práctica me gusto el uso de las ventanas emergentes para poder abrir y guardar archivos con una interfaz gráfica. Pero se necesita cuidado al momento de leer un archivo, ya que lee por string y se tienen que convertir a el tipo de dato que es requerido. Y así se solucionan los problemas de lectura del archivo.