

# Diario di lavoro

Luogo	SAM Trevano
Data	08.11.2019

## Lavori svolti

Per prima cosa ho fatto degli insert dalla view della descrizione, perché la scorsa lezione mi ero dimenticato che anche se è una view separata da QdcView, scrive nella stessa tabella (Qdc). Gli insert funzionano anche da questa view. Dopo aver finito questo piccolo test, sono tornato a lavorare sul metodo per leggere (e scrivere) dei file Word. Siccome le soluzioni trovate ieri non funzionavano molto bene ed erano sviluppate in un' applicazione Console, ho deciso di cercare qualcos'altro. Ho trovato un modo interessante per aprire un file, leggerlo, copiarlo e modificarlo. Questa soluzione l'ho trovata al link: <https://www.codeproject.com/Articles/36432/Edit-Microsoft-Word-Documents-From-Your-Application?msg=4921673#xx4921673xx>

Per testare questo codice, ho creato un ulteriore nuovo progetto, di tipo Windows Form e ho inserito in una grafica semplice due textbox e un bottone. Al click del bottone viene generato il seguente metodo:

```
private void eseguiBT_Click(object sender, EventArgs e)
{
    // copia del file
    File.Copy("C:\\Users\\lucas\\...\\prova.doc",
        "C:\\Users\\lucas\\...\\provatemp.doc", true);
    // valori predefiniti per l'apertura del file
    object missing = Missing.Value;
    // crea l'oggetto che contiene l'istanza di Word
    Word.Application wordApp = new Word.Application();
    // crea l'oggetto che contiene il documento
    Word.Document aDoc = null;
    // oggetto che definisce il file copiato (e da modificare)
    object filename = "C:\\Users\\lucas\\...\\provatemp.doc";
    // Se il file esiste
    if (File.Exists((string)filename))
    {
        object readOnly = false;
        object isVisible = false;
        wordApp.Visible = false;
        // apertura del file copiato
        aDoc = wordApp.Documents.Open(ref filename, ref missing,
            ref readOnly, ref missing, ref missing, ref missing,
            ref missing, ref missing, ref missing, ref missing,
            ref missing, ref isVisible, ref missing, ref missing,
            ref missing);
        aDoc.Activate();
        // richiama la funzione FindAndReplace passandogli due parametri,
        // il testo da sostituire e con cosa sostituirlo
        this.FindAndReplace(wordApp, "<Date>", nomeTB.Text);
        this.FindAndReplace(wordApp, "<Name>", valTB.Text.Trim());
        // salva il file
        aDoc.Save();
    }
    else
    {
        MessageBox.Show("File does not exist.", "No File", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
    }
}
```

In poche parole il metodo copia il file da leggere, lo apre e richiama una funzione che si occupa di sostituire il contenuto del file con il contenuto delle TextBox (Nel codice è inserito in un blocco try {} catch{} per gestire gli errori. Questo metodo è funzionante. Per poter usare le librerie che ho usato ho inserito la seguente referenza:

```
using Word = Microsoft.Office.Interop.Word;
```

Siccome ci ho messo meno del previsto a trovare una soluzione che possa essere utilizzabile ed efficiente anche per il mio progetto. Ho deciso di creare un file che faccia da template a tutti i Qdc. In questo modo quando implementerò queste funzioni, l'app andrà a prendere i dati dal database e li sostituirà nel file di template. Il file di template è strutturato in questo modo:

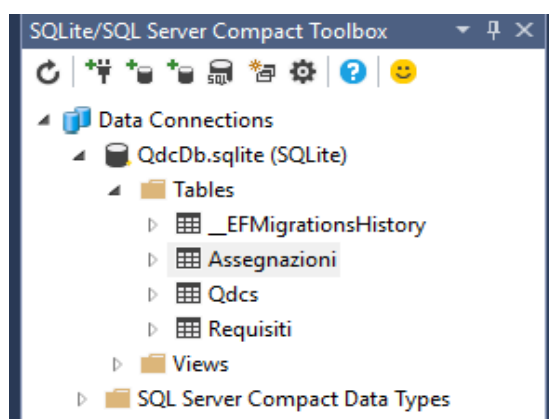
Superiore professionale	Nome: <NomeFormatore>	Cognome: <CognomeFormatore>
	<EmailFormatore>	
Perito 1	Nome: <NomePerito>	Cognome: <CognomePerito>
	<EmailPerito>	
Perito 2	Nome:	Cognome:
Periodo	<DataInizio> , <DataConsegna>	
Orario di lavoro	<Orainizio> - <OraFine>	
Numero di ore	<NumOre>	
Identificatore	Assegnazione	

Praticamente è strutturato come un qdc senza ancora nessuna informazione. Le parti modificabili sono indicate tra "<>" e sono evidenziate in grigio. Questo è solo un esempio, nel file sono presenti altre parti modificabili come il titolo, la descrizione e ovviamente i requisiti. Nel caso sia necessario inserire più informazioni attraverso il form, basterà inserire altri campi modificabili in questo singolo file.

Dopodichè sono tornato nel progetto principale e cercando il database tramite il server explorer di Visual Studio mi sono ricordato che vengono mostrati solo i database istanziati con SqlServer. Ho chiesto a Matteo e a Gabriele se per caso usassero SQLite e Gabriele mi ha detto che cercando in internet c'è un tool che permette di visualizzare i database SQLite nel explorer. Ho installato il tool ufficiale di Visual Studio **SQLite/SQL Server Compact Toolbox** al link:

<https://marketplace.visualstudio.com/items?itemName=ErikEJ.SQLiteServerCompactSQLiteToolbox>

Riavviando Visual Studio nel Server Explorer ora è possibile scegliere SQLite come tipo di database:



Infine ho aggiornato la documentazione con parte del lavoro svolto questa settimana.

#### Problemi riscontrati e soluzioni adottate

Inizialmente ho testato l'applicazione con un file .doc e non ho riscontrato nessun problema. Quando mi sono accorto di non aver usato un file docx ho modificato l'estensione e ho notato che anche in questo caso viene fatta una copia e avviene la modifica del file, però il file è apribile solo in modalità lettura. Fortunatamente a differenza della soluzione testata ieri il file non si chiude immediatamente. Guardando i permessi del file appena creato ho notato che sono identici al file di partenza, quindi non è quello il problema.

Alla fine ho provato a mostrare i dati del database nella view QdcView come test per vedere se le select (metodo Get) funzionassero. I dati non vengono mostrati nonostante l'uso dei Binding. Non ho ancora risolto questo errore.

#### Punto della situazione rispetto alla pianificazione

Purtroppo a causa di una leggera labirintite l'ultima settimana prima delle vacanze sono stato assente. Per questo motivo ho perso ben 12 ore di progetto. Al momento ho deciso di cercare di recuperare il meno possibile a casa perché non è questo lo scopo del progetto, se a lungo andare mi accorgerò non riuscire a recuperare in tempo ovviamente impiegherò anche del tempo a casa per finire il progetto.

Nonostante le ultime due lezioni abbia fatto molti passi avanti, non ho recuperato molte ore in base al gantt preventivo.

#### Programma di massima per la prossima giornata di lavoro

Risolvere l'errore sulle select e quello che impedisce di editare i file salvati dal codice in formato docx cominciare l'implementazione della soluzione trovata oggi nel progetto principale (riuscire a leggere il file dei requisiti, che è molto più grande di quello testato da me e riuscire a trovare i requisiti partendo dal titolo).