



# ***TIROCINIO FORMATIVO***

***ITS APULIA DIGITAL MAKER – DEVELOPER 4.0 MOLFETTA 22/24***

***Christian De Nicolo***

***Tutor -> Pasquale De Tullio***

# INTRODUZIONE A ENI PRIMA



Eni Prima è un progetto, che ha come clienti/business l'azienda Eni, di una gara con durata quinquennale, fondato su 2 attività:

- Assistenza;
- Sviluppo a task.

In particolar modo si basa sulla gestione del rischio in ambito energetico con vari applicativi che comunicano tra loro. L'obiettivo principale è quindi quello di produrre della documentazione, con strumenti di reportistica, e far sì che i dati siano conformi.

Uno tra questi è QlikView, ovvero un applicativo utilizzato per la reportistica che consente di sviluppare e fornire rapidamente applicazioni di analisi guidata interattive.

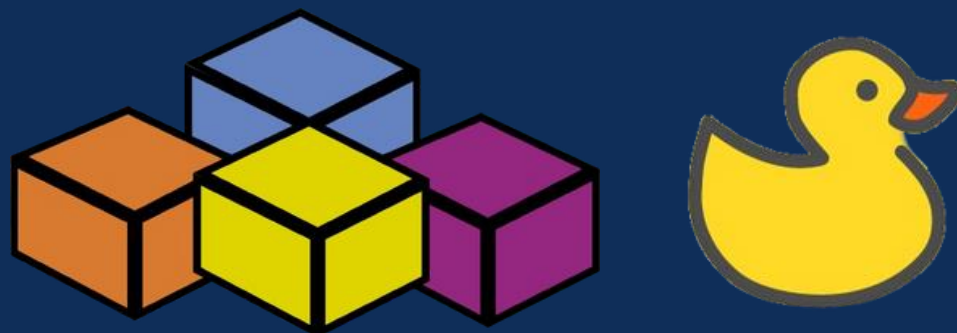


# VISUAL BASIC FOR APPLICATIONS

Come prima parte del mio percorso di tirocinio ho lavorato con il linguaggio di programmazione **VBA Excel** per la costruzione di **Macro**, cioè delle mini applicazioni che consentono di automatizzare delle attività eseguite di frequente in modo da risparmiare tempo all'utente.

In particolar modo, mi sono occupato dell'estrazione di JSON di input da varie macro utilizzando VBA e un altro software open-source chiamato **Rubberduck**, che mi ha permesso di esportare le macro su un repository locale, in modo da avere un "pezzo di storia" con le modifiche correnti prima di modificarlo in qualcosa di nuovo.

L'obiettivo principale di queste operazioni è stato quello di migrare algoritmi da un applicativo all'altro. Uno tra questi è **AaaS** (Algorithm as a Service), ovvero il principale applicativo dove vengono archiviati algoritmi.

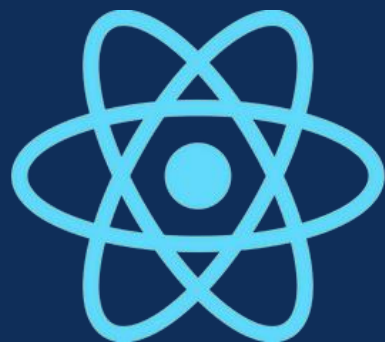


# EVOLUTIVA DI FRONT-END



Questo task consisteva nell'andare a creare un "tab" completo nella sezione "uploader" di un altro applicativo chiamato **ERS-Scenario**, per lo sviluppo di scenari istituzionali, utilizzando Javascript con il framework di **React** con **Visual Studio Code** come IDE.

Per accedere e lavorare su questo applicativo ho utilizzato un progetto già esistente con varie pagine JavaScript XML. Successivamente, attraverso l'interfaccia a riga di comando di Powershell, in cui ho lanciato un container su **Docker Desktop**, ho compilato e startato il Node Package Manager che mi ha permesso di aprire una porta locale su cui runnare l'applicativo, in questo caso la 3000.

A screenshot of the 'Upload File' interface in the ERS-Scenario application. The interface is white with a light gray border. At the top left, there is a tab labeled 'Upload File' with a close button. Below the tab, there is a 'Scenario Type' dropdown menu with 'ISTITUZIONALE' selected. Below that is a 'Label' text input field. To the right of these fields is a large dashed rectangular box with the text 'Drag files here...'. Below the 'Label' field is a checkbox labeled 'Overwrite?'. At the bottom, there is a small text block with a note: 'N.B: Gli scenari MID-DAY precalcolati sono solo un caso particolare gestito in modo differente. Bisogna inserire all'inizio della label la stringa [PRECALCOLATO] seguito da del testo a piacere. Sarà vostra premura non caricare scenari precalcolati che possono andare in conflitto con gli scenari standard/calcolati.'



# ***GIT E I REPOSITORY***

Un altro strumento fondamentale per lo sviluppo che ho avuto modo di utilizzare durante il tirocinio è la tecnologia **Git**, ovvero un software open-source per il controllo delle versioni, che mi ha permesso di dividere il codice in rami/branch ognuno con codice diverso, andandolo a salvare per step (commit) e molto altro, sia su riga di comando con la console del Git Bash che su interfaccia grafica utilizzando l'IDE di Visual Studio Code.

Questo in ambito locale, per quanto riguarda in remoto, ho avuto l'occasione di utilizzare **GitHub** che avevo già utilizzato durante il corso ITS e **GitLab**, che sono dei repository/servizi web cloud-based che mi hanno permesso di pushare e viceversa pullare codice e progetti, visibili da tutti, in modo da creare una gestione più coordinata dei dati.

Tuttavia, ciò che mi ha permesso di condividere il lavoro di front-end con tutto il team di sviluppo è stato **Git Extensions**, ovvero uno strumento/applicativo UI per la gestione dei repository git, che è in grado sfogliare la cronologia dei commit con i rami che vengono visualizzati utilizzando un grafico che evidenzia, per l'appunto, i commit.





***GRAZIE DELL'ATTENZIONE!***