

# UNIVERSIDAD POLITECNICA SALESIANA

**NOMBRE:** GADIEL PAHUANQUIZA

**CARRERA:** CIENCIAS DE LA COMPUTACION

**FECHA:** 2022-06-14

**MATERIA:** PROGRAMACION ORIENTADA A OBJETOS

## DEBER-CONSULTA

- **MANEJO DE ARCHIVOS DE TEXTO (QTextStream) y archivos binarios (QDataStream).**

### -QTextStream

El manejo de archivos en resumidas palabras nos proporciona una interfaz en el cual podemos leer y escribir de una mejor forma, para ser más técnicos podemos decir que nos ayudan a escribir palabras, líneas y números de manera cómoda, ya que cuenta con algunas opciones como lo es el formato para el relleno, la alineación de campos y el formato para los números, ahora al momento de implementarlo y trabajar en los constructores del QTextStream podemos realizar la configuración donde esta operando el mismo, por ejemplo creamos métodos booleanos de entrada y salida el cual se encarguen de leer datos, en el cual nos va a dar un valor verdadero cuando todos los datos que se ingresaron hayan sido leídos y caso contrario su resultado será falso, con todo lo dicho al final el QTextStream se encargara de borrar todos los datos registrados anteriormente.

Existen tres formas de leer los archivos del QTextStream, uno de ellos tiene por nombre “trozo a trozo”, otra forma es la llamada “palabra por palabra”, sin embargo, aquí se encuentran con limitaciones tanto en los espacios, pero con los espacios que estén en blanco este se encargara de omitirlos de forma automática, y por ultimo tenemos al que se lo llama “carácter por carácter” en el cual se encarga de analizar los archivos que entran para un mejor manejo, esto lo hace de forma independiente por cada carácter que encuentre junto con la semántica final de cada línea.

Con todo lo visto ahora tenemos la biblioteca del QTextStream el cual cuenta con varias funciones de nivel mundial y al ser en C++ se mantiene.

- **QTextStream::FieldAlignment**

Indicadores que nos permiten la alineación de campos en el texto.

CONSTANTE	DESCRIPCION
QTextStream::AlignLeft	Almohadilla en el lado derecho de los campos.
QTextStream::AlignRight	Almohadilla en el lado izquierdo de los campos.
QTextStream::AlignCenter	Almohadilla a ambos lados del campo.
QTextStream::AlignAccountingStyle	Igual que AlignRight, excepto que el signo de un número está al ras a la izquierda.

- **QTextStream::NumberFlag**

Indicadores que nos permiten configurar la salida de numeros enteros y decimales.

CONSTANTE	DESCRIPCION
QTextStream::ShowBase	Muestra la base como un prefijo
QTextStream::ForcePoint	Se encarga siempre de separar el decimal en números, incluso si no hay decimales.

QTextStream::ForceSign	Se encarga siempre de colocar el signo en números, incluso para números positivos.
QTextStream::UppercaseBase	Utiliza versiones en mayúsculas de los prefijos base
QTextStream::UppercaseDigits	Utiliza letras mayúsculas para expresar los dígitos 10 a 35 en lugar de minúsculas.

- **QTextStream::Status**

Indicadores que nos muestra el estado del flujo de texto de forma actual.

CONSTANTE	DESCRIPCION
QTextStream::Ok	El flujo de texto funciona de manera correcta.
QTextStream::ReadPastEnd	El flujo de texto ha leído más datos finales del dispositivo adyacente.
QTextStream::ReadCorruptData	El flujo de texto ha leído datos corruptos.
QTextStream::WriteFailed	El flujo de texto no puede escribir en el dispositivo adyacente.

Existen muchas otras más pero estas son las que consideramos las más importantes por ahora.

### -Archivos binarios (QDataStream).

El QDataStream se usa para el QTextStream para el flujo del texto. Este se encarga de brindarnos la serialización de los datos binarios, para esto debemos tener claro que los mismos que no dependen del sistema operativo por ningún modo así como tampoco del dispositivo que se esté ocupando, como dijimos anteriormente al ser un flujo de datos de manera binaria podemos escribir y leer los datos directamente, así como también codificarlos como son las videos, audios, entre otras cosas.

QDataStream tiene un libre acceso a datos de C++ como lo son algunos atributos como int, char, double entre otros. Se encarga de dividir las clases que están bien estructuradas o complejas en unidades de almacenamiento más pequeñas y básicas de esta clase principal.

Podemos leer, escribir y modificar los archivos. Un dato importante en todo esto es la recomendación de usar siempre números enteros, ya que al momento de ejecutarlo en otra plataforma o compilador puede ser que llegue a tener errores y toque implementar más pasos de los normales debido a los decimales, para las enumeraciones el QDataStream se encarga de colocar las mismas, sin la necesidad que hagamos cada uno de nosotros de forma manual. Por todos los cambios que ha tenido el formato binario se debe asegurar siempre que se trabaje en la misma versión del flujo ya que muchas veces se tiene el problema de que no se puede leer ni escribir el archivo por la diferencia de versiones.

Para la lectura y escritura de datos sin la necesidad de el uso del flujo de datos de manera directa, lo podemos hacer mediante un char \* y preasignando un “readRawData” para leer el archivo y un “writeRawData” para la escritura de datos, así como también la eliminación o codificación de los mismos

### -BIBLIOGRAFIAS

- QTextStream Class | Qt Core 6.3.2. (s. f.). Qt Documentation. Recuperado 14 de junio de 2022, de <https://doc.qt.io/qt-6/qtextstream.html>
- User Interface Design Basics | Usability.gov. (s. f.). usability.gov. Recuperado 8 de junio de 2022, de <https://www.usability.gov/what-and-why/user-interface-design.html>
- Qt - Clase QDataStream - La clase QDataStream proporciona serialización de datos binarios a un QIODevice - Español. (s. f.). Runebook.dev. Recuperado 14 de junio de 2022, de <https://runebook.dev/es/docs/qt/qdatastream>