



**UNIVERSITÀ  
DEGLI STUDI  
DI BERGAMO**

Dipartimento di  
Ingegneria

Corso di laurea in  
Ingegneria Informatica

Classe n. L-8 - Classe delle lauree in ingegneria  
dell'informazione

## Applicazione per la fusione dei dati dalla scena del crimine

Candidato:  
*Michele Verdi*

Matricola n.  
1067606

Relatore:  
*Chiar.mo Prof.  
Angelo Michele Gargantini*

Correlatore:  
*Dott. Giorgio Felizzato*

Anno Accademico  
2022/2023



## Sommario

|   |           |
|---|-----------|
| <b>1. Introduzione .....</b>                            | <b>3</b>  |
| <b>2. Software Lifecycle .....</b>                      | <b>5</b>  |
| <b>3. Software Configuration management .....</b>       | <b>7</b>  |
| <b>4. People management and team organization .....</b> | <b>9</b>  |
| <b>5. Software Quality .....</b>                        | <b>11</b> |
| <b>6. Requirement Engineering .....</b>                 | <b>13</b> |
| <b>7. Modeling .....</b>                                | <b>15</b> |
| <b>8. Software Architecture .....</b>                   | <b>17</b> |
| <b>9. Software Testing .....</b>                        | <b>19</b> |
| <b>10. Software Maintenance.....</b>                    | <b>21</b> |
| <b>11. Software Design .....</b>                        | <b>23</b> |
| <b>12. Codice .....</b>                                 | <b>25</b> |
| <b>13. Fonti.....</b>                                   | <b>29</b> |
| <b>14. Ringraziamenti .....</b>                         | <b>31</b> |



# 1. Introduzione

Il comportamento dei terroristi nell'uso degli esplosivi è cambiato negli ultimi anni di attentati: esplosivi improvvisati con sostanze chimiche presenti sul mercato stanno prendendo il posto di esplosivi commerciali e militari.

Il Parlamento Europeo e il Consiglio Europeo, il 15 gennaio 2014, hanno adottato la legge n. 98/2013 sulla vendita e l'utilizzo di sostanze chimiche potenzialmente utilizzabili per la produzione di esplosivi.

Secondo questa legge 7 sostanze chimiche non sono disponibili a tutti gli acquirenti se comprate in grandi dimensioni, invece altre sostanze potranno essere comprate ma verranno monitorate in caso di acquisti sospetti da parte del pubblico.

|   | ANNEX I Regulated Substance | Limit Value (w/w) |
|---|-----------------------------|-------------------|
| 1 | Hydrogen peroxide           | 12%               |
| 2 | Nitromethane                | 30%               |
| 3 | Nitric acid                 | 3%                |
| 4 | Potassium chlorate          | 40%               |
| 5 | Potassium perchlorate       | 40%               |
| 6 | Sodium chlorate             | 40%               |
| 7 | Sodium perchlorate          | 40%               |

Le tracce di queste sostanze usate per la produzione di esplosivi (particolati, vapori e sostanze a base d'acqua) presenti nell'ambiente, indicano la vicinanza di una "fabbrica di bombe" e permettono alle forze di polizia di fare controlli dove vi è il sospetto della produzione di esplosivo.

Questo metodo per proteggere i cittadini dagli attentati esplosivi è considerato molto efficace rispetto ai semplici pattugliamenti fatti ad un possibile target.

Il progetto LOTUS e il progetto BONAS hanno sviluppato dei sensori per rilevare la presenza di sostanze chimiche utilizzate per la produzione di esplosivi.

Il sistema di controllo permette: il rilevamento di una sostanza in base ai dati raccolti da un singolo sensore, il rilevamento di una sostanza in base ai dati raccolti da più sensori e la produzione di un livello di allarme globale basato sulla fusione dei dati provenienti da tutti i sensori presenti nella rete.



## 2. Software Lifecycle

Date le circostanze e le caratteristiche dei componenti del gruppo è stato deciso di utilizzare un modello di tipo Agile.

Il modello Agile, come sottolineato dal manifesto agile per lo sviluppo del software: Considera più importanti gli individui e le loro interazioni a discapito dei processi e dei tools. In questo modo si enfatizza l'utilità di ciascun membro del team;

Preferisce un software funzionante rispetto ad una documentazione comprensiva, infatti prende poco in considerazione la documentazione e si basa molto sulla conoscenza tacita dei membri del team;

Favorisce una collaborazione ed un dialogo con il cliente che può essere più leggera e volubile, rispetto alla presenza di contratti e accordi che spesso sono troppo rigidi ed influenzano in modo negativo il lavoro del team.

Infine è molto più importante rispondere al cambiamento, che può verificarsi, durante la fase di un sviluppo di un ciclo, piuttosto che attenersi ad un piano che non varia mai durante la stesura del codice e quindi possa risultare obsoleto.

Il modello Agile utilizzato è quello di tipo Scrum.

Il modello Agile Scrum è composto da un Product owner che all'inizio ha redatto un product backlog che è una lista di punti che dovranno essere presenti all'interno dell'applicazione. Questa lista viene consegnata al team in modo tale che possa avere dei punti di partenza per la stesura dell'applicazione.

Durante la fase di sprint, il team prende qualche punto del product backlog ed inizia a discutere su come implementare queste parti nel modo più adeguato. Nel nostro caso le funzionalità dell'applicazione sono state sviluppate poco per volta, in modo tale che ci si concentrasse su una parte per volta.

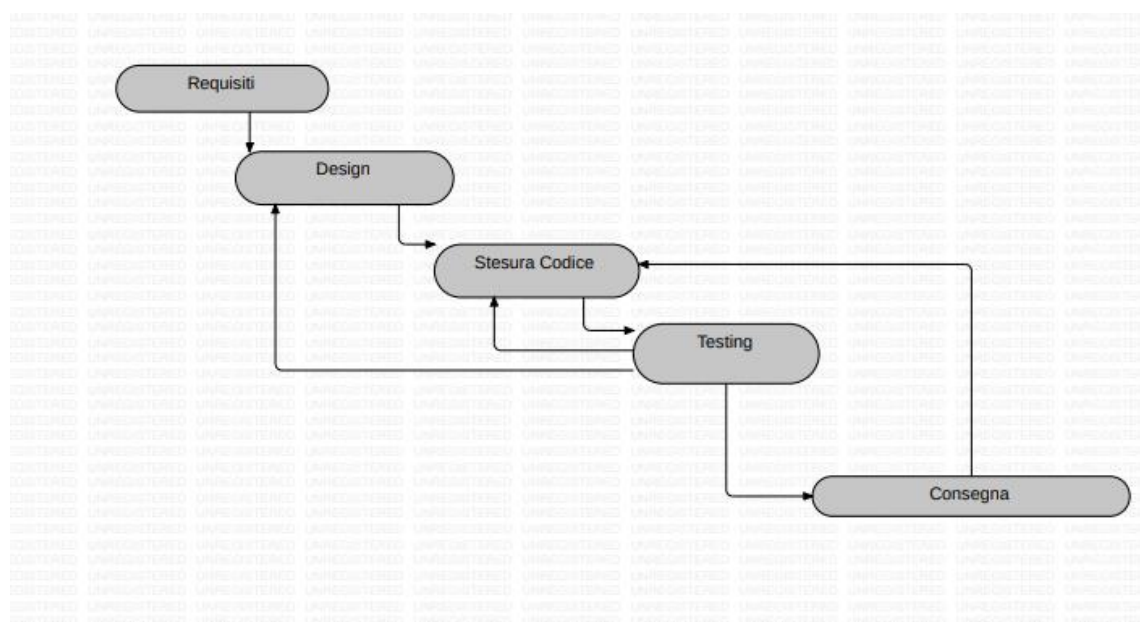
Nel modello Scrum potrebbe non essere tutto noto all'inizio di uno sprint, ma i processi e le esigenze vengono regolati in base alla necessità e in base alle informazioni che si ottengono durante il processo di sprint. Durante la stesura dell'applicazione ci sono state svariate volte in cui gli algoritmi sono cambiati dopo un dialogo tra i membri del team.

Il team ogni 2/4 settimane ha completato alcuni punti della lista e successivamente si è riunito ed ha continuato con gli altri.

A capo del team è presente un ScrumMaster che mantiene il focus sul progetto ed inoltre è a conoscenza in maniera dettagliata di tutto ciò che richiede la lista dei punti, in modo tale da poter fornire le informazioni corrette a tutto il team.

Questa routine di analisi dei punti, discussione tra i membri del team e stesura del codice, procede fino a quando l'applicazione non è terminata e tutti i punti del product backlog sono esauriti.

Quindi alla fine il software lifecycle è composto da 5 fasi che in base alle difficoltà riscontrate durante la costruzione dell'applicazione, si sono sovrapposte l'una sopra l'altra in modo tale da non tralasciare alcun particolare e lavorare al meglio.





### **3. Software Configuration management**

Durante lo sviluppo dell'applicazione è stata importante la condivisione del materiale sia per quanto riguarda la condivisione del codice, che per l'invio di file Excel per il lineare funzionamento dell'applicazione.

Questa condivisione è stata fatta mediante l'utilizzo di Google drive, per quanto riguarda i file Excel, e mediante l'utilizzo di Github e mail, per quanto riguarda la condivisione del codice.

Github è un servizio utile per il passaggio di codice perché ci ha permesso di condividere una cartella con tutti coloro che hanno fatto parte del progetto, ed inoltre può essere condivisa in un futuro, con ulteriori persone che andranno a costruire ulteriori parti del progetto.

Github ci ha consentito, tramite la sua funzione "Commit", di caricare il codice, scritto in python su Visual Studio Code, in modo tale da renderlo disponibile a tutti.

Gli altri membri del team, presenti all'interno della cartella Github, non devono fare altro che entrare all'interno di quest'ultima, prendere il codice selezionando la funzione "Clone repository" che permette di scaricare la cartella sul computer, tenendola sempre aggiornata con la funzione "Pull".

Dopo questi passaggi i membri del team avranno a disposizione il codice ed inoltre tutte le parti coinvolte all'interno del progetto potranno modificare il codice e condividerlo nuovamente.

Inoltre Github ci ha consentito la creazione di "issues" che vengo utilizzate per indicare quando si hanno problemi con il codice e serve una mano da parte di un altro membro del team per risolverlo.

Al completamento dell'applicazione si è ricorso inoltre all'utilizzo ulteriore di Google drive per la condivisione del file zip, contenente il file eseguibile "home.py".

Questo eseguibile è stato creato per permettere la fruizione dell'applicazione, anche a coloro che non possiedono competenze nell'utilizzo di python e Visual Studio Code, ma vogliono usufruire dell'applicazione.



## **4. People management and team organization**

Dato che il nostro team, come detto precedente, ha deciso di intraprendere la strada di un processo Agile Scrum.

La sua struttura è così composta da:

Un Product Owner : Prof Angelo Michele Gargantini e Prof. Francesco Saverio Romolo

Uno Scrum Master: Dr. Giorgio Felizzato

Due Sviluppatori: Michele Verdi, Dr. Giorgio Felizzato.

Il Product Owner deve saper trasmettere al team ciò che vogliono i clienti, deve prendere tutte le decisioni utili per il completamento del progetto e deve essere disponibile con il team in modo tale da verificare che il prodotto venga implementato correttamente seguendo le indicazioni presenti nel product backlog.

Lo Scrum Master aiuta il team a rimanere concentrato sul lavoro e aiuta la collaborazione tra ogni membro del team. Deve incontrarsi settimanalmente, come è stato fatto per la maggior parte delle volte lungo la costruzione dell'applicazione.

Il team degli sviluppatori è composto da 2 componenti che tramite la loro interazione sono stati in grado di costruire l'applicazione step-by-step, aiutandosi e fornendosi informazioni a vicenda.



## 5. Software Quality

L'applicazione rispetta numerose qualità di McCall tra cui: correttezza, affidabilità, efficienza, integrità, utilizzabilità, testabilità, portabilità e riutilizzabilità

La correttezza indica la misura con cui il programma soddisfa le sue specifiche e soddisfa gli obiettivi dell'utente. L'applicazione ha esaudito tutte le richieste fatte dal Product owner.

L'affidabilità indica la misura con cui un programma ci si aspetta che performi la sua funzione prevista con la precisione richiesta. L'applicazione, con una grande quantità di dati, performa in modo preciso e fornisce output puntuali in base alle richieste fatte.

L'efficienza indica l'ammontare delle risorse computazionali e del codice richiesto da un programma per performare una funzione. Questa caratteristica è garantita dall'applicazione, grazie all'utilizzo di numerose librerie compatibili con python e alle funzioni scritte dagli sviluppatori, che hanno aiutato a snellire il codice

L'integrità indica la misura per cui ogni accesso al software o ai dati da parte di persone non autorizzate, viene controllato. L'applicazione è condivisa su github e il suo eseguibile può essere condiviso tramite un file zip quindi l'accesso è totalmente controllato. Inoltre non ha alcun collegamento ad internet o a database esterni e quindi non si possono avere ulteriori accessi da parte di terzi, se non voluti dai possessori dell'applicazione.

L'utilizzabilità è lo sforzo richiesto per imparare, operare, preparare input e interpretare output di un programma. Questa qualità è rispettata in quanto per utilizzare l'applicazione non servono numerose competenze, grazie anche alla presenza di un file "istruzioni" all'interno dell'applicazione che può chiarificare ogni dubbio. Chiaramente l'output dei file non fornisce grosse informazioni a coloro che utilizzano l'app senza conoscere un minimo i dati di cui tratta il progetto.

La testabilità è lo sforzo richiesto per testare un programma per assicurarsi che performi le sue funzioni richieste. Per verificare che l'applicazione funzioni basta fare qualche prova inserendo alcuni file Excel e testare tutte le funzionalità rispettando le istruzioni di utilizzo date.

La portabilità è lo sforzo richiesto per trasferire un programma da un ambiente all'altro. L'applicazione essendo facilmente condivisibile, può essere trasferita da un ambiente all'altro tramite l'invio dello zip, oppure tramite la condivisione del codice presente su Github che dovrà essere aperto su un editor di codice che supporta python.

Infine abbiamo la riutilizzabilità che è la misura con la quale un programma può essere riutilizzato in altre applicazioni. Siccome il nostro codice è disponibile a tutti, è possibile copiarne funzioni che potranno essere liberamente trasportate in altre applicazioni.

## 6. Requirement Engineering

Il professor A. Gargantini e il professor F. Romolo durante una call online, hanno parlato del progetto RISEN, spiegando cosa avrebbero voluto dall'applicazione che avremmo dovuto produrre e le finalità che quest'ultima avrebbe avuto all'interno del progetto.

Dopo l'incontro conoscitivo, sono stati fatti numerosi incontri con il Dr. Giorgio Felizzato con il quale sono stati stabiliti gli effettivi obbiettivi dell'applicazione.

La richiesta era di produrre un'applicazione che fosse in grado di prendere dei dataset di tipo Excel e di fare una "data fusion" in modo tale da poter esaminare più dataset concatenati tra di loro.

Inizialmente abbiamo deciso di creare un'applicazione desktop che sarebbe stata resa disponibile a tutti tramite la distribuzione del file eseguibile, e per intraprendere questa strada sarebbero stati utilizzati la piattaforma VSCode, Python 3.12.2 e l'aiuto di numerose librerie tra le quali Tkinter, che è una libreria che viene utilizzata per la visualizzazione grafica e Pandas che è anch'essa una libreria, che viene utilizzata per il trattamento di file Excel e CSV.

L'applicazione deve essere in grado di poter scegliere più file Excel da inserire contemporaneamente, i quali dovranno poter essere modificati tramite la presenza di funzioni che permettano la cancellazione di righe o colonne, in caso in cui alcuni dati non siano del tutto corretti.

Inoltre l'applicazione deve poter visualizzare numerosi grafici che dopo aver analizzato i dati, restituiscano un output utile per poter verificare la validità del file Excel inserito. Per risolvere questa problematica abbiamo pensato di utilizzare le librerie Plotly e Seaborn che servono per la costruzione di grafici e la libreria Webview che serve per la visualizzazione dei grafici stessi.

In aggiunta ci sarebbero dovute essere numerose operazioni di tipo statistico che sono state eseguite grazie alla presenza di Pandas che tramite le sue funzionalità ci ha permesso di poter trattare i database inseriti senza alcuna difficoltà e con facile comprensibilità dei dati.

L'applicazione avrebbe dovuto garantire all'utente una grande libertà per quanto riguarda la scelta delle operazioni da svolgere e per risolvere questo problema abbiamo pensato di inserire alcune "combo box" a scelta multipla, sempre tratte da Tkinter, per poter permettere all'utente di sfruttare tutte le potenzialità dell'app decidendo autonomamente che operazioni svolgere.

Le ultime due richieste sono state di poter avere un output di alcuni risultati, rendendolo fruibile all'utente e la presenza di una pagina che contenesse delle istruzioni per come si sarebbero dovuti formattare i file Excel. Per il primo problema abbiamo pensato di poter far scaricare all'utente alcune tabelle create dalla nostra applicazione, tra le quali quella dei file concatenati, tramite l'utilizzo della libreria OS che ci è stata utile per rintracciare il percorso del desktop sul computer su cui stiamo facendo girare l'applicazione. In questo modo salviamo i file Excel, elaborati all'interno dell'applicazione, sul desktop del computer che si sta utilizzando. Per il secondo problema invece abbiamo pensato di inserire una pagina che riporta tutte le istruzioni utili per l'utilizzo dell'applicazione con l'aiuto della libreria PdfReader, che come si può intuire dal nome, ci permette di leggere un file pdf, il quale conterrà le istruzioni necessarie.

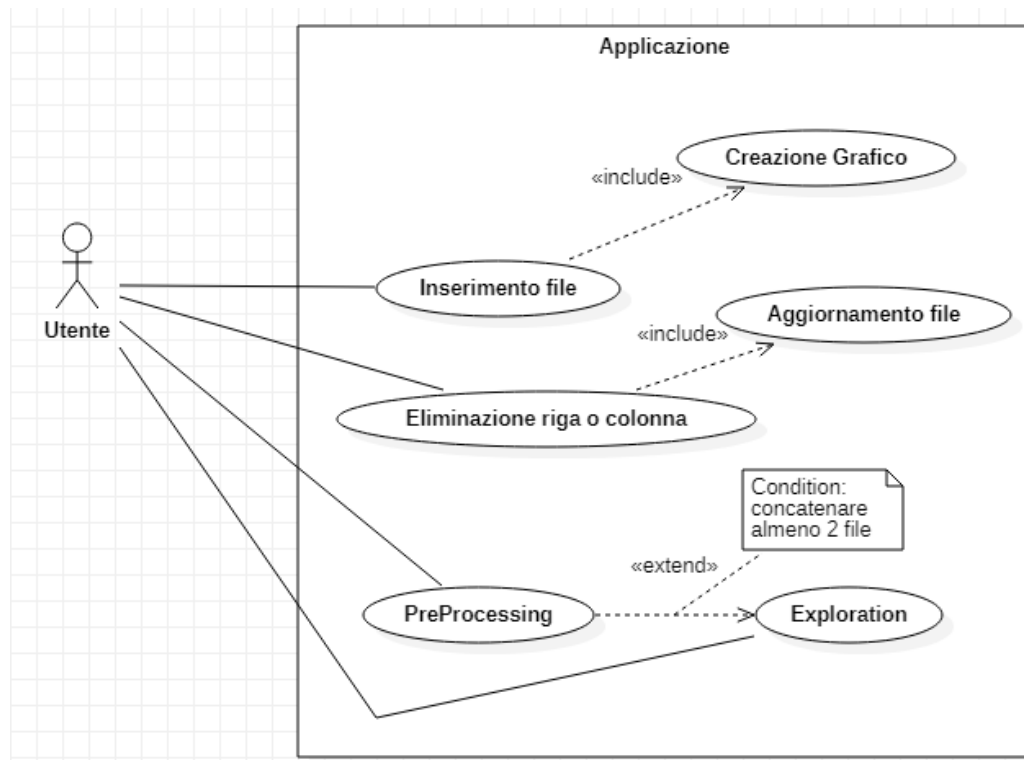
Chiaramente alcune idee espresse sopra hanno subito alcuni cambiamenti a causa della difficile realizzazione o della poca fattibilità e di conseguenza sono state modificate. Queste situazioni capitano spesso in un progetto Agile di tipo Scrum in modo tale da poter soddisfare comunque i desideri del committente, senza dover essere obbligati a seguire la strada stabilita all'inizio che potrebbe risultare tortuosa e poco redditizia al fine dello sviluppo del progetto.



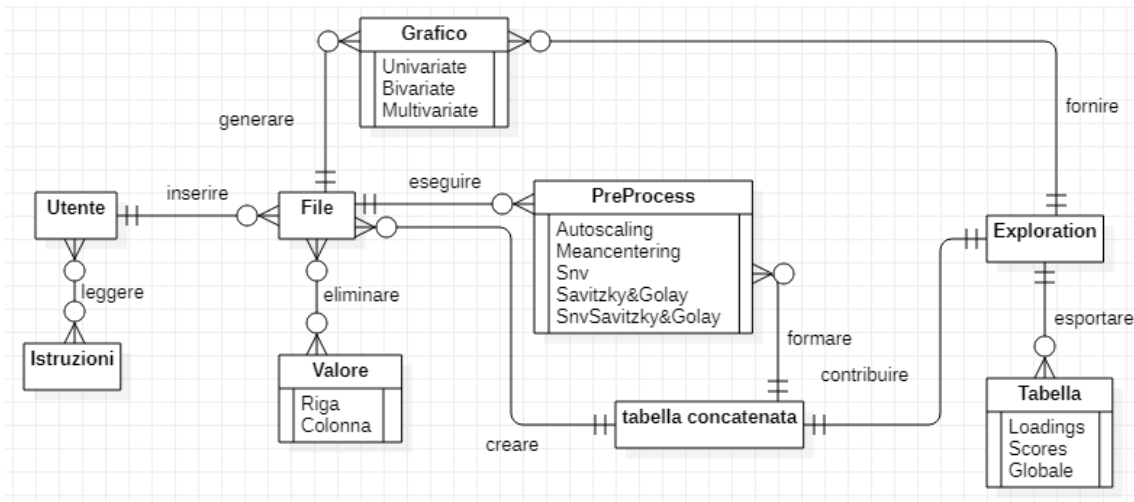
## 7. Modeling

I diagrammi, per spiegare la struttura dell'applicazione, sono stati costruiti tramite l'utilizzo di starUML e sono stati scelti in base alle loro funzionalità.

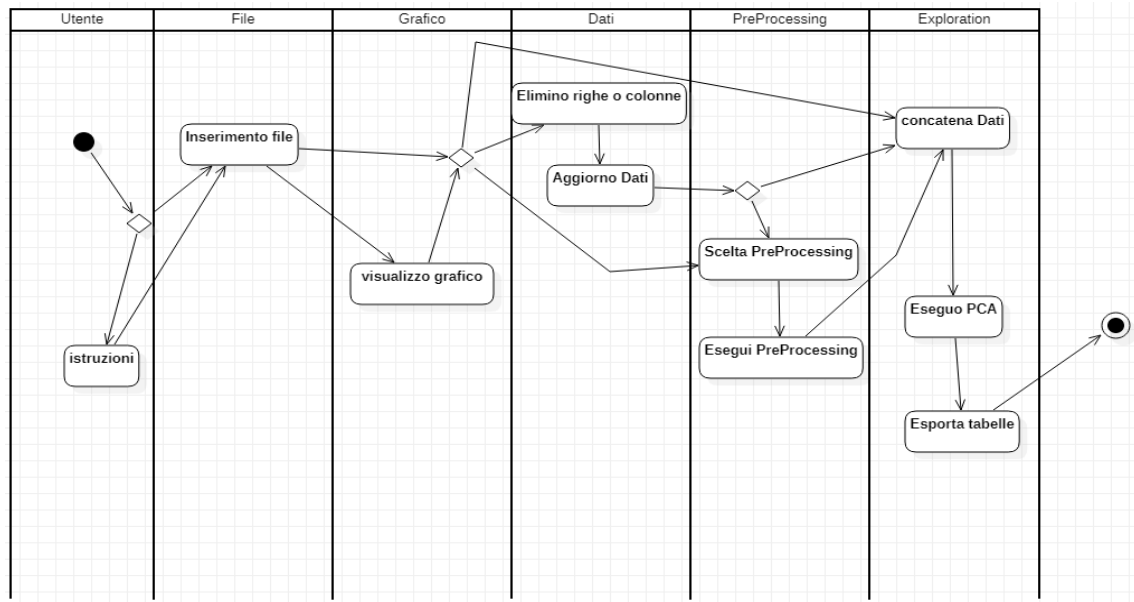
Lo Use Case Diagram viene utilizzato perché rappresenta il comportamento che ci si attende dalla nostra applicazione quando viene utilizzata in un determinato modo.



Il diagramma ER (Entità-Relazione) mostra come le entità, cioè le persone, gli oggetti e i concetti, interagiscono tra di loro. Questo diagramma aiuta lo sviluppatore a visualizzare le relazioni tra gli elementi chiave del software.



Infine l'Activity Diagram viene utilizzato per fornire una vista semplificata di ciò che accade durante un operazione o un processo. In questo diagramma si ha un pallino nero pieno che indica il punto di inizio del processo e un pallino nero pieno con attorno un cerchio, che indica la fine del processo.



## 8. Software Architecture

In questa fase è stato scelto il pattern architetturale più opportuno per il progetto. Il pattern serve per definire lo scheletro che dovrà avere l'applicazione, quali servizi dovranno entrare in gioco, le parti coinvolte e la logica di interazione tra di esse.

I pattern architetturali svolgono innanzitutto un ruolo di comunicazione tra gli stakeholders, ovvero i protagonisti del progetto, offrendo un'astrazione comune del sistema. I pattern offrono una base per la mutua comprensione, negoziazione e consenso, senza che sia richiesta conoscenza informatica dettagliata.

Un'architettura inoltre definisce l'organizzazione strutturale che dovrà essere rispettata, e potrà essere poi utilizzata come metro di valutazione del progresso del progetto.

Essa facilita inoltre la gestione del cambiamento e garantisce una maggiore robustezza del sistema.

Il pattern architetturale più congruo per l'applicazione è il Model-View-Controller conosciuto anche come MVC, è il pattern maggiormente utilizzato per applicazioni e nei frameworkweb.

Nel MVC i protagonisti sono tre: il model, la view e il controller. Il primo contiene le funzionalità e gestisce i dati, definendo gli algoritmi e le funzioni che descrivono la logica del programma. La view rappresenta la parte visiva, ed è con essa che si interfaccia l'utente. Infine, il controller si pone tra i due, da una parte gestendo l'input dell'utente e dall'altro fornendo le informazioni necessarie al modello. Se seguito adeguatamente, permette una totale distinzione tra le parti, che possono essere riutilizzate, modificate, sostituite senza incidere sulle altre.

Le dipendenze del progetto, chiamate anche dipendenze delle attività, sono relazioni tra le attività in base alla loro sequenza. Le attività dipendenti richiedono che una o più altre attività siano completate o avviate prima che il team possa iniziare a lavorarvi.

Le dipendenze del progetto sono una parte fondamentale della pianificazione del progetto. Avere una chiara sequenza di attività aiuta a garantire che i team completino il lavoro in modo corretto ed efficiente.

Tramite la funzione `Pipreqs`, già predefinita dal linguaggio python, è stato possibile estrarre tutte le librerie utilizzate durante la produzione dell'applicazione.

Molte di queste librerie erano già presenti all'interno di VSCode, invece molte altre sono state scaricate e inserite manualmente tramite la funzione `pip`.

Ecco l'elenco completo di tutte le librerie presenti all'interno del progetto:

```
altgraph==0.17.4      asttokens==2.4.1      attrs==23.2.0      backcall==0.2.0      beautifulsoup4==4.12.3
bleach==6.1.0         bottle==0.12.25       certifi==2024.2.2   cffi==1.16.0         charset-normalizer==3.3.2
clr-loader==0.2.6     colorama==0.4.6       contourpy==1.2.0    cycler==0.12.1       decorator==5.1.1
defusedxml==0.7.1    docopt==0.6.2         et-xmlfile==1.1.0   executing==2.0.1     fastjsonschema==2.19.1
fonttools==4.48.1    future==0.18.3        idna==3.6           imageio==2.34.0       ipython==8.12.3      jedi==0.19.1
Jinja2==3.1.3         joblib==1.3.2         jsonschema==4.21.1  jsonschema-specifications==2023.12.1
jupyter_client==8.6.0 jupyter_core==5.7.1    jupyterlab_pygments==0.3.0 kiwisolver==1.4.5
lazy_loader==0.3      MarkupSafe==2.1.5     matplotlib==3.8.2   matplotlib-inline==0.1.6 mistune==3.0.2
nbclient==0.9.0       nbconvert==7.16.2     nbformat==5.9.2     networkx==3.2.1       numexpr==2.9.0       numpy==1.26.4
openpyxl==3.1.2       packaging==23.2       pandas==2.2.0       pandastable==0.13.1   pandocfilters==1.5.1
parso==0.8.3          pefile==2023.2.7      pickleshare==0.7.5  pillow==10.2.0        pipreqs==0.5.0
platformdirs==4.2.0  plotly==5.18.0        prompt-toolkit==3.0.43 proxy_tools==0.1.0   pure-eval==0.2.2
pyparser==2.21        Pygments==2.17.2      pyimage==0.1.1      pyinstaller==6.3.0
pyinstaller-hooks-contrib==2024.0 pyparsing==3.1.1      pypdf==4.0.1        python-dateutil==2.8.2
pythonnet==3.0.3      pytz==2024.1          pywebview==4.4.1    pywin32==306          pywin32-ctypes==0.2.2
pyzmq==25.1.2         referencing==0.33.0    requests==2.31.0    rps-py==0.18.0        scikit-image==0.22.0
scikit-learn==1.4.0   scipy==1.12.0         seaborn==0.13.2     setuptools==69.0.3    six==1.16.0
soupsieve==2.5        stack-data==0.6.3     tenacity==8.2.3     threadpoolctl==3.2.0  tiffio==2024.2.12
tinycss2==1.2.1       tkinterhtml==0.7      tornado==6.4        traitlets==5.14.1     typing_extensions==4.9.0
tzdata==2023.4        urllib3==2.2.1        wcwidth==0.2.13     webencodings==0.5.1   wheel==0.42.0
xlrd==2.0.1           yarg==0.1.9
```

## 9. Software Testing

Nel progetto la prima fase di testing è stata svolta innanzitutto dal programmatore, per quanto riguarda il codice, invece la seconda parte è stata svolta dal Dr. Felizzato che con svariati tentativi ha verificato la validità degli output forniti e le varie funzionalità dall'applicazione.

Il programmatore nella sua fase di testing utilizza lo static testing, il quale si basa sull'utilizzo della console fornita da vsCode che indica automaticamente se ci sono problemi all'interno del codice. In tal caso si procede al controllo e alla correzione del codice fino a quando il compilatore non trova alcun problema e quindi il codice funziona.

A volte può capitare che nel codice ci siano dei malfunzionamenti che non creano problemi al compilatore e quindi in questo caso deve intervenire attentamente lo sviluppatore per correggere l'algoritmo che ha portato a quell'errore.

Una ulteriore problematica che ci può essere all'interno del codice dell'applicazione, è la presenza di bug che, se troppo numerosi, possono portare a malfunzionamenti.

Infine abbiamo la presenza dei cosiddetti warning emessi da parte del compilatore. Questi warning non creano alcun problema né a livello di codice né a livello di algoritmi. Uno dei principali warning che può capitare indica il fatto che si stia utilizzando delle funzioni python, che in un futuro prossimo potranno essere cambiate o deprecate e che quindi, in tal caso, bisognerà cambiare il codice.

Per quanto riguarda la seconda fase di testing: essa consiste nell'inserimento di svariati e diversi file excel, che vengono analizzati dal codice dell'applicazione. In questo caso il Dr. Felizzato dovrà testare tutte le funzioni dell'applicazione, controllare che non ci siano anomalie e che i risultati siano corretti.

Nel caso di errore nell'output, ha provveduto alla segnalazione nel più breve tempo possibile, così da permettere tempestivamente la correzione degli algoritmi che hanno portato all'errore.

Uno dei principali errori all'interno del codice è stata l'interazione tra i dati con le funzioni dei grafici e i calcoli statistici. Queste due componenti hanno portato a numerosi

errori dovuti all'incompatibilità dei dati, in quanto le funzioni statistiche operavano solo con una determinata tipologia di dato, il quale è stato sistemato appositamente per poter funzionare regolarmente.

Un ulteriore errore è dovuto al fatto che inizialmente, prima di entrare nella sezione di Exploration, premendo il suo Button, era necessario eseguire tutti i tipi di PreProcess, invece nel corso del tempo il problema è stato risolto con l'utilizzo di controlli che hanno dato la possibilità all'utente di entrare nella sezione Exploration senza alcun vincolo.

## 10. Software Maintenance

La manutenzione del progetto può essere fatta in due diverse fasi specifiche: la prima consiste nel sistemare il codice ancor prima che l'applicazione sia stata consegnata, snellendone alcune parti oppure eliminando parti superflue e la seconda invece avverrà dopo la consegna dell'applicazione.

Durante la stesura del codice sono state svolte numerose operazioni di refactoring. Il refactoring consiste nel cambiamento di alcune parti del codice perché risultavano troppo lunghe e quindi computazionalmente troppo complicate.

Ad esempio per quanto riguarda la visualizzazione delle tabelle tramite l'utilizzo di Tkinter, inizialmente sono state scritte numerose righe di codice che prendevano ogni riga del file excel e man mano venivano inserite una alla volta tramite dei cicli for. Successivamente, vista la lunghezza del codice, è stata inserita una funzionalità della libreria Pandastable che ci ha permesso di visualizzare tutto il file excel solamente con l'inserimento di una funzione.

Altra cosa importante in questa fase, è stata accorgersi della duplicazione di alcune parti di codice, di conseguenza sono state aggiunte alcune funzioni che vengono richiamate quando servono, senza dover riscrivere il codice che andrebbe ad appesantire l'applicazione.

Per quanto riguarda la seconda parte della manutenzione, essa viene svolta dopo il rilascio dell'applicazione. Spesso consiste nel migliorare alcune parti del codice, oppure nel cambiare alcune parti perché, a causa degli aggiornamenti del compilatore o del linguaggio, vengono deprecati alcuni metodi utilizzati durante la stesura del codice.

Tipicamente queste operazioni vengono svolte dallo sviluppatore che ha scritto il codice, o dall'azienda che ha venduto il codice. In questo caso le operazioni di refactoring verranno fatte eventualmente da persone terze. Per questo motivo sono stati inseriti numerosi commenti e spiegazioni all'interno del codice del progetto, per poter permettere a chiunque stia lavorando con il codice di capire appieno cosa ci sia scritto al suo interno e poter interagirci senza perdere troppo tempo per cercare quale parte dover revisionare.





## 11. Software Design

Il software design è una parte importante all'interno del progetto perché è utile per stabilire l'architettura dell'applicazione, la quale è suddivisa in tre parti: astrazione, modularità e complessità

All'interno del nostro progetto è presente un'astrazione procedurale denotata dal fatto che ogni procedura ha il proprio nome, dal quale si può comprendere quale sia la funzione all'interno del programma. Ne è un esempio la funzione autoscaling, la quale si occupa, come si nota dal nome, dell'autoscaling del dato inserito.

Per quanto riguarda la modularità esistono l'accoppiamento, che rappresenta l'interdipendenza tra i vari moduli dell'applicazione, e la coesione, che è il grado con il quale gli elementi del modulo sono funzionalmente correlati.

Nel nostro progetto ci sono alcune parti in cui il grado di accoppiamento è molto alto siccome per eseguire determinate operazioni, bisogna prima eseguirne altre, come nella funzione Exploration in cui è possibile eseguire le procedure solo dopo aver eseguito obbligatoriamente altre operazioni precedentemente, come l'inserimento, il PreProcess e la concatenazione. Invece ci sono altre parti del progetto in cui il grado di accoppiamento è basso. Ad esempio per eseguire la cancellazione delle righe e delle colonne oppure la visualizzazione dei grafici iniziali, si ha solamente la necessità che l'utente abbia inserito almeno un file.

Per quanto riguarda la coesione, la nostra applicazione garantisce un grado molto alto siccome all'interno della finestra di dialogo con il cliente, sono presenti in poche pagine, tutte le funzionalità del codice per garantire un utilizzo appieno dell'applicazione. La coesione dell'applicazione è sia di tipo sequenziale, in quanto alcuni output di alcune procedure vengono utilizzati come input da altre, che di tipo funzionale perché alcune procedure trasformano un input singolo in un output singolo. Tutte queste funzionalità variano in base alle decisioni prese dall'utente durante l'utilizzo dell'applicazione.

La complessità si riferisce all'ammontare di risorse che sono presenti all'interno del codice per la vera e propria costruzione dell'applicazione. Il codice scritto ha alcune pagine in cui sono presenti poche righe di codice come la pagina "home.py", invece ci

sono alcune pagine come “PreProcess.py” che contiene molte righe di codice composte da funzioni, anche nidificate, che vanno ad appesantire il programma. L’applicazione si è basata sulla presenza di moduli molto snelli, che però a loro volta chiamano ulteriori moduli. Questa scelta è dovuta al fatto che persone terze, nel caso dovessero prendere in mano il codice, possano comprendere facilmente ogni funzione scritta.

## 12. Codice

Il codice è stato scritto in python 3.12.2 con l'ausilio del compilatore VSCode. In questa parte è stato inserito qualche algoritmo che è stato scritto durante la stesura del codice.

### Concatena e controllo

```
def concatena(database, primo):
    global a
    database=database.drop(database.columns[1], axis=1)
    column_headers = list(database.columns.values)
    ID=column_headers[0]
    ID=str(ID)
    result= pd.merge(primo,database, on=[ID] )
    primo=result
    passaggionewarray(primo)
    newprimo={}
    newprimo[0]=primo
    stampaTabelle(newprimo,0, "tabella concatenata", True)
    passaggioPCA(primo)

def controllo(parametro):
    global nuovoarray, temp2
    if len(nuovoarray)==0:
        temp2[0]=parametro
        stampaTabelle(temp2,0,"tabella concatenata", True)
        passaggionewarray(temp2[0])
    else:
        temp2[0]=parametro
        concatena(temp2[0], nuovoarray)
```

Questo algoritmo è presente nella pagina scelta\_Exploration.py ed è stato costruito per permettere all'utente di concatenare più tabelle tra di loro. All'interno delle istruzioni del programma viene specificato che i file per essere concatenati tra di loro devono avere gli ID inseriti nella prima colonna del file ed inoltre devono essere uguali, tra più file differenti. Altrimenti la concatenazione non viene eseguita correttamente.

La funzione controllo è stata scritta per verificare se la tabella che l'utente sta concatenando è la prima e quindi verrà messa in prima posizione oppure se c'è già una tabella presente, allora viene chiamata la funzione concatena.

La funzione concatena si occupa di eliminare la seconda colonna per evitare la ripetizione della tipologia delle sostanze e poi successivamente viene fatto il merge delle tabelle in base all'ID.

Questa nuova tabella viene sovrascritta al primo posto della funzione controllo in modo tale che possa essere nuovamente concatenata con un ulteriore file, tenendo i progressi salvati.

### Elimina colonna

```
if colonna!= '':
    for x in range (Num+1):
        if x==indice:
            dataframe= pd.DataFrame(Dati[indice])

            column_headers=list(dataframe.columns.values)
            for y in range (len(column_headers)):
                if str(column_headers[y])==str(colonna):
                    val=y
                    display(val)
            temp=dataframe
            temp=temp.drop(temp.columns[val], axis=1)
            newlist[c]=temp
            stampaTabelle(newlist,c,"Dati aggiornati", True)
            c+=1
```

Questo è un algoritmo che stato importante per la realizzazione del codice perché permette all'utente di cancellare una colonna presente all'interno del file excel, in modo tale da poter fare uno studio più preciso dei dati che ha a disposizione.

La funzione removecolumn è presente nella pagina "data\_removal.py". L'utente seleziona tramite due combobox quale colonna vuole eliminare e da quale file excel vuole farlo. L'algoritmo si occupa di selezionare il file giusto, verificare la presenza della colonna e successivamente la cancella tramite la funzione "drop" fornita da pandas. Successivamente il file viene aggiornato e di conseguenza anche la lista delle colonne eliminabili, in modo tale che se l'utente voglia eliminare un ulteriore colonna non ci sia un conflitto all'interno del codice.

## Esporta tabelle

```
def esportaTabelle():
    """Funzione esporta tabelle"""
    file= pd.DataFrame(newarray)
    filescores=pd.DataFrame(scores)
    fileloadings=pd.DataFrame(loadings)
    """ Trovo percorso al desktop del computer di chi usa il codice """
    path=os.path.expanduser("~/Desktop") + "/export_complete_table.xlsx"
    path2=os.path.expanduser("~/Desktop") + "/export_scores.xlsx"
    path3=os.path.expanduser("~/Desktop") + "/export_loadings.xlsx"

    """converto il file in un excel e lo salvo"""
    file.to_excel(path)
    filescores.to_excel(path2)
    fileloadings.to_excel(path3)

    lbl = Label(frameDatabase, text="File succesfully saved ", font=('arial', 25), bd=18, bg="white")
    lbl.grid(row=12, padx=100)
```

Infine questo è l'ultimo algoritmo degno di nota, che è stato espressamente richiesto dal product owner.

La funzione esportaTabelle presente nel file "scelta\_Exploration.py" è stata scritta in modo tale che l'utente possa esportare alcune tabelle costruite ed elaborate dall'applicazione.

Tramite l'utilizzo della libreria "os.path.expanduser" siamo in grado di ricostruire il percorso che porta al desktop di qualunque utente che stia utilizzando l'applicazione.

Avendo ricostruito il percorso, siamo in grado di salvare, grazie alla funzione "to\_excel", le tabelle prodotte dalla nostra applicazione sul desktop dell'utente.

Per assicurare l'avvenuto salvataggio è stata inserita una label di conferma.



## 13. Fonti

- Hans Van Vliet. Software Engineering: Principles and Practice. Third edition.
- Martina Seidl, MarionScholz, Chirstian Huemer, Gerti Kappel. UML @ Classroom: An introduction to Object-Oriented Modeling.
- Carlotta Ferrari , Alessandro Ulrici e Francesco Saverio Romolo. Article: Expert System for Bomb Factory Detection by Networks of Advance Sensors
- <https://stackoverflow.com/>
- <https://www.freecodecamp.org>
- <https://pandastable.readthedocs.io/>
- <https://pandas.pydata.org/>
- <https://www.geeksforgeeks.org/>
- <https://www.w3resource.com/>
- <https://www.tutorialspoint.com/>
- <https://medium.com/>
- <https://copyprogramming.com/>
- <https://www.codeproject.com/>
- <https://realpython.com/>
- <https://www.w3schools.com/>
- <https://www.nexsys.it/>





## 14. Ringraziamenti

Al termine di questo elaborato, vorrei ringraziare tutti coloro che mi hanno sostenuto durante il percorso universitario e di scrittura della tesi.

Un sentito ringraziamento va al mio relatore, il Professore Angelo Michele Gargantini, che mi ha seguito con disponibilità e gentilezza, fornendomi spunti fondamentali nella stesura di questo lavoro e per avermi dato consigli nei momenti di indecisione.

Successivamente ringrazio il mio correlatore, il Dottor Giorgio Felizzato, che con le sue conoscenze e le sue capacità, si è reso disponibile per risolvere ogni mio dubbio in merito alla costruzione dell'applicazione.

Ringrazio mia mamma, che ha sempre creduto in me, soprattutto nei momenti difficili e che mi ha dato una grossa mano a raggiungere questo traguardo.

Ringrazio Mattia, mio compagno di università, che mi ha accompagnato in questo percorso universitario e a cui auguro il meglio per il futuro.

Per ultimi, ma non per meno importanti, ringrazio i miei amici, Alessandro e Daniele, che ci sono da sempre, ed inoltre ringrazio anche coloro che ci sono stati durante tutto il percorso.