UNIVERSITA' DEGLI STUDI DI BOLOGNA

Corso di Laurea in Ingegneria e Scienze Informatiche

Supporto allo sviluppo di una piattaforma semantica clinica: generazione di dati sintetici in Cancer Virtual Lab

Tesi di Laurea di:

Claudia Giannelli Taccarino

Relatore:

Prof.ssa Antonella Carbonaro

Anno accademico 2025-2026

INDICE

INTRODUZIONE

Nel dominio oncologico, i dati clinici non costituiscono un patrimonio unitario e coerente, ma risultano distribuiti tra sistemi informativi eterogenei, progettati per finalità differenti e fondati su modelli, standard e linguaggi non sempre compatibili. Ne consegue che lo stesso paziente viene spesso descritto attraverso rappresentazioni parziali e frammentate, difficilmente integrabili e, in molti casi, non interoperabili. Questa condizione limita la possibilità di ottenere una visione complessiva del quadro clinico e riduce l'efficacia delle attività di analisi e ricerca basate sui dati.

In tale contesto si colloca il progetto Cancer Virtual Lab (CVL), una piattaforma semantica clinica sviluppata con l'obiettivo di rappresentare i dati oncologici provenienti dalla pratica clinica in una forma standardizzata, computabile e interrogabile, capace di supportare attività avanzate di ricerca. L'approccio adottato mira a superare la frammentazione informativa mediante l'integrazione semantica, rendendo i dati più accessibili, confrontabili e riutilizzabili all'interno di un quadro informativo condiviso.

Tuttavia, l'accesso e l'impiego dei dati sanitari sono inevitabilmente condizionati da vincoli normativi, requisiti di sicurezza e obblighi stringenti in materia di tutela della riservatezza. Anche quando i dati risultano formalmente disponibili, essi possono presentare criticità rilevanti: incompletezza, sbilanciamento rispetto a specifiche condizioni cliniche, oppure insufficienza quantitativa e qualitativa. In questo scenario, la generazione di dati sintetici rappresenta una strategia chiave per ampliare la disponibilità di dataset utilizzabili in modo controllato, riducendo il rischio di re-identificazione e rendendo possibile la sperimentazione anche in contesti in cui l'impiego dei dati reali risulta limitato.

La presente tesi si inserisce in tale cornice e affronta la progettazione e l'implementazione di un processo di generazione di dati sintetici integrato all'interno di una piattaforma semantica clinica orientata al supporto della ricerca oncologica. Il lavoro svolto ha avuto come obiettivo la realizzazione di una sezione pienamente operativa, in grado di generare dataset sintetici a partire dai dati disponibili nel Cancer Virtual Lab, validarli rispetto ai vincoli del modello e salvarli nei repository dedicati, rendendoli immediatamente riutilizzabili nel flusso applicativo complessivo della piattaforma.

CAPITOLO 1

GENERAZIONE DATI SINTETICI

I dati sintetici sono informazioni artificialmente generate che riproducono le caratteristiche dei dati reali. A differenza dei dati raccolti direttamente da osservazioni, esperimenti o sensori, i dati sintetici sono prodotti attraverso algoritmi, modelli matematici o tecniche di machine learning. Lo scopo principale è riprodurre la struttura statistica e le relazioni presenti in un dataset reale, pur rimanendo scorrelati dai dati reali.

Secondo le analisi di Gartner, nei primi anni del decennio si prevedeva che entro il 2024 circa il 60% dei dati utilizzati nei processi di business analytics e nei sistemi di AI sarebbe stato generato artificialmente. La previsione di Gartner si è rivelata corretta nel cogliere il cambio di paradigma, evidenziando una transizione strutturale verso modelli *data-driven* basati su generazione controllata, più che su raccolta massiva di dati reali. Questa evoluzione segna il passaggio da una logica di *data collection* a una logica di *data engineering*, in cui la qualità del modello generativo diventa più rilevante della quantità del dato grezzo.

1.1 - Tipi di dati sintetici

1.1.1 - Dati sintetici parziali

I dati sintetici parziali derivano da dataset reali nei quali solo una parte delle informazioni viene sostituita con valori artificiali, mantenendo il resto dei dati originali. Questa tecnica è utilizzata principalmente per proteggere le informazioni sensibili, sostituendo attributi ad alto rischio con valori sintetici, preservando al contempo la struttura complessiva del dataset. Nei dati soggetti a vincoli di privacy, la sintesi parziale consente di mascherare selettivamente le variabili critiche, mantenendo le proprietà statistiche e relazionali dei dati non sensibili. Essa risulta inoltre utile per colmare lacune informative e gestire dataset incompleti.

1.1.2 - Dati sintetici completi

I dati sintetici completi consistono in dataset interamente generati artificialmente, che non contengono alcun dato reale. La generazione completamente sintetica si fonda sulla modellazione degli attributi,

delle dipendenze e delle relazioni strutturali del dominio, al fine di emulare in modo coerente il comportamento informativo dei dati reali senza includere informazioni riconducibili a soggetti o eventi reali. Questa tipologia è particolarmente utilizzata nel test e nello sviluppo di modelli di machine learning, soprattutto in contesti caratterizzati da scarsità di dati reali o difficoltà di accesso ai dati.

1.1.3 - Dati sintetici ibridi

I dati sintetici ibridi combinano dataset reali e dataset completamente sintetici, accoppiando record reali con controparti sintetiche generate artificialmente. La sintesi ibrida permette di preservare la complessità del dominio reale, riducendo al contempo il rischio di esposizione di informazioni sensibili.

1.2 - Pipeline di generazione dei dati sintetici

1.2.1 - Acquisizione e raccolta dati

Raccolta dei dati di partenza, provenienti da fonti eterogenee, che costituiscono il riferimento semantico e statistico per il modello generativo.

1.2.2 - Pulizia e pre-processing dei dati

Eliminazione di valori anomali, incongruenze, duplicazioni e dati incompleti, al fine di garantire la qualità informativa del dataset di input. La qualità dei dati sintetici generati è direttamente proporzionale alla qualità dei dati reali utilizzati come base di modellazione.

1.2.3 - Analisi e modellazione dei dati

Analisi statistica e strutturale del dominio informativo, finalizzata all'identificazione delle distribuzioni, delle correlazioni, delle dipendenze tra variabili e delle relazioni semantiche. In questa fase si costruisce il modello generativo del dominio.

1.2.4 - Generazione dei dati sintetici

Applicazione dei modelli generativi per la produzione dei dataset sintetici.

1.2.5 - Convalida e valutazione

La convalida e la valutazione dei dati sintetici rappresentano una fase essenziale del processo generativo, finalizzata a garantirne qualità. In questa fase si verifica: la fedeltà statistica dei dati generati, la coerenza semantica delle relazioni, l'utilità applicativa nei contesti di impiego previsti, il livello di sicurezza rispetto al rischio di re-identificazione e l'assenza di bias artificiali.

1.3 - Metodi di generazione dei dati sintetici

1.3.1 - Modelli statistici basati su distribuzioni di probabilità

In questo approccio, i dati reali vengono prima analizzati per identificare le distribuzioni sottostanti, come le distribuzioni normali, esponenziali o chi-quadrate. A partire da tali distribuzioni, vengono generati campioni sintetici che riproducono le proprietà del dataset originale, consentendo la creazione di dati artificiali statisticamente coerenti.

1.3.2 - Modelli generativi basati su machine learning

I modelli generativi ML-based apprendono la struttura, i pattern e le relazioni dei dati reali attraverso processi di addestramento supervisionato o non supervisionato, per poi generare nuovi dati artificiali coerenti con il dominio appreso. Questi modelli consentono una rappresentazione avanzata delle dipendenze complesse e delle strutture latenti del dataset, ma richiedono quantità significative di dati reali per l'addestramento iniziale.

- Variational Autoencoder: i codificatori automatici variazionali sono modelli generativi probabilistici che producono nuovi dati a partire da una rappresentazione appresa dai dati originali. Il modello, di tipo non supervisionato, apprende la distribuzione probabilistica del dataset e utilizza un'architettura encoder-decoder per la generazione dei campioni sintetici. L'encoder mappa i dati di input in uno spazio latente a dimensionalità ridotta, modellato come

distribuzione (tipicamente gaussiana), mentre il decoder campiona da tale spazio e ricostruisce nuovi dati coerenti con la distribuzione appresa.

- Generative Adversarial Networks: GAN è un modello generativo supervisionato che può essere utilizzato per generare rappresentazioni realistiche e altamente dettagliate. Funziona addestrando due reti neurali, una che genera dati artificiali (un generatore) e l'altra che mira a distinguere dati artificiali da quelli reali (un discriminatore). Entrambe le reti vengono addestrate iterativamente, con il feedback del discriminatore che migliora l'output del generatore fino a quando il discriminatore non è più in grado di distinguere i dati sintetici da quelli reali.

1.3.3 - Generazione basata su regole

La generazione *rule-based* si fonda su regole, vincoli e logiche predefinite, che guidano la costruzione del dataset sintetico. Questo approccio consente un controllo fine delle caratteristiche del dato (valori minimi, massimi, medie, distribuzioni, vincoli semantici, relazioni strutturali), permettendo la creazione di dataset altamente personalizzati. A differenza dei dati completamente generati dall'intelligenza artificiale, che non hanno personalizzazione, i dati sintetici basati su regole offrono una soluzione su misura per soddisfare requisiti operativi distinti. Questo processo di generazione di dati sintetici si dimostra particolarmente utile nei test, nello sviluppo e nell'analisi, dove è essenziale una generazione di dati precisa e controllata.

1.3.4 - Generazione semantica SHACL-driven

Approccio *model-driven* basato su vincoli semantici e SHACL shapes, in cui i dati sintetici vengono generati come istanze conformi al modello ontologico del dominio. La sua applicazione è limitata ai contesti in cui il dominio informativo risulta formalizzato attraverso ontologie e modelli semantici espliciti.

1.4 - Vantaggi

1.4.1 - Generazione di dati illimitata

La generazione sintetica consente la produzione di volumi di dati virtualmente infiniti, superando i limiti strutturali dei dataset reali. La quantità e l’incompletezza dei dati non sono più un vincolo e

anche eventi rari possono essere riprodotti. Una volta progettato un generatore di dati sintetici, è possibile creare dataset di qualsiasi dimensione, adattandoli a specifiche esigenze di calcolo o analisi.

1.4.2 - Conformità normativa e tutela della privacy

Le leggi e i regolamenti sulla privacy dei dati possono influire sulla raccolta e l'archiviazione dei dati. I set di dati contenenti dati privati presentano un rischio perché il loro utilizzo potrebbe comportare una violazione della conformità. Se generati correttamente, i dati sintetici non sono riconducibili a persone reali, né in modo diretto né indiretto. Di conseguenza, viene meno il rischio di re-identificazione e tali dati possono collocarsi al di fuori dell'ambito di applicazione della normativa sulla protezione dei dati personali. In questa prospettiva, i dati sintetici rappresentano una soluzione coerente con i principi di *privacy by design*, in quanto integrano la tutela dei dati fin dalla fase di progettazione.

1.4.3 - Miglioramento dell'apprendimento nei modelli di machine learning

I dati sintetici favoriscono tecniche di data augmentation, aumentando la capacità di generalizzazione dei modelli e riducendo il rischio di overfitting.

1.4.4 - Riproducibilità

I dati sintetici consentono la replicabilità degli esperimenti attraverso la possibilità di ricostruire identiche condizioni di input e di contesto informativo, permettendo la riproduzione controllata dei processi sperimentali.

1.4.5 - Controllo dei bias

Se progettati correttamente, i dati sintetici possono evitare i bias intrinseci che spesso affliggono i dati reali. Questo consente di testare modelli in condizioni più neutrali e controllate.

1.4.6 - Riduzione dei costi

Riduzione significativa dei costi di raccolta, acquisizione, gestione, protezione e messa in sicurezza dei dati, in particolare in settori ad alta complessità tecnologica e normativa.

1.4.7 - Velocità di sviluppo

I dati possono essere generati *on demand*, senza dipendere da tempi di acquisizione, autorizzazioni o disponibilità del dato reale, accelerando i cicli di progettazione, test e sperimentazione.

1.4.8 - Condivisione dei dati e collaborazione

I dati sintetici facilitano la cooperazione tra enti e gruppi di ricerca, superando limiti legali, vincoli proprietari e barriere normative allo scambio dei dati.

1.4.9 - Qualità e controllo del dato

Poiché i dati sono generati artificialmente, è possibile modellarne con precisione le caratteristiche strutturali e statistiche. I dataset sintetici possono così essere progettati e personalizzati in funzione delle esigenze specifiche dei contesti applicativi, garantendo un controllo diretto e sistematico sul dominio informativo. In questo quadro, i dati sintetici consentono anche la costruzione controllata di scenari complessi e casi limite, inclusi eventi estremi e condizioni rare difficilmente osservabili nel mondo reale. La possibilità di rappresentare tali situazioni riduce le distorsioni dovute alla loro sottorappresentazione nei dataset reali e migliora la capacità dei modelli di machine learning di apprendere e gestire correttamente condizioni eccezionali.

1.5 - Applicazioni

1.5.1 - Sanità

I dati sintetici rappresentano una risorsa strategica per la sanità digitale, consentendo la creazione di coorti cliniche artificiali destinate alla ricerca, alla sperimentazione e allo sviluppo tecnologico senza il ricorso a dati sanitari reali. In ambito sanitario, la disponibilità del dato è spesso limitata da vincoli

autorizzativi, normative sulla privacy, incompletezza della documentazione clinica e rarità di specifiche condizioni patologiche. I dati sintetici permettono di superare tali limiti, rendendo possibile l'utilizzo di informazioni ad alta sensibilità in contesti di ricerca e sviluppo, garantendo al contempo il rispetto dei principi di protezione dei dati personali.

1.5.2 - Intelligenza artificiale e machine learning

I dati sintetici consentono di aumentare in modo controllato la quantità e la varietà dei dati disponibili per l'addestramento dei modelli di intelligenza artificiale, migliorandone robustezza e capacità di generalizzazione. Essi permettono di costruire dataset bilanciati, di rappresentare condizioni rare e di testare modelli in scenari complessi.

1.5.3 - Cybersecurity e sicurezza informatica

Nel dominio della sicurezza informatica, i dati sintetici consentono la simulazione di attacchi, la modellazione delle minacce e l'addestramento dei sistemi di difesa. Attraverso dataset artificiali è possibile testare sistemi di *intrusion detection*, valutare la resilienza delle infrastrutture digitali e simulare scenari di *data breach*, favorendo una gestione preventiva della sicurezza.

1.5.4 - Ricerca scientifica e metodologia della ricerca

In ambito scientifico, i dati sintetici favoriscono la riproducibilità e la replicabilità degli studi, consentendo la condivisione dei dataset e la verifica indipendente dei risultati. Essi permettono la costruzione di esperimenti controllati, la simulazione di scenari teorici e la validazione dei modelli senza dipendere da dataset proprietari o difficilmente accessibili.

1.5.5 - Finanza, assicurazioni e gestione del rischio

Nel settore finanziario e assicurativo, i dati sintetici consentono la simulazione di mercati, la costruzione di modelli di rischio e l'analisi predittiva di scenari critici. Essi permettono di effettuare stress test, simulazioni di crisi sistemiche, addestramento di modelli antifrode e valutazioni prospettiche senza l'utilizzo di dati sensibili reali.

1.5.6 - Digital twin

Nel contesto dei *digital twin* i dati sintetici consentono di alimentare i modelli digitali con informazioni artificiali coerenti con la struttura del sistema reale, rendendo possibili simulazioni, analisi predittive e sperimentazioni operative senza l'impiego di dati reali. Essi permettono di testare scenari e dinamiche evolutive in modo controllato, riproducibile e privo di impatti sul sistema reale.

1.6 - Limiti e criticità

1.6.1 - Controllo della qualità

La qualità di un dataset sintetico dipende direttamente dalla capacità del modello generativo di rappresentare correttamente il dominio informativo di riferimento. Modelli mal progettati o addestrati su dati incompleti, distorti o non rappresentativi possono produrre dataset artificiali affetti da errori strutturali e rappresentazioni distorte della realtà. I processi di validazione e controllo qualitativo risultano quindi essenziali, ma possono diventare complessi e onerosi in termini di risorse computazionali e temporali, soprattutto in presenza di grandi volumi di dati sintetici.

1.6.2 - Compromesso tra realismo e privacy

Una delle principali criticità riguarda il bilanciamento tra realismo dei dati e tutela della privacy. I dati sintetici devono essere sufficientemente realistici da preservare le strutture informative rilevanti per l'analisi e l'addestramento dei modelli, ma al contempo devono garantire l'assenza di riconducibilità a soggetti reali. Un eccesso di realismo può aumentare il rischio di re-identificazione, mentre un'eccessiva artificialità compromette l'utilità scientifica e applicativa del dataset.

1.6.3 - Bias sintetici

Sebbene i dati sintetici possano essere utilizzati per ridurre i bias presenti nei dataset reali, esiste il rischio di introdurre bias artificiali derivanti da presupposti errati nella progettazione del modello generativo. Inoltre, quando i modelli sintetici riproducono fedelmente i dati reali, possono replicare gli stessi bias strutturali già presenti nei dataset originari, trasferendo le distorsioni nei dati artificiali.

CAPITOLO 2

GENERAZIONE DI DATI SINTETICI

NEI KNOWLEDGE GRAPH RDF

2.1 - Approcci alla generazione di dati sintetici nei knowledge graph RDF

2.1.1 - Definizione formale di vincoli per la generazione model-driven: SHACL

SHACL (Shapes Constraint Language) è uno standard del World Wide Web Consortium (W3C) per la descrizione, la validazione e la modellazione di grafi RDF [1]. Basato su RDF e SPARQL, SHACL si colloca all'interno dell'ecosistema degli standard semantici insieme a RDF Schema e OWL, configurandosi come linguaggio formale per la definizione di vincoli su struttura e contenuto dei grafi RDF. Le specifiche SHACL sono definite mediante *shapes*, che rappresentano insiemi di vincoli applicabili a nodi e proprietà del grafo. Le shapes consentono di specificare condizioni relative al numero di occorrenze di una proprietà (cardinalità), ai tipi di dato, ai range numerici, ai pattern testuali e alle relazioni tra entità, operando come meccanismo di validazione semantica e strutturale dei dati rispetto al modello ontologico di riferimento. Nell'ambito della generazione di dati sintetici, SHACL non si limita a svolgere una funzione di validazione a posteriori, ma assume il ruolo di linguaggio formale per la definizione del dominio informativo, rendendo esplicita la struttura semantica che i dati devono rispettare.

2.1.2 - Generazione di grafi RDF a partire da modelli semantici: RDF Graph Generator

RDFGraphGen (RDF Graph Generator) è un framework per la generazione automatica di *knowledge graph* RDF sintetici, progettato secondo un paradigma *model-driven*, in cui i dati vengono costruiti ex novo a partire da una specifica semantica formale del dominio [2]. Il processo generativo non si basa su dataset preesistenti, ma sulla definizione di un modello descrittivo che governa la struttura del grafo. Elemento centrale di questo approccio è l'impiego delle SHACL shapes come linguaggio di specifica del dominio informativo. In questo contesto, SHACL non si configura unicamente come strumento di validazione dei dati, ma come modello formale del dominio.

RDFGraphGen è un framework *domain-agnostic* e scalabile, in grado di generare grafi RDF di dimensione variabile in qualunque dominio applicativo. La qualità del dataset prodotto non dipende da meccanismi casuali, ma dalla qualità e coerenza del modello semantico che lo descrive.

A differenza dei modelli generativi tradizionali RDFGraphGen non integra meccanismi di generazione basati su distribuzioni statistiche: i grafi sono prodotti esclusivamente sulla base dei vincoli strutturali e semantici definiti nelle SHACL shapes. Di conseguenza, quando l'obiettivo è la riproduzione delle distribuzioni dei dati reali, è necessario ricorrere a modelli generativi di tipo statistico, rule-based o basati su machine learning.

Il funzionamento di RDFGraphGen si basa sulla traduzione dei vincoli espressi nelle SHACL shapes in regole di generazione automatica. Le *NodeShape* descrivono i tipi di entità da generare, mentre le *PropertyShape* descrivono le proprietà di queste entità, cioè quali informazioni devono avere, come devono essere collegate tra loro e con quali regole.

Il processo di generazione avviene in modo ordinato:

1. Il sistema individua le *NodeShape* e crea le entità principali del grafo (i nodi di base)
2. Su ogni entità applica le *PropertyShape*, che definiscono l'insieme delle proprietà associate a ciascun nodo e, per ogni proprietà, stabiliscono:
    - il numero di occorrenze da generare per ciascun nodo (vincoli di cardinalità)
    - il tipo del valore (tipo di dato semplice oppure riferimento a un'altra entità)
    - i vincoli sui valori
3. Specifiche regole SHACL permettono di introdurre una variabilità controllata nella generazione dei dati, consentendo al sistema di scegliere valori da insiemi predefiniti, garantendo diversità informativa senza compromettere la coerenza semantica del modello.
4. La quantità complessiva di dati generati è controllata da un parametro numerico (*scale-factor*) che permette di decidere quante entità principali creare. Da queste, il resto del grafo si sviluppa automaticamente in base alle relazioni definite nel modello.

Il risultato finale è un grafo RDF sintetico che non è una copia di dati reali, ma una realizzazione artificiale di un modello semantico. Le relazioni, le strutture e i vincoli riflettono il dominio concettuale definito dalle shapes, garantendo riproducibilità e controllabilità del processo.

2.1.3 - Generazione di dati sintetici ontology-driven e data-driven: Onto-CGAN

Un limite strutturale degli approcci puramente *model-driven,* come quelli basati su specifiche SHACL, è l'assenza di meccanismi intrinseci per la riproduzione delle distribuzioni statistiche

osservate nei dati reali. In tali modelli, la coerenza semantica e strutturale è garantita dal rispetto dei vincoli ontologici, ma il realismo statistico delle variabili dipende da fasi di *post-processing* o da integrazioni esterne.

Per superare questo limite, la letteratura recente propone approcci generativi che combinano conoscenza ontologica e modelli statistici *data-driven*. In questo contesto si colloca Onto-CGAN, un framework di *ontology-enhanced Conditional Generative Adversarial Networks* progettato per la generazione di dati sintetici clinici anche in assenza di dati reali relativi alla patologia di interesse [3].

Onto-CGAN integra la conoscenza fornita da ontologie biomediche all'interno dell'architettura di una Conditional GAN, utilizzando embedding ontologici come variabili di condizionamento del generatore. Le ontologie vengono trasformate in rappresentazioni vettoriali mediante tecniche di embedding basate su OWL2Vec, che consentono di preservare sia la struttura del grafo ontologico sia le informazioni semantiche associate.

Durante la fase di addestramento, Onto-CGAN utilizza dati clinici reali relativi a patologie osservate, mentre la generazione è condizionata sugli embedding ontologici di patologie non presenti nel dataset. In questo modo, il modello è in grado di produrre dati sintetici che rispettano sia le relazioni concettuali del dominio clinico sia le distribuzioni statistiche apprese dai dati reali.

Tuttavia, tali approcci presentano alcune criticità rilevanti: richiedono dataset di addestramento sufficientemente ampi, introducono una maggiore complessità computazionale e riducono il grado di controllabilità formale del processo generativo. Inoltre, il comportamento del modello è in parte opaco, rendendo meno immediata la tracciabilità delle decisioni generative rispetto agli approcci basati su specifiche semantiche esplicite.

2.2 - Valutazione della qualità dei grafi RDF sintetici

In letteratura, la valutazione dei dati sintetici viene comunemente ricondotta a tre dimensioni principali: *fidelity*, intesa come coerenza statistica e semantico-strutturale rispetto al dataset reale; *utility*, intesa come capacità del dataset sintetico di supportare gli stessi casi d'uso e interrogazioni; e *privacy*, intesa come controllo del rischio di divulgazione informativa.

Fidelity

2.2.1 - Valutazione strutturale dei grafi sintetici

I primi approcci alla valutazione dei grafi sintetici si collocano nell'ambito delle reti complesse, dove il confronto tra grafo reale e generato viene condotto attraverso proprietà topologiche globali. In tale contesto, la letteratura non privilegia il confronto nodo per nodo né la verifica di isomorfismo, ma adotta valutazioni basate sulla coerenza statistica delle caratteristiche strutturali complessive [4].

La fidelity strutturale viene pertanto stimata mediante il confronto tra distribuzioni di metriche topologiche consolidate, quali distribuzione dei gradi, coefficiente di clustering, lunghezza media dei cammini minimi, assortatività e struttura delle componenti connesse. Tali misure consentono di verificare in che misura il grafo sintetico preservi le proprietà globali del grafo osservato senza costituirne una replica [4].

2.2.2 - Qualità dei Knowledge Graph

Nel contesto dei Knowledge Graph, la valutazione della qualità è stata ampiamente formalizzata nella letteratura sui *Linked Data,* intesi come grafi RDF pubblicati sul Web secondo principi di interoperabilità e interlinking. Poiché i Knowledge Graph condividono la rappresentazione RDF e i medesimi meccanismi semantici (URI, triple, ontologie, SPARQL), tali contributi costituiscono un riferimento metodologico consolidato anche per l'analisi della qualità dei grafi RDF [5][6].

In tale quadro, la letteratura individua dimensioni ricorrenti quali accuratezza, completezza, consistenza, interoperabilità e accessibilità, associate a metriche volte a misurare copertura informativa, correttezza sintattica e semantica, coerenza rispetto a vincoli di schema, nonché disponibilità e riusabilità dei dati. Sebbene tali tassonomie siano state originariamente sviluppate per dataset reali, esse risultano rilevanti anche per i grafi sintetici, in particolare per la verifica della conformità strutturale e semantica rispetto al modello di riferimento [5][6].

2.2.3 - Validazione test-driven e constraint-based

Studi nell'ambito dei Knowledge Graph e dei dataset RDF hanno proposto approcci *test-driven* alla valutazione della qualità, nei quali i criteri di qualità vengono esplicitati sotto forma di *test case*

eseguibili in SPARQL, finalizzati a individuare in modo sistematico errori ricorrenti, inconsistenze e anomalie strutturali nei dataset RDF [7]. L'idea alla base è che la qualità non debba essere trattata come un concetto astratto o esclusivamente descrittivo, ma come un insieme di requisiti verificabili automaticamente e ripetibili nel tempo, analogamente a quanto avviene nella validazione del software.

In una prospettiva complementare, la validazione dei grafi RDF può essere affrontata mediante linguaggi di vincolo dedicati, che consentono di formalizzare requisiti strutturali e semantici legati al modello di riferimento. In particolare, SHACL permette di specificare vincoli relativi a cardinalità, datatype, classi ammesse e pattern di relazione, rendendo possibile la verifica automatica del grafo rispetto a requisiti espliciti e la produzione di report sulle violazioni [8].

Utility

2.2.4 - Valutazione basata sull'utilità delle query

L'utility dei grafi sintetici può essere valutata analizzando la stabilità dei risultati ottenuti da interrogazioni SPARQL eseguite, in condizioni equivalenti, sul grafo reale e su quello sintetico. In tale prospettiva, *benchmark* quali Berlin SPARQL Benchmark (BSBM) e WatDiv, pur concepiti originariamente per la valutazione prestazionale dei *triplestore*, forniscono workload controllati e riproducibili utili per confrontare dataset differenti rispetto alla selettività dei dati, alla struttura delle query e al comportamento dei risultati [9][10].

In questo approccio, l'utilità non viene misurata come mera similarità statistica globale, bensì come capacità del dataset sintetico di supportare interrogazioni realistiche senza introdurre variazioni significative nei risultati attesi. Il confronto può essere condotto verificando la stabilità delle cardinalità, delle aggregazioni e dei ranking prodotti dalle medesime query applicate ai due grafi.

In ambito applicativo, l'utility può inoltre essere interpretata in senso più ampio, includendo la capacità del dataset sintetico di sostenere procedure analitiche e processi decisionali analoghi a quelli condotti sui dati reali. In tale prospettiva, la valutazione consiste nel verificare se le stesse pipeline di analisi (ad esempio estrazioni, aggregazioni, stratificazioni o metriche di outcome) producano risultati compatibili tra grafo reale e grafo sintetico.

2.2.5 - Privacy e limiti della similarità

La letteratura evidenzia infine come una similarità eccessiva tra grafo reale e grafo sintetico possa comportare rischi di *leakage strutturale.* Studi sull'anonimizzazione dei grafi mostrano che la replica fedele di nodi ad alta centralità o di sottografi rari può consentire forme di re-identificazione anche in assenza di attributi espliciti [11].

La valutazione dei dati sintetici non può limitarsi alla sola fidelity e utility, ma deve includere metriche di *privacy disclosure,* distinguendo tra *membership disclosure*, ossia la possibilità di inferire se un soggetto sia presente nel dataset reale, e *attribute disclosure,* ossia la possibilità di ricostruire o dedurre attributi sensibili relativi a un soggetto. Operativamente, tali rischi vengono valutati mediante procedure di *attack-based evaluation*, nelle quali si simulano attaccanti che tentano di inferire la presenza di un soggetto nel dataset reale o di ricostruirne attributi sensibili, nonché mediante analisi basate sulla vicinanza tra record reali e sintetici, finalizzate a individuare eventuali casi di eccessiva similarità e potenziale *leakage informativo.* [12].

Ne consegue che una differenza strutturale controllata tra grafo reale e sintetico non costituisce necessariamente una perdita di qualità, ma può rappresentare un requisito di sicurezza coerente con i principi di *privacy-by-design.*

CAPITOLO 3

CANCER VIRTUAL LAB

Il progetto Cancer Virtual Lab mira a sviluppare uno strumento di supporto alla ricerca in ambito oncologico. È una piattaforma già prototipata su oltre 30.000 pazienti oncologici presso l'IRCCS IRST "Dino Amadori" (Italia).

La sua fiinalità è trasformare cartelle cliniche in una rappresentazione semantica, standardizzata e computabile della realtà clinica, a supporto di interrogazioni avanzate e analisi computazionali, in coerenza con le priorità europee in materia di EHDS (European Health Data Space), AI Act e Digital Health and Care Strategy.

Per conseguire questo obiettivo, CVL è costruito su un'ontologia biomedica che costituisce il livello semantico fondativo dell'intera piattaforma.

Un'ontologia biomedica è un modello concettuale formalizzato che descrive in modo univoco le entità del dominio clinico e le relazioni logiche che intercorrono tra esse. Essa non si limita a elencare termini, ma ne definisce il significato, la gerarchia, le dipendenze e i vincoli, consentendo ai sistemi informatici di interpretare i dati clinici in modo coerente. In questo modo, l'informazione sanitaria viene trasformata da semplice dato strutturato in conoscenza semanticamente interpretabile e interrogabile.

Su questa base semantica, CVL offre agli utenti clinici e ai ricercatori un ambiente interattivo per l'esplorazione dei dati oncologici, che consente di:

- definire e filtrare coorti di pazienti mediante criteri clinici
- analizzare pattern, esiti terapeutici ed evoluzione dei percorsi di cura
- generare dataset sintetici a fini di ricerca
- interrogare i dati tramite intelligenza artificiale in linguaggio naturale, con risposte generate direttamente a partire dal knowledge graph clinico

3.1 - Obiettivi

Il Cancer Virtual Lab è progettato per fornire al ricercatore e al clinico una visione globale, strutturata e affidabile dei dati dei pazienti oncologici, superando la frammentazione dei sistemi informativi tradizionali.
Gli obiettivi primari della piattaforma sono:

- garantire un alto grado di disponibilità e interrogabilità dei dati clinici
- mettere a disposizione strumenti di analisi statistica e di intelligenza artificiale
- offrire strumenti avanzati a supporto della ricerca scientifica

In parallelo, CVL mira a costruire una base dati semantica a grafo in grado di supportare inferenza automatica e ragionamento clinico, permettendo di derivare nuova conoscenza a partire dai dati osservati, in modo verificabile e tracciabile.

3.2 - Impatto clinico

Questa architettura rende possibile:

- il supporto decisionale in tempo reale: CVL non interroga semplicemente tabelle o referti, ma un modello semantico della realtà clinica. Le inferenze ontologiche permettono di correlare entità secondo regole cliniche formalizzate. Il sistema può quindi rispondere a domande complesse non come un motore di reporting, ma come un motore inferenziale di ragionamento clinico.
- l'uso multicentrico dei dati: supera il modello dei *data warehouse* centralizzati adottando un paradigma federato, in cui le query possono essere distribuite tra più istituzioni sanitarie, preservando la localizzazione dei dati presso i rispettivi titolari e riducendo la necessità di trasferimento o duplicazione. Questo consente analisi su coorti, studi osservazionali distribuiti e confronti di outcome tra strutture, senza violare i vincoli di privacy, sovranità del dato e normativa europea (GDPR, EHDS).
- l'evoluzione verso un AI Digital Twin: un Digital Twin clinico è una controparte digitale dinamica del paziente reale. Non si tratta di una semplice copia dei dati, ma di un modello continuamente aggiornato che riflette lo stato clinico del paziente. Avendo una rappresentazione formale, verificabile e completa del profilo clinico del paziente, CVL rende

possibile simulare scenari terapeutici alternativi, valutare in silico protocolli clinici e stimare l'impatto di decisioni diverse prima di applicarle al paziente reale. A differenza delle *black box* di Machine Learning, queste simulazioni sono trasparenti, tracciabili e spiegabili, perché ogni risultato è ancorato a regole cliniche, ontologie e dati osservabili.

3.3 - Dataset

Il Cancer Virtual Lab si distingue nel panorama delle piattaforme cliniche per la dimensione e la profondità del patrimonio informativo su cui è costruito. CVL integra dati provenienti dalla pratica clinica reale di un grande istituto oncologico, comprendendo oltre 27.000 pazienti, più di 700.000 somministrazioni terapeutiche, oltre 1.000.000 di osservazioni di laboratorio, circa 100.000 valutazioni ECOG, più di 55.000 eventi avversi, oltre 30.000 classificazioni TNM e più di 22.000 percorsi terapeutici [14].

Una base dati di questa scala consente alla piattaforma di operare in un regime di *real-world evidence*, ancorando inferenze, correlazioni e supporto decisionale a popolazioni cliniche reali e a percorsi terapeutici effettivamente osservati, conferendo al sistema una solidità empirica difficilmente raggiungibile con dataset di dimensioni ridotte.

Al contempo, il dataset presenta vincoli strutturali rilevanti. I pazienti inclusi sono esclusivamente soggetti deceduti, in conformità ai criteri di utilizzo e tutela dei dati sanitari, e alcune condizioni cliniche rare risultano rappresentate in modo numericamente limitato. In questo contesto, il dataset si presta all'esplorazione di strategie di generazione di dati sintetici, finalizzate ad arricchire il dataset con istanze plausibili ma artificiali, mantenendo coerenza clinica e rispetto dei vincoli semantici del modello.

Il dataset utilizzato da CVL non viene impiegato nella sua forma grezza. Tutti i dati clinici che alimentano CVL sono preventivamente normalizzati e validati attraverso la *FHIR Transformation Pipeline,* che trasforma i flussi eterogenei provenienti dai sistemi ospedalieri in risorse HL7-FHIR strutturate e semanticamente coerenti [14].

La persistenza dei dati è realizzata attraverso un knowledge graph RDF ospitato in GraphDB, che funge da triplestore semantico della piattaforma. Questo livello non si limita alla memorizzazione dei dati, ma ne consente anche l'interrogazione semantica tramite SPARQL, rendendo possibile esprimere relazioni cliniche complesse e collegare informazioni provenienti da fonti eterogenee in

modo coerente e computabile. Il knowledge graph costituisce pertanto la base informativa unica e condivisa dell'intera piattaforma.

3.4 - Standard HL7 FHIR

Lo standard HL7 FHIR (Fast Healthcare Interoperability Resources) rappresenta oggi il principale riferimento internazionale per la modellazione e lo scambio delle informazioni sanitarie in forma strutturata. È stato progettato per superare i limiti degli standard sanitari precedenti, che spesso risultavano complessi da implementare e poco flessibili rispetto all'evoluzione dei sistemi informativi. Il suo obiettivo è fornire un modello comune per rappresentare i concetti fondamentali della pratica clinica, consentendo a sistemi eterogenei di interpretare e utilizzare i dati in modo coerente. A tal fine, FHIR adotta un approccio modulare basato su risorse cliniche standardizzate, ciascuna delle quali descrive un'entità rilevante nel dominio sanitario.

Ogni risorsa FHIR è definita tramite uno schema formale che specifica gli attributi disponibili, i tipi di dato ammessi, le cardinalità e le relazioni con altre risorse. Questo consente di costruire rappresentazioni cliniche che non si limitano a elencare valori, ma esprimono anche il contesto e il significato delle informazioni. Le relazioni esplicite tra le risorse rendono possibile una ricostruzione strutturata del quadro clinico.

FHIR prevede inoltre un forte legame con i sistemi di codifica e le terminologie cliniche standard, come SNOMED CT, LOINC e ICD, che possono essere utilizzati per codificare in modo univoco diagnosi, procedure e osservazioni. Questo aspetto è fondamentale per garantire che i dati non siano solo formalmente corretti, ma anche semanticamente interpretabili e confrontabili tra sistemi diversi.

3.5 - FHIR Transformation Pipeline

I dati utilizzati dal Cancer Virtual Lab provengono dai sistemi informativi dell'IRST IRCCS "Dino Amadori" di Meldola e riflettono i flussi operativi della pratica clinica, diagnostica e amministrativa. Come avviene tipicamente nei contesti ospedalieri, tali informazioni sono archiviate in strutture tabellari eterogenee, progettate per esigenze gestionali e operative piuttosto che per l'integrazione semantica e l'analisi avanzata. In questa forma originaria, i dati risultano privi di una semantica esplicita, di relazioni cliniche formalizzate e di standard di interoperabilità.

La FHIR Transformation Pipeline del CVL nasce per colmare questa distanza tra dato operativo e conoscenza clinica computabile. Il suo scopo non è una mera conversione di formato, ma la trasformazione di informazione frammentata in un modello semanticamente coerente, standardizzato e interoperabile capace di alimentare il knowledge graph di CVL [14].

La pipeline del Cancer Virtual Lab segue un paradigma ETL semantico (*Extract-Transform-Load*) articolato in cinque fasi:
Input, Refinement, Mapping, Validation e Graph Deployment.

Questa organizzazione consente di separare nettamente:

- l'estrazione dei dati clinici
- la trasformazione dei dati
- la validazione ontologica
- il caricamento nel knowledge graph

3.5.1 - Input

L'estrazione dei dati avviene nell'ambiente istituzionale IRST tramite query SQL eseguite in Qlik Sense, che interrogano i sistemi di cartella clinica elettronica e i registri operativi ospedalieri. Questa fase include tre processi fondamentali:

- **Anonimizzazione**
  È applicata una tecnica di *statistical noise additction* per ciascun paziente, con l'obiettivo di ridurre il rischio di re-identificazione individuale.
- **Pulizia dei dati**
  Vengono eliminati record incompleti, valori inconsistenti, date mancanti e formati non validi. La presenza di dati mancanti o incompleti è comune nei contesti clinici reali, dove la raccolta delle informazioni è orientata prima di tutto alla cura del paziente e non alla completezza del dato a fini di analisi.
- **Standardizzazione**
  Variabili categoriche vengono uniformate per eliminare ambiguità sintattiche.

3.5.2 - Refinement

In questa fase i dati cessano di essere semplici righe tabellari e vengono preparati come oggetti clinici futuri. Il refinement opera direttamente sui CSV esportati e svolge funzioni cruciali per la coerenza FHIR:

- scomposizione di campi complessi (es. T2N1M1 → T=2, N=1, M=1)
- trasformazione di stringhe linguistiche ("Sì", "No") in tipi formali (booleani)
- verifica dei campi obbligatori FHIR (date, codici, identificativi)
- sostituzione degli identificativi locali dei pazienti con identificatori FHIR anonimi e persistenti

3.5.3 - Mapping

La fase di mapping realizza la trasformazione semantica vera e propria ed è articolata in due livelli.

- **CSV → FHIR JSON**

Ogni entità clinica viene mappata su una risorsa FHIR R4 standard (Patient, Condition, Observation, CarePlan, Medication, MedicationAdministration, AdverseEvent, Organization, ecc.) secondo regole dichiarative.

Le risorse vengono inoltre arricchite mediante binding a terminologie biomediche standardizzate, tra cui SNOMED CT (diagnosi cliniche), LOINC (test di laboratorio e biomarcatori), ICD-9/10 (codifica diagnostica), UMLS (integrazione e riconciliazione semantica tra terminologie), RxNorm/ATC (farmaci), MedDRA (eventi avversi), NCIt (stadi, linee di terapia e setting terapeutici) e HGNC (geni).

Il risultato di questa fase è una collezione di file *FHIR-compliant* JSON, generati e validati tramite le librerie *fhir.resources.*

- **FHIR JSON → FHIR RDF**

I file JSON vengono trasformati in RDF mediante la libreria *fhirtordf*, che utilizza l'ontologia OWL ufficiale di FHIR per convertire ogni risorsa in triple semanticamente valide.
La libreria *rdflib* costruisce i grafi RDF e li serializza in formato Turtle. Ogni risorsa mantiene un

identificatore persistente, garantendo la continuità dei riferimenti tra le diverse entità cliniche anche nel grafo RDF.

3.5.4 - Validation

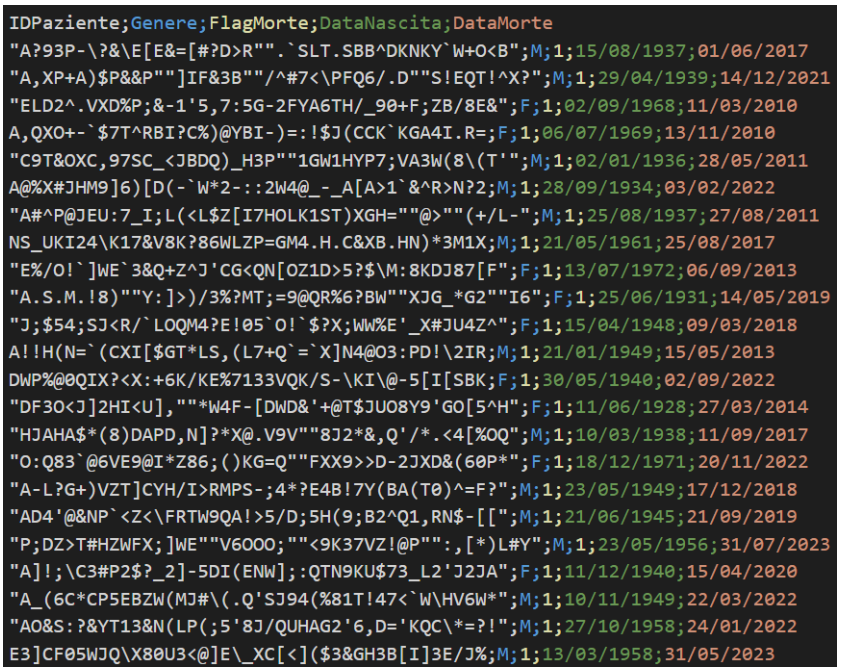
La validazione non è una fase separata ma è incorporata nella trasformazione. Ogni risorsa viene validata rispetto all'ontologia OWL di FHIR attraverso:

- controllo dell'esistenza del tipo FHIR
- verifica dei domini e dei range RDF
- validazione dei datatype XSD
- rispetto delle gerarchie FHIR

Ogni violazione genera un'eccezione che interrompe la pipeline, impedendo la produzione di RDF semanticamente invalido. La pipeline è quindi *semantic-by-construction.*

3.5.5 - Deployment

I file RDF in formato Turtle, prodotti dalla pipeline, vengono importati nel triplestore GraphDB insieme all'ontologia CVL, che definisce il modello concettuale del dominio oncologico. Una volta completato il caricamento, viene eseguita automaticamente una serie di query SPARQL INSERT che arricchiscono e riallineano le risorse HL7-FHIR al modello ontologico di CVL.



```
IDPaziente;Genere;FlagMorte;DataNascita;DataMorte
"A?93P-\?&\E[E&=[#?D>R"".`SLT.SBB^DKNKY`W+O<B";M;1;15/08/1937;01/06/2017
"A,XP+A)$P&&P""]IF&3B""/^#7<\PFQ6/.D""S!EQT!^X?";M;1;29/04/1939;14/12/2021
"ELD2^.VXD%P;&-1'5,7:5G-2FYA6TH/_90+F;ZB/8E&";F;1;02/09/1968;11/03/2010
A,QXO+-`$7T^RBI?C%)@YBI-)=:!$J(CCK`KGA4I.R=;F;1;06/07/1969;13/11/2010
"C9T&OXC,97SC_<JBDQ)_H3P""1GW1HYP7;VA3W(8\(T'";M;1;02/01/1936;28/05/2011
A@%X#JHM9]6)[D(-`W*2-::2W4@_-_A[A>1`&^R>N?2;M;1;28/09/1934;03/02/2022
"A#^P@JEU:7_I;L(<L$Z[I7HOLK1ST)XGH=""@>""(+/L-";M;1;25/08/1937;27/08/2011
NS_UKI24\K17&V8K?86WLZP=GM4.H.C&XB.HN)*3M1X;M;1;21/05/1961;25/08/2017
"E%/O!`]WE`3&Q+Z^J'CG<QN[OZ1D>5?$\M:8KDJ87[F";F;1;13/07/1972;06/09/2013
"A.S.M.!8)""Y:]>)/3%?MT;=9@QR%6?BW""XJG_*G2""I6";F;1;25/06/1931;14/05/2019
"J;$54;SJ<R/`LOQM4?E!05`O!`$?X;WW%E'_X#JU4Z^";F;1;15/04/1948;09/03/2018
A!!H(N=`(CXI[$GT*LS,(L7+Q`=`X]N4@O3:PD!\2IR;M;1;21/01/1949;15/05/2013
DWP%@0QIX?<X:+6K/KE%7133VQK/S-\KI\@-5[I[SBK;F;1;30/05/1940;02/09/2022
"DF3O<J]2HI<U],""*W4F-[DWD&'+@T$JUO8Y9'GO[5^H";F;1;11/06/1928;27/03/2014
"HJAHA$*(8)DAPD,N]?*X@.V9V""8J2*&,Q'/*.<4[%OQ";M;1;10/03/1938;11/09/2017
"O:Q83`@6VE9@I*Z86;()KG=Q""FXX9>>D-2JXD&(60P*";F;1;18/12/1971;20/11/2022
"A-L?G+)VZT]CYH/I>RMPS-;4*?E4B!7Y(BA(T0)^=F?";M;1;23/05/1949;17/12/2018
"AD4'@&NP`<Z<\FRTW9QA!>5/D;5H(9;B2^Q1,RN$-[[";M;1;21/06/1945;21/09/2019
"P;DZ>T#HZWFX;]WE""V6OOO;""<9K37VZ!@P"":,[*)L#Y";M;1;23/05/1956;31/07/2023
"A]!;\C3#P2$?_2]-5DI(ENW];:QTN9KU$73_L2'J2JA";F;1;11/12/1940;15/04/2020
"A_(6C*CP5EBZW(MJ#\(.Q'SJ94(%81T!47<`W\HV6W*";M;1;10/11/1949;22/03/2022
"AO&S:?&YT13&N(LP(;5'8J/QUHAG2'6,D='KQC\*=?!";M;1;27/10/1958;24/01/2022
E3]CF05WJQ\X80U3<@]E\_XC[<](3&GH3B[I]3E/J%;M;1;13/03/1958;31/05/2023
```

Figura 3.1 - File pazienti.csv contenente i dati dei pazienti estratti dalla piattaforma Qlik Sense

```
[
  {
    "resourceType": "Patient",
    "id": "1",
    "identifier": [
      {
        "value": "A?93P-\\?&\\E[E&=[#?D>R\".`SLT.SBB^DKNKY`W+O<B"
      }
    ],
    "gender": "male",
    "birthDate": "1937-08-15",
    "deceasedDateTime": "2017-06-01"
  },
  {
    "resourceType": "Patient",
    "id": "2",
    "identifier": [
      {
        "value": "A,XP+A)$P&&P\"]IF&3B\"/^#7<\\PFQ6/.D\"S!EQT!^X?"
      }
    ],
    "gender": "male",
    "birthDate": "1939-04-29",
    "deceasedDateTime": "2021-12-14"
  },
```

Figura 3.2 - File pazienti.json contenente una collezione di risorse HL7-FHIR di tipo Patient in formato json, generate a partire dal file tabellare pazienti.csv

```
@prefix fhir: <http://hl7.org/fhir/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://hl7.org/fhir/Patient/1> a fhir:Patient ;
    fhir:nodeRole fhir:treeRoot ;
    fhir:Patient.birthDate [
        fhir:value "1937-08-15"^^xsd:date
    ] ;
    fhir:Patient.deceasedDateTime [
        fhir:value "2017-06-01"^^xsd:date
    ] ;
    fhir:Patient.gender [
        fhir:value "male"
    ] ;
    fhir:Patient.identifier [
        fhir:index "0"^^xsd:integer ;
        fhir:Identifier.value [
            fhir:value "A?93P-\\?&\\E[E&=[#?D>R\".`SLT.SBB^DKNKY`W+O<B"
        ]
    ] ;
    fhir:Resource.id [
        fhir:value "1"
    ] .
```

Figura 3.3 - File pazienti.ttl contenente le risorse Patient in formato RDF, ottenute dalla trasformazione delle corrispondenti risorse HL7-FHIR presenti nel file pazienti.json

CAPITOLO 4

CVL KNOWLEDGE GRAPH

Il knowledge graph del Cancer Virtual Lab è costruito in modo incrementale lungo la pipeline di trasformazione dei dati. All'interno dello stesso grafo coesistono la rappresentazione FHIR-native dei dati clinici, le entità e relazioni del modello ontologico CVL derivate tramite query di allineamento, le asserzioni inferite dal motore di *reasoning,* nonché i riferimenti a ontologie e risorse biomediche esterne [13].

In questo grafo, un'informazione clinica non è un semplice valore in una colonna, ma una entità formalmente definita, collegata ad altre entità tramite relazioni esplicite. Il percorso oncologico del paziente emerge come una rete semantica navigabile, che rappresenta l'evoluzione della malattia e delle decisioni terapeutiche nel tempo.

Questo approccio supera i limiti strutturali del modello relazionale, che richiede schemi rigidi, migrazioni frequenti e join complessi per rappresentare relazioni cliniche intrinsecamente multilivello. Nel knowledge graph, invece, lo schema è aperto ed estensibile: nuove entità e nuove relazioni possono essere introdotte senza compromettere quelle esistenti, perché il modello è basato su una rete di concetti e relazioni, non su tabelle.

In questo modo, il knowledge graph di CVL non si limita a conservare informazioni, ma è in grado di inferire automaticamente nuove relazioni, pattern e significati clinicamente rilevanti a partire dai dati disponibili, secondo regole formali, verificabili e riproducibili. Le inferenze prodotte sono quindi trasparenti e auditabili, un requisito fondamentale per l'utilizzo in ambito clinico.

Parallelamente, il sistema monitora la qualità della conoscenza rappresentata attraverso metriche di copertura clinica, copertura terminologica e coerenza ontologica, trasformando il knowledge graph in un asset clinico controllato e non in una semplice collezione di dati eterogenei.

4.1 - Ontologia di CVL

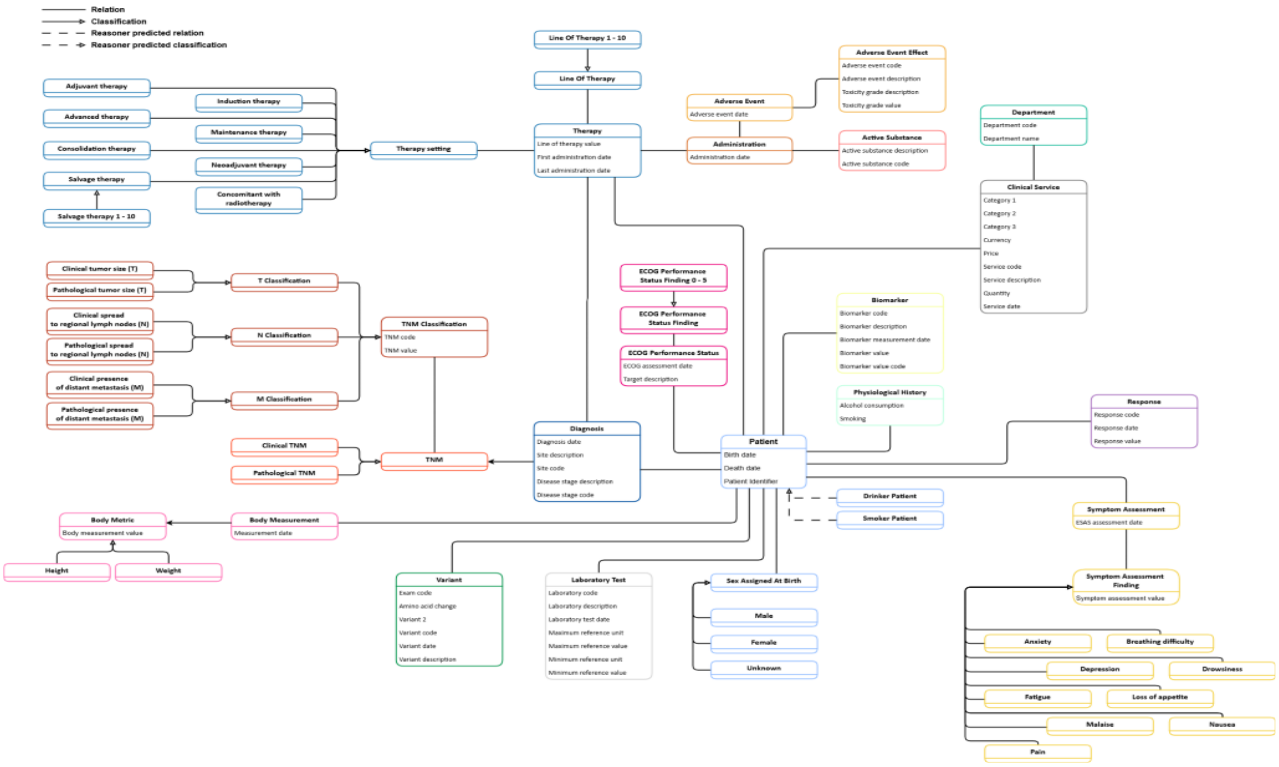


Figura 4.1 - Ontologia CVL [13]

L'architettura a grafo del Cancer Virtual Lab è resa possibile dall'uso di un'ontologia formale che definisce in modo esplicito il significato clinico delle entità e delle relazioni rappresentate. Grazie a questa base ontologica, il CVL Knowledge Graph non è soltanto un contenitore di dati, ma una rappresentazione esplicita della conoscenza clinica oncologica.

Nel Cancer Virtual Lab l'ontologia (*cvl_ontology*) adotta una struttura *patient-centric,* in cui la classe *cvl:Patient* costituisce il nodo semantico principale da cui si diramano tutti i flussi informativi clinici. Il paziente è caratterizzato da attributi demografici (data di nascita, sesso, eventuale data di decesso, identificativo pseudonimizzato) ed è collegato tramite relazioni ontologiche a tutte le dimensioni rilevanti del percorso oncologico, incluse *cvl:Diagnosis, cvl:Therapy, cvl:LaboratoryTest, cvl:Biomarker, cvl:Variant, cvl:ECOGStatus, cvl:SymptomAssessment, cvl:PhysiologicalHistory, cvl:BodyMeasurement, cvl:Response, cvl:Service e cvl:Department.*

La diagnosi (*cvl:Diagnosis*) rappresenta l'evento clinico che identifica la malattia oncologica ed è direttamente collegata al paziente tramite la relazione *is diagnosis of*. Ogni diagnosi è caratterizzata da un insieme strutturato di attributi, tra cui data di diagnosi, sede tumorale, stadio e classificazione

TNM. Il modello distingue esplicitamente la stadiazione clinica e patologica, rappresentate rispettivamente tramite le classi *CT, CN, CM* e *PT, PN, PM*, che confluiscono nelle entità di *TNM clinico* e *TNM patologico*, consentendo di rappresentare formalmente sia la valutazione iniziale sia quella post-chirurgica.

A partire dalla diagnosi si sviluppa il dominio terapeutico, centrato sulla classe *cvl:Therapy*, che rappresenta un piano di trattamento strutturato. Ogni terapia è associata a una linea di trattamento *(cvl:LineOfTherapy),* che consente di distinguere prima, seconda e successive linee, e a un therapy setting (neoadiuvante, adiuvante, avanzata, di mantenimento, di salvataggio, ecc.). La terapia è collegata alle somministrazioni (*cvl:Administration*), che ne descrivono la dinamica temporale, e indirettamente alle sostanze attive (*cvl:ActiveSubstance*) che ne costituiscono il contenuto farmacologico.

Gli eventi avversi (*cvl:AdverseEvent*) sono modellati come entità autonome, collegate sia alla terapia tramite la somministrazione (*cvl:Administration)* sia ai loro effetti clinici, permettendo di rappresentare in modo esplicito la tossicità associata ai trattamenti.

L'efficacia dei trattamenti è rappresentata dalla classe *cvl:Response*, che descrive l'esito clinico osservato nel paziente in relazione al suo percorso terapeutico. Collegata alla classe *cvl:Patient*, essa consente di modellare in modo strutturato l'evoluzione della risposta alla cura nel tempo e di supportare analisi longitudinali sull'andamento della malattia. In parallelo, lo stato clinico globale del paziente è descritto tramite lo stato di performance ECOG (*cvl:ECOGStatus*), le valutazioni dei sintomi (*cvl:SymptomAssessment*) e le misurazioni corporee (*cvl:BodyMeasurement*), che forniscono una rappresentazione formale della qualità di vita e della capacità funzionale nel tempo.

Il livello biologico e molecolare è modellato tramite *cvl:LaboratoryTest*, *cvl:Biomarker* e *cvl:Variant.* I test di laboratorio producono valori quantitativi, i biomarcatori rappresentano le entità biologiche osservate (geni, proteine, marcatori tumorali) e le varianti descrivono le alterazioni molecolari, permettendo di collegare direttamente il profilo genetico del tumore alle terapie e alle risposte cliniche.

Infine, il grafo integra anche la dimensione organizzativa della cura attraverso *cvl:Service* e *cvl:Department*, che consentono di modellare, rispettivamente, il servizio erogato e l'unità organizzativa responsabile dell'erogazione.

4.2 - Integrazione di ontologie biomediche

La solidità semantica dell'ontologia di CVL è garantita dall'integrazione delle principali ontologie biomediche.

- SNOMED CT costituisce il vocabolario clinico di base dell'ontologia CVL. Viene utilizzato per rappresentare in modo standardizzato diagnosi cliniche, stadi di malattia, punteggi ECOG, misure corporee, fattori di stile di vita, sintomi del paziente, consentendo di modellare la condizione clinica del paziente in modo interoperabile e computabile.
- LOINC fornisce la codifica formale per esami di laboratorio, biomarcatori e test molecolari, inclusi parametri ematochimici, test immunoistochimici e risultati di biologia molecolare. Questo permette di trattare le osservazioni di laboratorio come entità semanticamente definite, confrontabili tra istituzioni e utilizzabili per analisi quantitative.
- NCIt (NCI Thesaurus) rappresenta il nucleo ontologico oncologico del sistema. È utilizzato per modellare tipologie tumorali, classificazioni TNM, stadi di malattia, biomarcatori oncologici, regimi e linee di terapia, fornendo una struttura concettuale coerente per la rappresentazione della patologia oncologica.
- MedDRA, in integrazione con CTCAE, consente di rappresentare in modo rigoroso eventi avversi, tossicità e reazioni indesiderate ai trattamenti. Questo è essenziale per l'analisi della sicurezza terapeutica, per il monitoraggio delle complicanze e per la valutazione del profilo rischio–beneficio delle terapie.
- ICD-10 garantisce l'allineamento con i sistemi informativi ospedalieri e amministrativi, permettendo di collegare il knowledge graph ai flussi di codifica clinica utilizzati nella pratica sanitaria reale.
- ATC consente di rappresentare le terapie in termini di sostanze attive, classi farmacologiche e meccanismi d'azione, rendendo possibile l'analisi dei trattamenti sia a livello molecolare sia a livello di categoria terapeutica.
- HGNC (HUGO Gene Nomenclature Committee) fornisce la nomenclatura standardizzata dei geni, permettendo di integrare nel knowledge graph dati genomici, mutazioni e biomarcatori molecolari in modo univoco e interoperabile.

Questa integrazione permette al grafo di CVL di rappresentare in modo coerente e computabile l'intero spettro dell'informazione oncologica. L'adozione di terminologie standard e identificatori condivisi rende inoltre il knowledge graph intrinsecamente federabile, favorendo l'integrazione con dataset e knowledge graph esterni senza necessità di centralizzazione dei dati.

4.3 - Reasoner

Nel Cancer Virtual Lab il knowledge graph non è un semplice archivio di dati clinici, ma una base di conoscenza attiva, grazie all'integrazione di un motore di ragionamento semantico che implementa le regole di inferenza OWL 2 RL e opera sulle ontologie CVL e sui dati RDF memorizzati nel triplestore. Il *reasoner* costituisce il nucleo logico del sistema e, applicando le definizioni formali delle ontologie biomediche e le regole dichiarative del dominio clinico, consente di:

- inferire ed esplicitare conoscenza ontologica implicita, come gerarchie e relazioni definite nell'ontologia CVL
- verificare la coerenza e la correttezza dei dati clinici rispetto ai vincoli del modello
- supportare inferenze clinicamente rilevanti per analisi e supporto decisionale

Oltre alla funzione inferenziale, il reasoner svolge un ruolo fondamentale nella validazione dei dati. In combinazione con i vincoli SHACL, esso verifica che le istanze del knowledge graph rispettino le regole cliniche e strutturali del modello. Questo meccanismo impedisce che informazioni incomplete, incoerenti o clinicamente implausibili entrino nel grafo o vengano utilizzate per analisi e supporto decisionale.

Dal punto di vista dell'intelligenza artificiale, l'impiego del reasoner consente di realizzare un'*Explainable AI by design*, in cui ogni risposta è fondata su relazioni ontologiche e regole formali esplicite, anziché su meccanismi di apprendimento opachi. I risultati prodotti dal sistema possono quindi essere ricondotti a catene di ragionamento verificabili, rendendo le conclusioni clinicamente interpretabili, giustificabili e auditabili.

4.4 - Interrogazione del grafo

L'accesso e l'esplorazione del patrimonio informativo rappresentato nel grafo avvengono attraverso un endpoint *SPARQL 1.1–compliant*, che espone l'intero contenuto del knowledge graph per l'interrogazione semantica. In CVL l'endpoint SPARQL 1.1 è fornito dal triplestore che ospita il grafo, tramite un'interfaccia standard conforme alle specifiche W3C, consentendo interrogazioni semantiche direttamente sui concetti ontologici.

4.4.1 - SPARQL

SPARQL (SPARQL Protocol and RDF Query Language) è il linguaggio standard del W3C per interrogare dati rappresentati in formato RDF all'interno di un knowledge graph. A differenza di linguaggi di interrogazione relazionali come SQL, SPARQL non si limita all'estrazione dei dati, ma supporta anche operazioni di aggiornamento e modifica del grafo, consentendo l'inserimento, la cancellazione e la trasformazione di triple RDF tramite istruzioni come INSERT, DELETE e UPDATE. Questa caratteristica rende SPARQL uno strumento non solo di interrogazione, ma anche di gestione attiva del knowledge graph, permettendo di arricchire dinamicamente la base di conoscenza mantenendo coerenza con il modello semantico sottostante.

Una query SPARQL si costruisce definendo innanzitutto i prefissi, che abbreviano gli identificatori delle ontologie utilizzate, e poi specificando un blocco SELECT che indica quali variabili si vogliono ottenere. La parte centrale della query è la clausola WHERE, nella quale si descrivono i pattern di triple che devono essere soddisfatti nel grafo, ad esempio una specifica entità che ha una determinata proprietà con un certo valore. Il motore SPARQL ricerca nel knowledge graph tutte le combinazioni di soggetti, predicati e oggetti che rispettano quei vincoli e restituisce le istanze che li soddisfano. Attraverso costrutti come FILTER, OPTIONAL, GROUP BY e funzioni aggregate è possibile raffinare ulteriormente la query, ottenendo interrogazioni espressive e controllabili, in grado di estrarre informazione strutturata dal grafo.

4.4.2 - Esempi di query SPARQL in CVL

Di seguito sono riportati due esempi di query SPARQL utilizzate per esplorare e analizzare il contenuto del knowledge graph del Cancer Virtual Lab.

La prima query (Figura 4.2) restituisce, per ciascun principio attivo, il numero totale di terapie in cui è stato somministrato e la durata media delle terapie che lo utilizzano, espressa in mesi. I risultati dell'interrogazione (Figura 4.3) evidenziano una netta differenziazione nei pattern di utilizzo: alcuni principi attivi risultano altamente frequenti, come cyclophosphamide e paclitaxel, mentre altri, pur meno ricorrenti, presentano durate medie di trattamento più elevate, come letrozole e palbociclib.

```
PREFIX cvl:  <http://irst.emr.it/cvl/>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd:  <http://www.w3.org/2001/XMLSchema#>

SELECT
  ?activeSubstance
  (COUNT(DISTINCT ?therapy) AS ?numTherapies)
  (ROUND(AVG(?monthsDuration)) AS ?avgMonthsDuration)
WHERE {
  {
    SELECT DISTINCT
      ?therapy
      ?activeSubstance
      ?monthsDuration
    WHERE {
      ?therapy a cvl:Therapy ;
               cvl:first_administration_date ?firstAdmin ;
               cvl:last_administration_date  ?lastAdmin ;
               cvl:has_administration ?administration .

      ?administration cvl:administer_active_substance ?aS .
      ?aS cvl:active_substance_description ?activeSubstance .

      BIND( YEAR(xsd:dateTime(?lastAdmin))  - YEAR(xsd:dateTime(?firstAdmin)) AS ?dYears  )
      BIND( MONTH(xsd:dateTime(?lastAdmin)) - MONTH(xsd:dateTime(?firstAdmin)) AS ?dMonths )
      BIND( (?dYears * 12) + ?dMonths AS ?monthsDuration )
    }
  }
}
GROUP BY
  ?activeSubstance
ORDER BY
  DESC(?numTherapies)
  DESC(?avgMonthsDuration)
```

Figura 4.2 - Query SPARQL (query 1)

activeSubstance	numTherapies	avgMonthsDuration
cyclophosphamide	2673	8.0
paclitaxel	2056	11.0
trastuzumab	1520	14.0
doxorubicin	1252	10.0
docetaxel	922	8.0
epirubicin	891	8.0
fluorouracil	811	9.0
capecitabine	799	12.0
fulvestrant	569	15.0
vinorelbine	555	12.0
methotrexate	393	9.0
carboplatin	392	7.0
eribulin	264	7.0
gemcitabine	195	6.0
pertuzumab	191	13.0
letrozole	173	27.0
palbociclib	171	16.0
lapatinib	129	8.0

Figura 4.3 - Risultato query SPARQL (query 1)

La seconda query (Figura 4.4) analizza il profilo di sicurezza dei trattamenti, mostrando, per ciascun principio attivo, il numero totale di eventi avversi associati alle somministrazioni, il numero di eventi con grado di tossicità severo (valore ≥ 3) e la percentuale di eventi severi sul totale. I risultati dell'interrogazione sono riportati in Figura 4.5 e mostrano che, nel dataset considerato, alcuni principi attivi presentano una quota proporzionalmente più elevata di eventi avversi severi rispetto al totale, come fluorouracil e gemcitabine. L'ordinamento per percentuale evidenzia inoltre come farmaci con un numero assoluto di eventi più contenuto possano comunque risultare critici in termini relativi, confermando l'utilità dell'indicatore percentuale nel confronto tra trattamenti.

```
SELECT
  ?activeSubstanceCode
  ?activeSubstanceDescription
   (COUNT(DISTINCT ?effect) AS ?totalEffects)
  (SUM(IF(?toxicityGradeValue >= 3, 1, 0)) AS ?severeEffects)
  (CONCAT(
     STR(
       ROUND(
         (
           xsd:decimal(SUM(IF(?toxicityGradeValue >= 3, 1, 0))) * 100.0
           / xsd:decimal(COUNT(DISTINCT ?effect))
         ) * 100.0
       ) / 100.0
     ),
     "%"
   ) AS ?percSevereEffects)
WHERE {
  ?therapy a cvl:Therapy .
  ?therapy        cvl:has_administration   ?administration .
  ?administration cvl:causes               ?adverseEvent .
  ?adverseEvent   a                        cvl:AdverseEvent .

  ?administration cvl:administer_active_substance ?activeSubstance .
  ?activeSubstance cvl:active_substance_code ?activeSubstanceCodeIRI .
  BIND(REPLACE(STR(?activeSubstanceCodeIRI), ".*/", "") AS ?activeSubstanceCode)
  ?activeSubstance cvl:active_substance_description ?activeSubstanceDescription .

  ?adverseEvent cvl:has_effect ?effect .
  ?effect       cvl:toxicity_grade_value ?toxicityGradeValue .
}
GROUP BY
?activeSubstanceCode
?activeSubstanceDescription
ORDER BY DESC(
  xsd:decimal(SUM(IF(?toxicityGradeValue >= 3, 1, 0))) * 100.0
  / xsd:decimal(COUNT(DISTINCT ?effect))
)
```

Figura 4.4 - Query SPARQL (query 2)

activeSubstanceCode	activeSubstanceDescription	totalEffects	severeEffects	percSevereEffects
L01BC02	fluorouracil	157	12	7.64%
L01BC05	gemcitabine	365	23	6.3%
L01FG01	bevacizumab	16	1	6.25%
L01XK01	olaparib	48	3	6.25%
L01BA01	methotrexate	283	11	3.89%
L01CA04	vinorelbine	916	34	3.71%
L01EM03	alpelisib	38	1	2.63%
L01XA02	carboplatin	1615	40	2.48%
L01EG02	everolimus	179	4	2.23%
L01XA01	cisplatin	45	1	2.22%
L01CD02	docetaxel	866	19	2.19%
L01XX41	eribulin	745	16	2.15%
L01DB01	doxorubicin	881	17	1.93%
L01BC06	capecitabine	2660	49	1.84%
L01FD02	pertuzumab	1584	25	1.58%
L02BG03	anastrozole	64	1	1.56%
L01EH01	lapatinib	193	3	1.55%
L01AA01	cyclophosphamide	2152	33	1.53%

Figura 4.5 - Risultato query SPARQL (query 2)

CAPITOLO 5
ARCHITETTURA E COMPONENTI DI CVL

5.1 - Architettura logica

Il Cancer Virtual Lab è progettato secondo un'architettura a tre livelli che consente di separare in modo netto la gestione dei dati, i processi di elaborazione e l'interazione con l'utente. Questa organizzazione garantisce modularità e scalabilità, requisiti essenziali in un contesto clinico e di ricerca che tratta grandi volumi di dati eterogenei e ad alta complessità semantica.

- Il livello di persistenza è basato su un knowledge graph RDF ospitato in GraphDB, che funge da triplestore semantico della piattaforma. Questo livello non si limita alla memorizzazione dei dati, ma ne consente anche l'interrogazione semantica tramite SPARQL, rendendo possibile esprimere relazioni cliniche complesse e collegare informazioni provenienti da fonti eterogenee in modo coerente e computabile. Il knowledge graph costituisce pertanto la base informativa unica e condivisa dell'intera piattaforma.
- Il livello di logica applicativa rappresenta lo strato intermedio dell'architettura del Cancer Virtual Lab ed è responsabile dell'integrazione tra knowledge graph, servizi applicativi e componenti di intelligenza artificiale. In questo livello viene orchestrata la pipeline di trasformazione dei dati, che standardizza i dati clinici in HL7-FHIR e ne realizza la conversione e il caricamento nel knowledge graph in formato RDF. Vi operano inoltre i moduli che espongono l'accesso applicativo ai dati del grafo e i componenti di AI utilizzati per attività di analisi e supporto decisionale. In questo livello si collocano infine i moduli dedicati alla generazione controllata di dataset sintetici e alla relativa validazione.
- Il livello di presentazione è implementato tramite un'interfaccia web basata su Streamlit, che rappresenta il punto di accesso degli utenti alla piattaforma. Attraverso questa interfaccia è possibile esplorare coorti di pazienti, interrogare il knowledge graph, visualizzare informazioni cliniche e accedere ai servizi di intelligenza artificiale. L'interfaccia comunica con i servizi del livello di logica applicativa, mantenendo una separazione rispetto al livello di persistenza.

5.2 - Architettura fisica

L'architettura del Cancer Virtual Lab è organizzata in più macro-gruppi di servizi, ciascuno dei quali realizza un livello funzionale distinto della piattaforma. Tutti i componenti sono interconnessi tramite una rete Docker esterna comune (*cvl_network*), che funge da *backbone* di comunicazione dell'intero sistema.

All'interno della rete containerizzata, ogni servizio è identificato dal proprio nome logico e comunica con gli altri esclusivamente tramite API interne, senza esposizione diretta verso l'esterno. Questo consente di separare l'architettura applicativa dall'accesso pubblico e di applicare politiche di sicurezza e isolamento adeguate a un contesto clinico.

L'adozione di una rete Docker condivisa consente al Cancer Virtual Lab di operare come una piattaforma distribuita e modulare, nella quale i diversi servizi sono orchestrati in modo coordinato pur mantenendo isolamento, controllo degli accessi e scalabilità, requisiti essenziali per un'infrastruttura destinata alla ricerca biomedica e all'elaborazione di dati clinici.

5.2.1 - cvlapi

Il livello cognitivo della piattaforma è realizzato dal servizio cvlapi, un'API basata su FastAPI ed eseguita tramite Uvicorn sulla porta 6000, che funge da orchestratore tra interfaccia utente, modelli di intelligenza artificiale e knowledge graph. Tramite variabili di ambiente, il servizio è configurato per collegarsi al runtime dei modelli LLM (Ollama) e all'API del knowledge graph (kg-api), coordinando l'intero flusso di interrogazione.

cvlapi implementa una pipeline che interpreta le richieste in linguaggio naturale, recupera contesto semantico tramite *vector store*, genera automaticamente query SPARQL coerenti con l'ontologia e le esegue sul knowledge graph, restituendo i risultati in forma tabellare e descrittiva. In questo modo costituisce il punto di integrazione operativo tra intelligenza artificiale e conoscenza clinica del Cancer Virtual Lab.

5.2.2 - fhir_converter

Il container fhir_converter esegue la conversione dei dati clinici in formato tabellare CSV in FHIR JSON standardizzati selezionando il dataset di interesse tramite il parametro --mode (sample, full, breast).

Per ciascun dominio clinico viene prodotto un file JSON dedicato mentre i dataset di grandi dimensioni vengono elaborati a blocchi generando più file progressivi per assicurare scalabilità e controllo della memoria.

5.2.3 - rdf_converter

Il servizio rdf_converter realizza la conversione dei file FHIR JSON in RDF/Turtle (.ttl). Per ciascun dominio clinico, il container legge i file JSON prodotti dal livello precedente e genera il corrispondente grafo RDF utilizzando *rdflib*, caricando un vocabolario FHIR in Turtle come base semantica.
I container fhir_converter e rdf_converter non sono servizi persistenti, ma *job batch-oriented:* vengono avviati, elaborano i dati, producono output e terminano. Questa modalità garantisce tracciabilità, riproducibilità e isolamento delle diverse popolazioni di dati.

5.2.4 - graphdb

Il container graphdb ospita l'istanza del database semantico Ontotext GraphDB. Questo servizio costituisce il triplestore persistente del knowledge graph di CVL e risiede sulla porta 7200. Attraverso un volume Docker, i dati del grafo sono persistenti, garantendo durabilità, versionamento e possibilità di ripristino.

5.2.5 - kg-api

L'accesso applicativo al knowledge graph è mediato dal servizio kg-api, un'API FastAPI esposta sulla porta 7202 che incapsula l'interazione con GraphDB. Questo livello fornisce funzionalità per l'esecuzione di query SPARQL.

5.2.6 - kg-initializer

Il container kg_initializer rappresenta la fase finale della pipeline e svolge una funzione di inizializzazione completa del repository semantico del Cancer Virtual Lab.

In una prima fase, il servizio crea il repository utilizzando un file di configurazione RDF4J/GraphDB che definisce in modo dichiarativo l'identità del repository, il ruleset di inferenza adottato, le politiche di consistenza e i principali parametri di indicizzazione e interrogazione.

Successivamente vengono caricati sia l'ontologia CVL sia i file RDF generati dalla pipeline FHIR.

Una fase ulteriore consiste nell'esecuzione di query SPARQL UPDATE di tipo INSERT, che creano nuove risorse e relazioni nel namespace CVL a partire dalle risorse HL7-FHIR già presenti nel grafo.

Infine, il servizio esegue operazioni di verifica e monitoraggio basate su metriche di qualità dell'ontologia e del grafo, come il controllo della presenza di etichette (rdfs:label), la consistenza dei datatype, il rispetto di vincoli funzionali, nonché la conformità a domain e range e l'assenza di entità in classi disgiunte o di classi/proprietà non definite.

5.2.7 - streamlit

Il livello di presentazione è fornito dal container UI (streamlit), accessibile sulla porta 8501, che rappresenta il punto di contatto tra l'infrastruttura semantica del CVL e gli utenti finali. Operando nella rete cvl_network, l'interfaccia comunica direttamente con kg-api per interrogare il knowledge graph e con cvlapi per accedere ai servizi di intelligenza artificiale.

5.3 - Tecnologie

5.3.1 - GraphDB

Nel Cancer Virtual Lab il knowledge graph è implementato su GraphDB, un database semantico RDF progettato per l'archiviazione, l'interrogazione e il ragionamento su grandi basi di conoscenza ontologiche. GraphDB non svolge il ruolo di semplice repository di triple, ma costituisce l'infrastruttura semantica che consente al sistema di operare come una vera base di conoscenza clinica. Attraverso il supporto nativo agli standard RDF, RDFS e OWL, GraphDB consente di

rappresentare entità cliniche, relazioni e vincoli ontologici in forma formalmente rigorosa, garantendo interoperabilità e coerenza semantica.

Una caratteristica centrale di GraphDB nel contesto di CVL è l'integrazione del reasoning ontologico direttamente a livello di database. Il motore di inferenza di GraphDB applica automaticamente le regole OWL e le gerarchie ontologiche ai dati memorizzati, rendendo esplicite le conoscenze implicite nel grafo. In questo modo, le query SPARQL non interrogano solo fatti espliciti, ma una base di conoscenza arricchita da inferenze semanticamente fondate. Il reasoning non si limita ad arricchire il grafo mediante l'inferenza di nuove triple, ma è essenziale anche per la verifica della coerenza del knowledge graph. In particolare, GraphDB consente di effettuare controlli di consistenza (*consistency check*), finalizzati a individuare contraddizioni logiche rispetto all'ontologia, come l'appartenenza simultanea di una stessa entità a classi dichiarate disgiunte, oppure l'uso di proprietà in modo incompatibile con i vincoli ontologici definiti. Questo controllo permette di intercettare errori che non emergerebbero tramite una semplice validazione sintattica o tramite l'ispezione manuale dei dati.

GraphDB fornisce inoltre un supporto avanzato per la validazione e la qualità dei dati attraverso l'integrazione con SHACL, permettendo di verificare che le istanze del knowledge graph rispettino i vincoli strutturali e clinici definiti dall'ontologia CVL. Questo è particolarmente rilevante in un contesto sanitario, in cui l'affidabilità e la correttezza delle informazioni sono requisiti imprescindibili. La possibilità di eseguire validazioni automatiche direttamente sul repository consente di intercettare incoerenze, errori di modellazione o violazioni delle regole cliniche prima che i dati vengano utilizzati per analisi o supporto decisionale.

Dal punto di vista dell'accesso ai dati, GraphDB espone un endpoint SPARQL 1.1 che consente interrogazioni complesse, aggregazioni, filtri temporali e navigazione delle relazioni tra entità cliniche. In CVL questo endpoint costituisce il punto di connessione tra il knowledge graph e i livelli superiori dell'architettura. Le richieste formulate in linguaggio naturale vengono tradotte in query SPARQL che vengono eseguite su GraphDB, garantendo che ogni risposta sia derivata direttamente dalla base di conoscenza semantica e sia quindi tracciabile e spiegabile.

Infine, la scelta di GraphDB consente al Cancer Virtual Lab di operare su grafi di dimensione reale, con milioni di triple e centinaia di migliaia di istanze cliniche, mantenendo prestazioni adeguate e supportando sia l'analisi esplorativa sia l'inferenza semantica.

5.3.2 - Streamlit

Streamlit è una piattaforma open-source per lo sviluppo di applicazioni web interattive orientate alla data science, progettata per consentire la trasformazione rapida di codice Python in interfacce utente dinamiche e fruibili via browser. A differenza dei framework web tradizionali, che richiedono la separazione tra logica di back-end e front-end, Streamlit adotta un paradigma unificato in cui l'interfaccia viene generata direttamente dall'esecuzione del codice Python, riducendo drasticamente la complessità di sviluppo e favorendo la prototipazione rapida di strumenti analitici avanzati.

Una caratteristica fondamentale di Streamlit è il suo modello di esecuzione reattivo: ogni interazione dell'utente innesca automaticamente il ricalcolo della parte rilevante dell'applicazione. Questo consente di costruire interfacce altamente dinamiche, in cui grafici, tabelle e visualizzazioni si aggiornano in tempo reale in risposta alle scelte dell'utente, senza la necessità di gestire manualmente eventi o stati complessi.

Streamlit fornisce inoltre un insieme di componenti ad alto livello per la visualizzazione dei dati, tra cui grafici interattivi, tabelle, metriche, controlli di input e layout responsive, che permettono di creare dashboard e ambienti di esplorazione dei dati con poche righe di codice. Questi componenti sono progettati per integrarsi nativamente con le principali librerie dell'ecosistema scientifico Python, come Pandas, NumPy, Matplotlib, Plotly e Scikit-learn, rendendo possibile la visualizzazione immediata dei risultati di analisi.

Dal punto di vista architetturale, Streamlit supporta la gestione dello stato applicativo, permettendo di mantenere contesti di lavoro persistenti durante la navigazione dell'utente, e facilita l'integrazione con servizi esterni tramite API, file system e database. Inoltre, la sua esecuzione in modalità *server-side* consente di demandare al back-end l'elaborazione e i calcoli intensivi, lasciando al browser unicamente la renderizzazione dell'interfaccia e dei risultati, con benefici in termini di prestazioni e controllo dell'esposizione dei dati. Grazie a tali caratteristiche, Streamlit risulta particolarmente adatto allo sviluppo di applicazioni *data-driven,* in cui l'obiettivo non è la mera presentazione di contenuti, ma l'interazione continua tra l'utente e processi di analisi, interrogazione e trasformazione dei dati.

5.3.3 - Ollama, LangChain, LangGraph, Chroma

Il Cancer Virtual Lab integra un articolato *stack* di intelligenza artificiale progettato per consentire l'accesso cognitivo alla conoscenza clinica. Tale stack è costruito attorno a un *Large Language Model*

eseguito localmente tramite Ollama, configurato nel sistema come modello di base ad alta capacità (gpt-oss:20b), evitando il ricorso a servizi cloud esterni e garantendo il pieno controllo sui dati sensibili. L'LLM è integrato attraverso il framework LangChain, che fornisce i meccanismi di orchestrazione delle richieste, la gestione del contesto e l'interazione con le fonti di conoscenza.

Per supportare il paradigma di *Retrieval-Augmented Generation* (RAG), CVL utilizza Chroma come database vettoriale, nel quale vengono indicizzate porzioni dell'ontologia, documentazione clinica e metadati mediante modelli di embedding di grandi dimensioni, come Qwen3-Embedding-8B. Questo consente di effettuare ricerche semantiche ad alta precisione, permettendo al sistema di recuperare i frammenti di conoscenza più rilevanti per una data domanda prima di coinvolgere l'LLM nella generazione della risposta. L'interazione tra retrieval vettoriale e generazione linguistica garantisce che le risposte siano ancorate a contenuti reali e semanticamente coerenti.

La logica di esecuzione e di controllo dei flussi conversazionali è gestita tramite LangGraph, che permette di modellare il comportamento dell'agente come un grafo di stati e transizioni. Questo consente di implementare strategie di interrogazione multipasso, validazione delle risposte, chiamate ripetute al retrieval e integrazione strutturata con le query SPARQL verso GraphDB. Nel loro insieme, questi strumenti realizzano una architettura di AI ibrida, in cui modelli linguistici e ricerca semantica cooperano per fornire un accesso robusto, spiegabile e controllato ai dati clinici del Cancer Virtual Lab.

5.3.4 - Docker

L'intera piattaforma è containerizzata tramite Docker e orchestrata mediante Docker Compose, garantendo isolamento dei servizi, modularità e portabilità dell'infrastruttura tra ambienti di sviluppo, test e produzione. Questo approccio consente di mantenere separati i moduli di trasformazione dei dati, di interfaccia, di intelligenza artificiale, di gestione del knowledge graph e di generazione dei dati sintetici riducendo la complessità operativa e favorendo la scalabilità del sistema.

5.3.5 - Python

Il nucleo computazionale dell'intera piattaforma è sviluppato in Python, che costituisce il linguaggio unificante di tutti i servizi e i moduli del sistema. Python consente di integrare in modo uniforme librerie per la manipolazione dei dati (come Pandas e NumPy), strumenti di machine learning e deep

learning (come PyTorch e Sentence-Transformers) e librerie semantiche per la gestione dei grafi RDF. In questo contesto, rdflib svolge un ruolo centrale, fornendo gli strumenti per creare, manipolare, interrogare e serializzare grafi RDF in modo programmatico. Attraverso rdflib, il sistema gestisce le triplette RDF e realizza il collegamento operativo tra dati strutturati, modelli ontologici e rappresentazioni in formato Turtle, che alimentano sia il knowledge graph sia i moduli di generazione dei dati sintetici.

5.3.6 - SHACL

Nel Cancer Virtual Lab, SHACL è utilizzato come linguaggio formale per la specifica dei vincoli semantici applicati al knowledge graph clinico e come base per la generazione controllata di dati sintetici, come descritto nel Capitolo 2.1.1.

5.3.7 - RDF Graph Generator

Nel Cancer Virtual Lab, l'adozione di RDFGraphGen, descritto nel Capitolo 2.1.2, è coerente con una scelta architetturale che utilizza i vincoli SHACL non esclusivamente come strumento di validazione, ma come specifica generativa del dominio informativo. In questa impostazione, il knowledge graph rappresenta la fonte primaria di verità strutturale del sistema, e i dataset sintetici vengono prodotti come istanze conformi al modello semantico della piattaforma. L'approccio *SHACL-driven* consente di rendere il processo di generazione formalizzato, riproducibile e controllabile, garantendo allineamento tra ontologia, knowledge graph e dati generati. I dati sintetici prodotti non costituiscono una replica dei dati reali, ma una realizzazione artificiale del modello che descrive il dominio clinico, in linea con i requisiti di coerenza semantica e tutela della privacy.

5.4 - Explainability e tracciabilità

L'uso del knowledge graph come base di conoscenza consente di superare la natura opaca tipica dei modelli *black-box*. L'AI di CVL è infatti *explainable by design:* le conclusioni prodotte dal sistema possono essere comprese e verificate ricostruendo il percorso inferenziale che le ha generate, e quindi collegandole in modo esplicito ai dati clinici, alle relazioni ontologiche e alle regole decisionali applicate.

Per migliorare ulteriormente l'interpretabilità dei risultati delle interrogazioni, le entità del grafo sono annotate con etichette leggibili dall'essere umano tramite la proprietà rdfs:label. Queste etichette garantiscono che gli output delle query SPARQL possano essere compresi direttamente da clinici e ricercatori, senza la necessità di interpretare URI opachi o identificativi tecnici.

L'*explainability* soddisfa non solo le esigenze dei clinici in termini di interpretabilità, ma anche i requisiti normativi europei di trasparenza, tracciabilità e controllo umano previsti dall'AI Act e dallo European Health Data Space.

In particolare, l'AI Act stabilisce che, nei contesti ad alto rischio come la sanità, i sistemi di intelligenza artificiale non possano mai sostituire il giudizio umano. Poiché i modelli di deep learning operano spesso come black box e non sono intrinsecamente affidabili, la decisione finale deve sempre restare al medico, secondo il principio dello *human-in-the-loop*, che impone supervisione continua e responsabilità umana sulle decisioni generate dall'AI. L'uso dell'AI è quindi ammesso come supporto al processo decisionale, non come suo sostituto, con requisiti tanto più stringenti quanto più elevato è il rischio applicativo.

Lo European Health Data Space si colloca nello stesso quadro, imponendo che i dati sanitari e i sistemi che li utilizzano siano tracciabili, interoperabili e utilizzabili in modo controllato, così da garantire che l'innovazione basata sui dati avvenga senza compromettere sicurezza, diritti del paziente e responsabilità clinica.

5.5 - Architettura LLM-augmented per l'interrogazione del knowledge graph

Un elemento distintivo di CVL è l'integrazione nativa tra dati clinici, knowledge graph e modelli linguistici (LLM). I dati, dopo la normalizzazione e validazione tramite pipeline FHIR, vengono caricati nel CVL Knowledge Graph, che assume il ruolo di fonte primaria di verità del sistema. Su questo strato semantico si innesta l'LLM, utilizzando il grafo come base strutturata per l'accesso e l'elaborazione della conoscenza clinica.

CVL adotta un'architettura *LLM-augmented* che consente di interrogare il knowledge graph in linguaggio naturale, superando la barriera tecnica dei linguaggi formali. Le richieste dell'utente vengono interpretate, tradotte in interrogazioni strutturate, eseguite sul grafo e ricostruite in risposte leggibili e clinicamente coerenti. Tale impostazione, riconducibile al paradigma *Retrieval-Augmented*

Generation (RAG), vincola la generazione alle informazioni effettivamente presenti e interrogabili nel grafo, migliorando tracciabilità e riducendo il rischio di allucinazioni.

Questa integrazione trasforma CVL in un *AI Agent* clinico, capace di orchestrare interrogazioni complesse sul knowledge graph, a supporto del processo decisionale sotto controllo umano.

In particolare, l'agente del CVL può:

- interpretare richieste cliniche in linguaggio naturale
- generare una o più query SPARQL
- eseguirle sul knowledge graph
- restituire risposte tracciabili, verificabili e clinicamente significative

Ne emerge un nuovo paradigma di interazione con i dati clinici: il medico non interroga più un archivio, ma interagisce con un sistema in grado di comprendere la richiesta, esplorare la struttura semantica dell'oncologia e produrre risposte rigorose e spiegabili.

CAPITOLO 6

INTERFACCIA CANCER VIRTUAL LAB

L'interfaccia del Cancer Virtual Lab è organizzata come un insieme di moduli funzionali, attraverso i quali l'utente può accedere a diverse modalità di utilizzo del knowledge graph clinico. Ogni modulo corrisponde a una specifica funzione della piattaforma (interrogazione, filtraggio, analisi, esplorazione, modellazione ontologica o generazione di dati) ed è selezionabile direttamente dall'interfaccia dopo avere scelto il repository su cui si vuole lavorare.

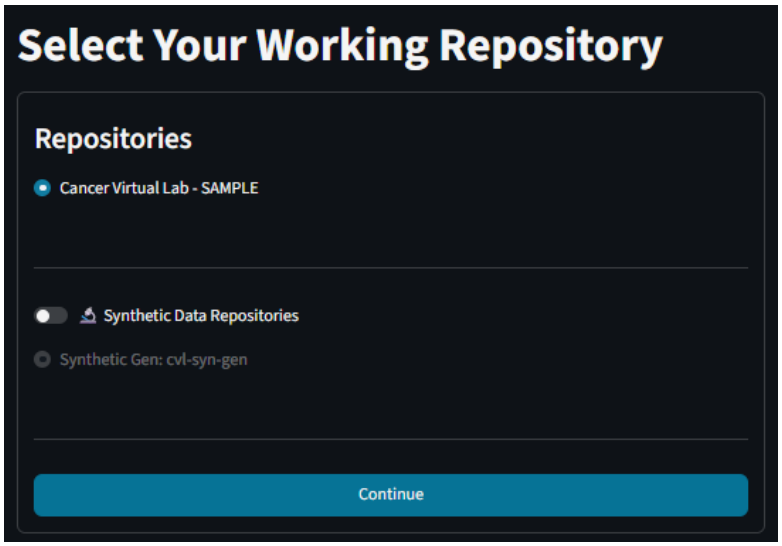


Figura 6.1 - Selezione del repository

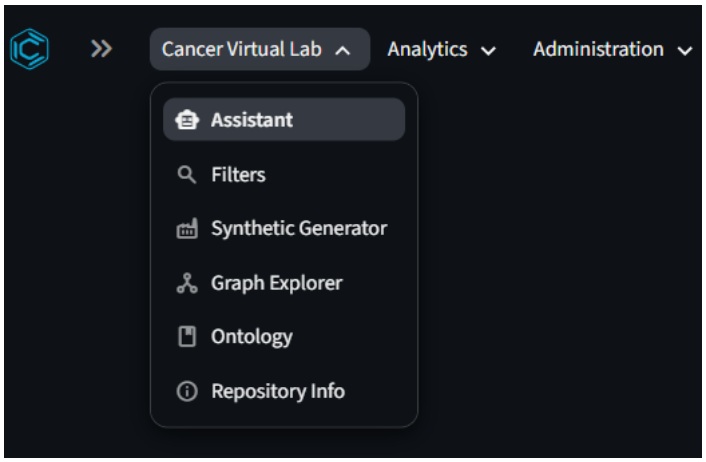


Figura 6.2 - Menu di navigazione dell'interfaccia

6.1 - AI Assistant

La pagina *AI Assistant* costituisce il punto di ingresso più immediato per l'utente. Essa consente di formulare domande in linguaggio naturale, come "trovare pazienti con diagnosi di mammella" o "ottenere tutti gli eventi avversi con descrizione anoressia". Queste richieste non vengono trattate come semplici stringhe di testo, ma come espressioni semantiche che contengono concetti clinici, relazioni e vincoli.

Il sistema utilizza un *Large Language Model* come interprete semantico, in grado di riconoscere entità cliniche e mapparle ai corrispondenti nodi e proprietà dell'ontologia CVL. A partire da questa rappresentazione intermedia, viene generata automaticamente una query SPARQL, che interroga il knowledge graph. I risultati ottenuti vengono quindi rielaborati in una forma leggibile e clinicamente interpretabile. In questo modo, l'utente può interrogare un grafo RDF ontologico senza conoscere il linguaggio formale sottostante, mantenendo tracciabilità del processo e possibilità di ricondurre ogni risposta alla query SPARQL che l'ha generata.

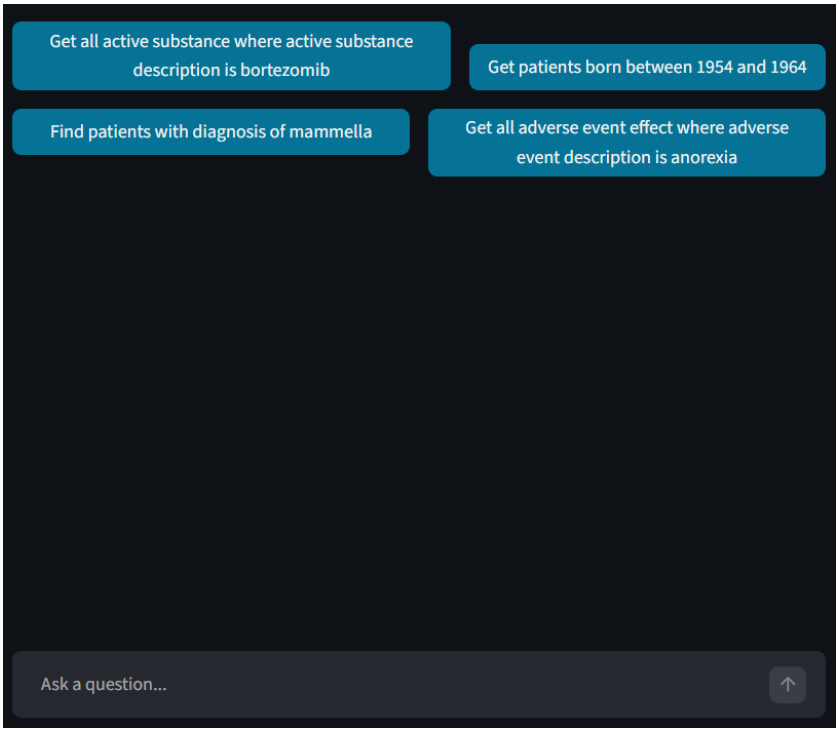


Figura 6.3 - Interfaccia AI assistant per l'interrogazione semantica del knowledge graph mediante linguaggio naturale

6.2 - Costruzione di coorti

La pagina *Filter the Graph* consente di costruire coorti cliniche in modo interattivo e controllato. L'utente definisce progressivamente i criteri di selezione, ad esempio filtrando i pazienti in base a una specifica diagnosi, al sito tumorale, al trattamento somministrato o ad altre variabili cliniche rilevanti. Ogni filtro contribuisce alla definizione incrementale di un sottografo del knowledge graph di CVL, delimitando progressivamente l'insieme di entità e relazioni rilevanti rispetto ai criteri clinici selezionati.

La storia dei filtri applicati è rappresentata come una catena semantica di relazioni, ad esempio *Patient* → *has diagnosis* → *Mammella femminile*. In tempo reale il sistema mostra il numero di istanze che soddisfano tali vincoli.

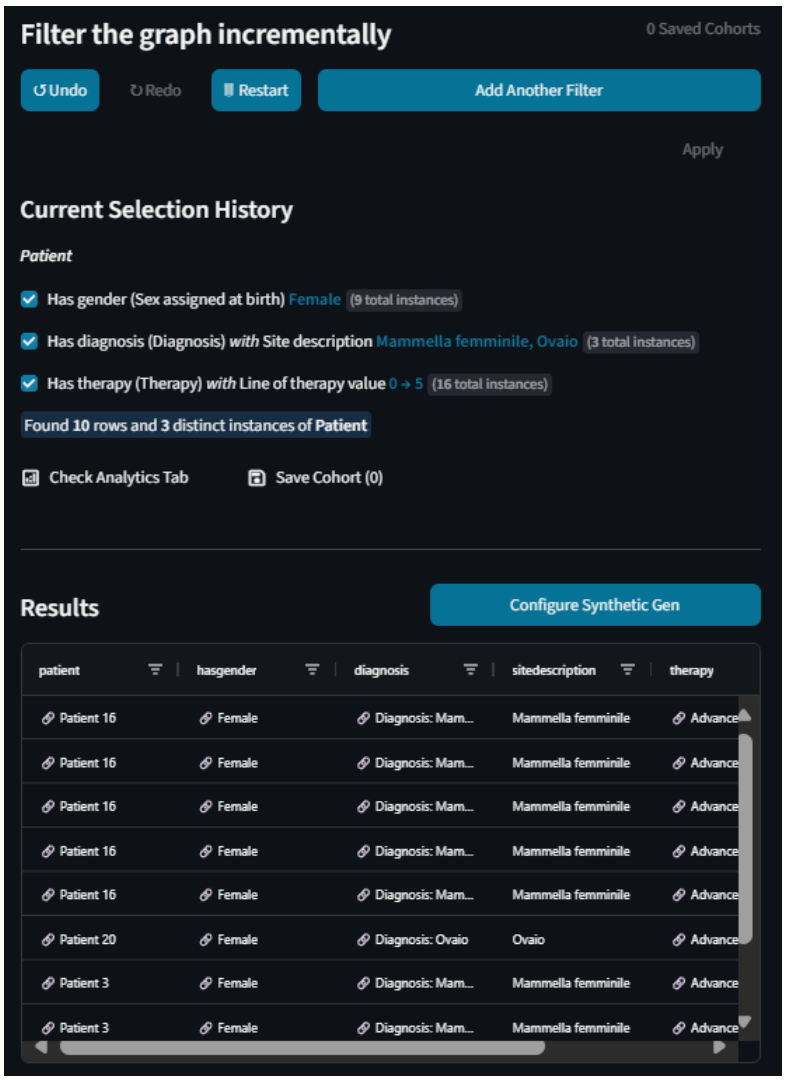


Figura 6.4 - Costruzione incrementale di coorti cliniche mediante filtri semantici

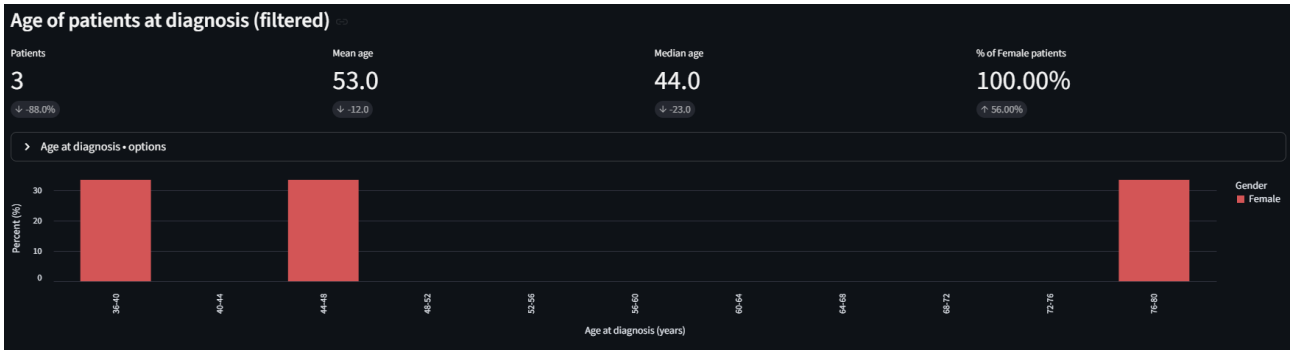
6.3 - Analisi della coorte

Dopo avere definito la coorte tramite i filtri, la scheda *Analytics* consente di visualizzare un insieme di indicatori e grafici derivati dai pazienti selezionati. Le analisi disponibili si articolano in tre sezioni: descrittiva, predittiva e prescrittiva.

La sezione di analisi descrittiva presenta una sintesi della coorte filtrata, riportando la distribuzione dell'età alla diagnosi e dell'età al decesso, accompagnate da metriche aggregate (ad es. media e mediana). La pagina include inoltre la distribuzione delle sedi di diagnosi, rappresentata mediante una visualizzazione percentuale aggregata.

La sezione di analisi predittiva presenta un'analisi di sopravvivenza complessiva della coorte filtrata mediante curva di Kaplan-Meier, accompagnata da indicatori sintetici (numero di pazienti, percentuale di decessi, età media alla diagnosi e al decesso, sopravvivenza media e mediana). La pagina include inoltre una sintesi degli eventi avversi più comuni, riportando il numero totale di eventi registrati, il numero di tipologie distinte e la distribuzione percentuale delle categorie maggiormente rappresentate.

Infine, la sezione prescrittiva propone per i pazienti della coorte possibili raccomandazioni terapeutiche con un relativo punteggio di confidenza. La pagina include una tabella con le raccomandazioni a più alta confidenza per ogni paziente e un grafico che rappresenta la distribuzione complessiva delle tipologie di trattamento suggerite. Questo permette di simulare scenari di supporto decisionale in modo controllato, basandosi sulle informazioni effettivamente disponibili nel knowledge graph.



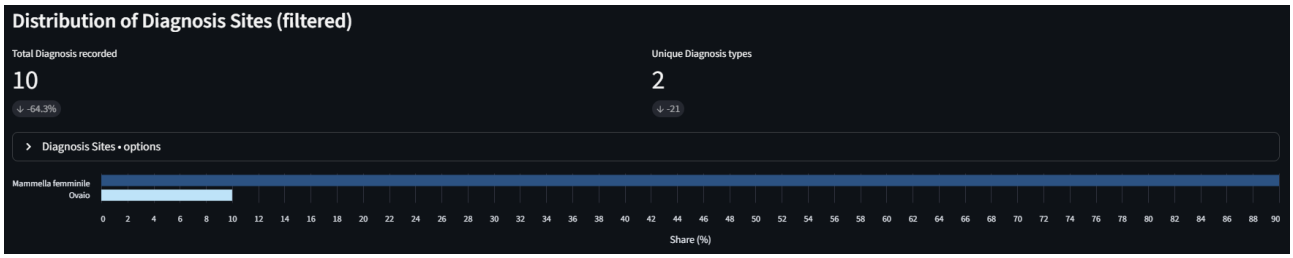


Figura 6.5 - Analisi descrittive della coorte selezionata

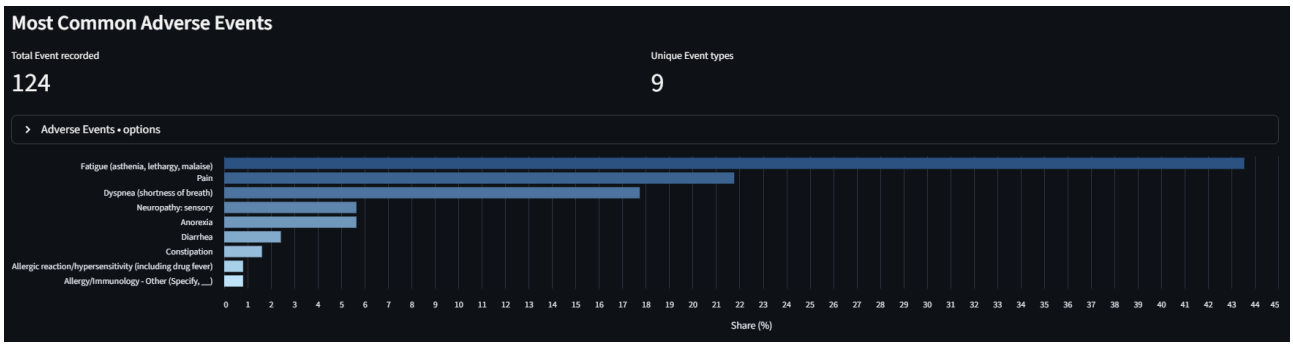


Figura 6.6 - Analisi predittive della coorte selezionata

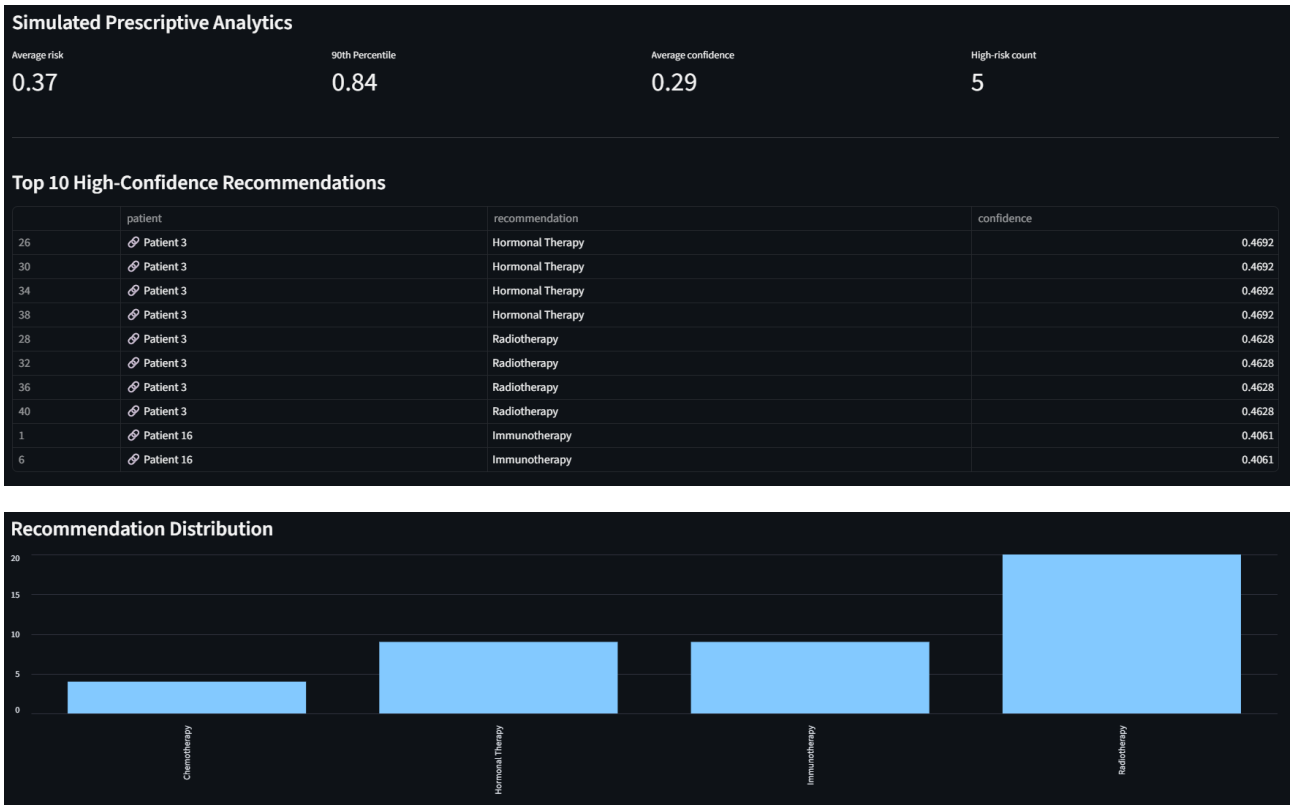


Figura 6.7 - Analisi prescrittive della coorte selezionata

6.4 - Esplorazione del grafo

Il *Graph Explorer* fornisce una rappresentazione grafica dei nodi e delle relazioni del knowledge graph. La visualizzazione non è generica sull'intero repository, ma viene generata a partire da una specifica entità selezionata tramite l'interfaccia Filters (ad es. un paziente, una diagnosi o una terapia). A partire da tale nodo, il sistema costruisce e mostra il sottografo locale, includendo le risorse cliniche collegate rappresentate come nodi e le relative relazioni come archi. Selezionando un nodo, l'utente può visualizzare non solo la tipologia dell'entità, ma anche i valori effettivi delle sue proprietà e proseguire la navigazione verso entità correlate.

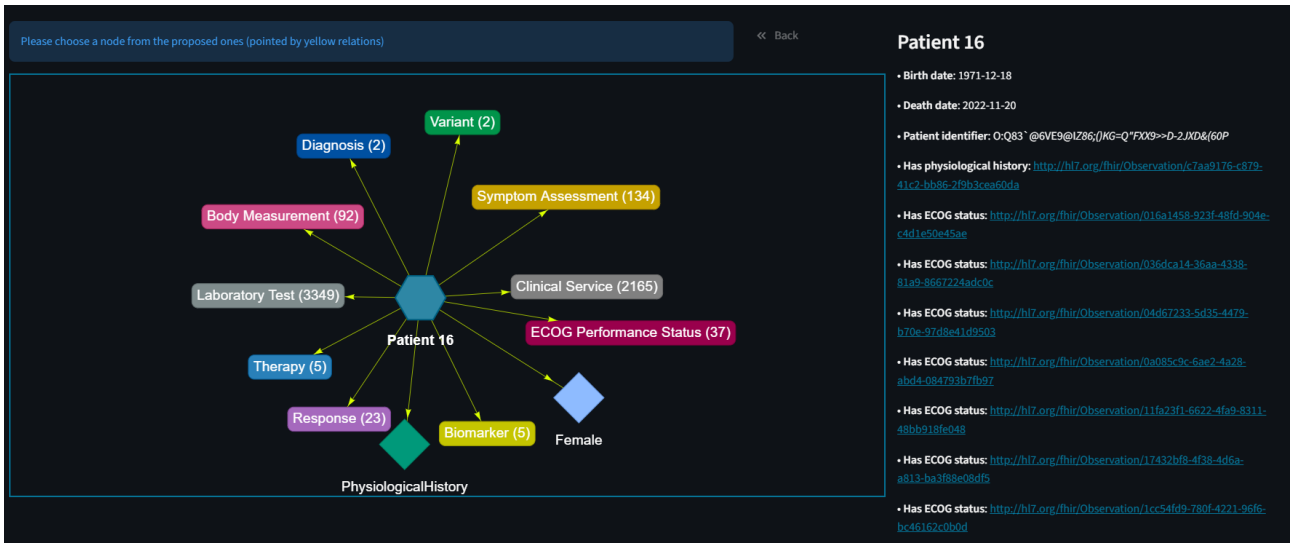


Figura 6.8 - Esplorazione visiva dei nodi e delle relazioni del knowledge graph

6.5 - Esplorazione dell'ontologia

La sezione *Ontology* consente di esplorare direttamente l'ontologia del sistema, visualizzando le classi e le proprietà che definiscono il modello concettuale del knowledge graph. Per ciascuna entità è possibile consultare le proprietà previste dal modello e le relazioni ammissibili con altre classi, ottenendo una descrizione strutturata e verificabile della semantica associata ai nodi del grafo.

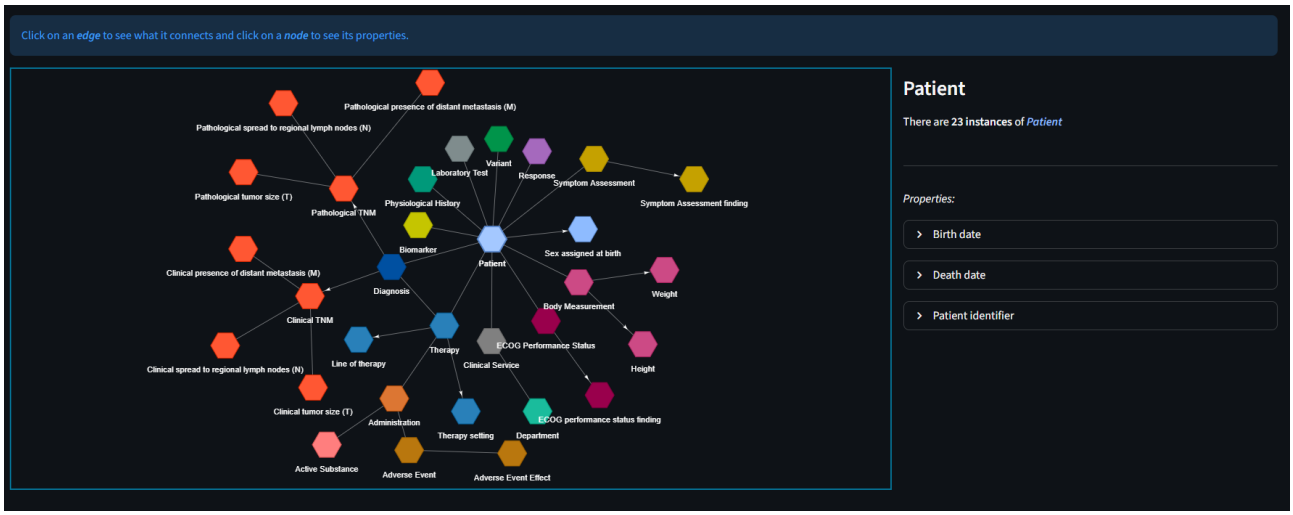


Figura 6.9 - Visualizzazione dell'ontologia CVL

6.6 - Generazione di dati sintetici

La pagina *Synthetic Data Generator* consente di avviare il processo di generazione di dati sintetici a partire da una coorte reale definita semanticamente. Il sistema analizza il sottografo selezionato, ne ricava distribuzioni di frequenza e cardinalità delle relazioni, le traduce in vincoli formali e utilizza tali vincoli per generare nuovi dati RDF coerenti con la struttura del knowledge graph.

In questo modo, l'interfaccia permette di passare direttamente dalla selezione clinica alla simulazione controllata, mantenendo coerenza ontologica e plausibilità clinica.

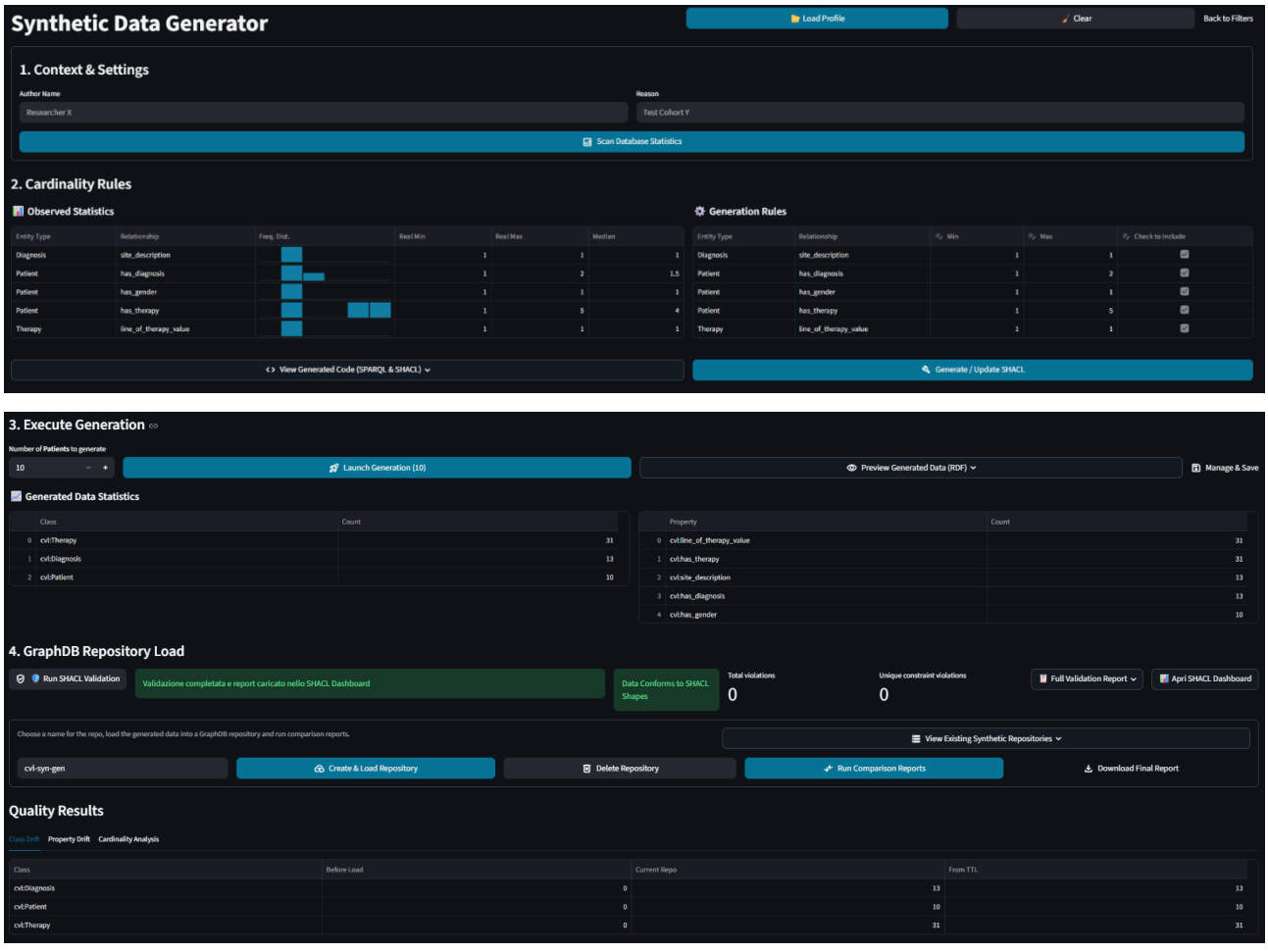


Figura 6.10 - Interfaccia di generazione dei dati sintetici a partire da una coorte

6.7 - Repository Info e monitoraggio della qualità semantica

La sezione *Repository Info* fornisce una vista sintetica sul repository attivo, consentendo di monitorare la qualità e la coerenza del knowledge graph. Le metriche sono organizzate in dimensioni

di accuratezza semantica, consistenza e concisione, che verificano rispettivamente la correttezza delle annotazioni e dei vincoli ontologici, l'assenza di incoerenze logiche e il livello di ridondanza del modello. I punteggi, normalizzati nell'intervallo [0, 1], permettono di valutare rapidamente l'affidabilità del repository prima di procedere con interrogazioni, analisi cliniche o generazione di dati sintetici. Questo modulo consente di valutare lo stato del repository contribuendo a garantire affidabilità e consistenza semantica dell'intero workflow.

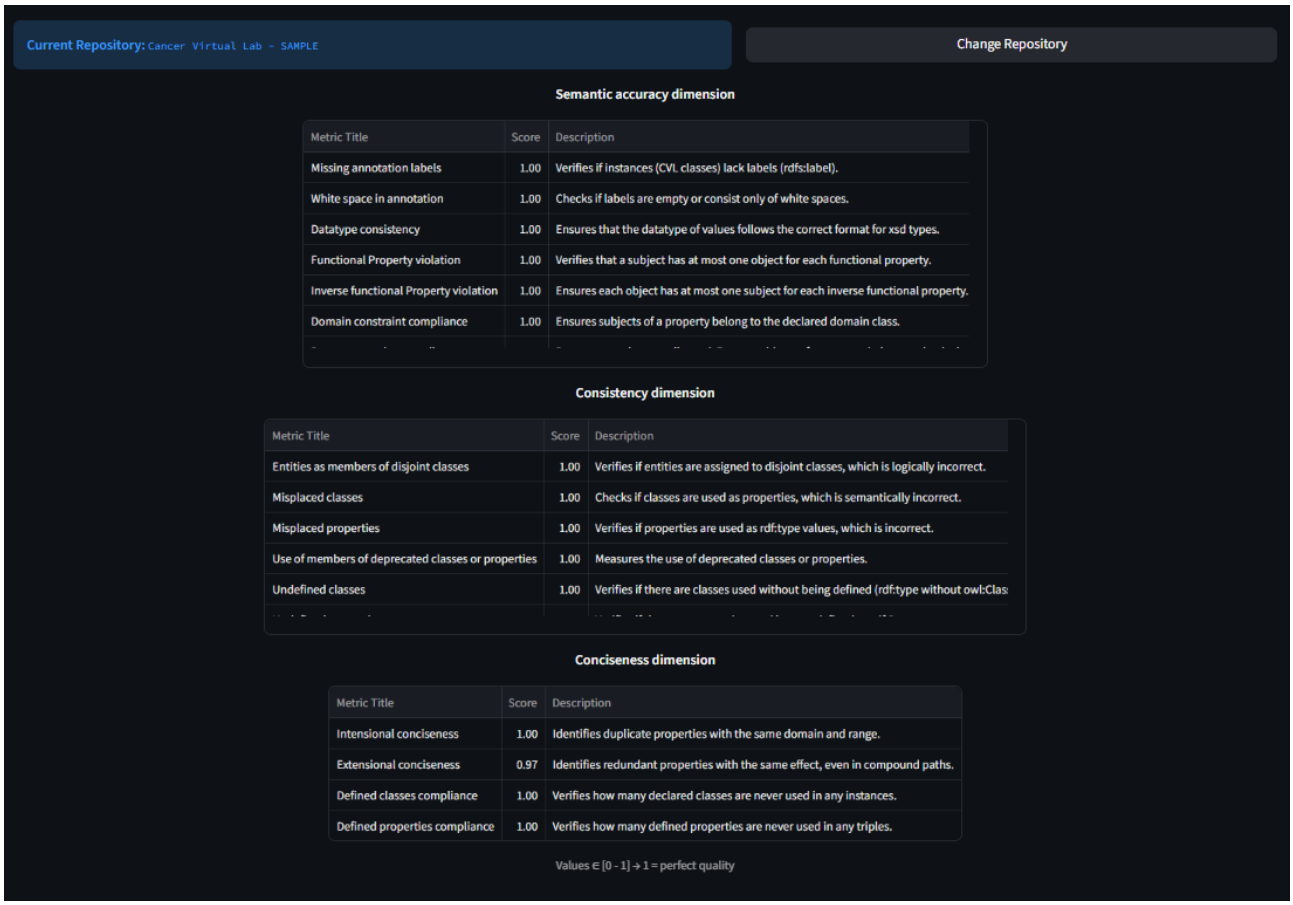


Figura 6.11 - Monitoraggio della qualità del knowledge graph

CAPITOLO 7

GENERAZIONE DATI SINTETICI IN CVL

7.1 - Pipeline di generazione dei dati sintetici

7.1.1 - Acquisizione e raccolta dati

L'estrazione dei dati avviene nell'ambiente istituzionale dell'IRST mediante query SQL eseguite in Qlik Sense, che interrogano i sistemi di cartella clinica elettronica e i registri operativi ospedalieri. I dati acquisiti costituiscono la base informativa primaria del sistema e rappresentano il riferimento per la modellazione semantica e per il processo di generazione dei dati sintetici.

7.1.2 - Pulizia e pre-processing dei dati

I dati estratti vengono sottoposti a un processo strutturato di anonimizzazione, pulizia e normalizzazione. In questa fase vengono eliminati record incompleti, valori inconsistenti o mancanti e formati non validi. La presenza di dati incompleti o parziali è una condizione tipica dei contesti clinici reali, nei quali la raccolta delle informazioni è orientata prioritariamente alla cura del paziente e non alla completezza informativa a fini analitici. Successivamente, i dati vengono mappati in risorse FHIR e sottoposti a validazione semantica rispetto all'ontologia OWL di FHIR. Questo processo garantisce la coerenza strutturale e semantica del dato, rendendolo idoneo all'integrazione nel knowledge graph della piattaforma.

7.1.3 - Analisi e modellazione dei dati

Il processo di generazione dei dati sintetici nel CVL si basa sulla selezione di una coorte di interesse, che costituisce la base da cui derivare l'intero processo generativo. La definizione della coorte non ha solo una funzione esplorativa, ma rappresenta una vera e propria fase di modellazione concettuale: attraverso la selezione dei filtri, l'utente delimita il sottoinsieme informativo rilevante del knowledge graph, individuando le classi, le relazioni e le proprietà che intende riprodurre nella generazione sintetica.

La configurazione dei filtri produce automaticamente una query SPARQL, che interroga il knowledge graph e ne estrae la struttura informativa sottostante. Tale query non viene utilizzata esclusivamente per il recupero dei dati, ma viene reinterpretata come rappresentazione formale della struttura semantica della coorte, dalla quale vengono derivate distribuzioni di frequenza, statistiche descrittive e vincoli SHACL funzionali alla generazione controllata dei dati sintetici.

A seguito della selezione della coorte di interesse, l'utente può visualizzare e analizzare le statistiche della coorte, in particolare la distribuzione di frequenza e i valori di media, minimo e massimo del numero di occorrenze di ciascuna proprietà per singolo soggetto, ossia quante volte una determinata relazione è associata a ciascun individuo.

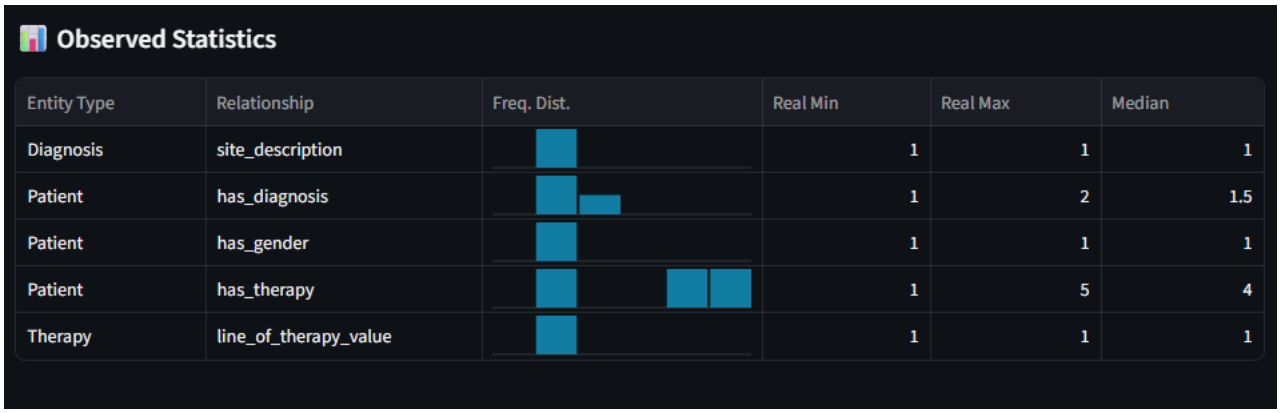


Figura 7.1 - Statistiche della coorte selezionata

Sulla base di tali informazioni, l'utente può intervenire attivamente sulla configurazione del processo generativo, modificando i vincoli e deselezionando le proprietà ritenute non rilevanti.

L'utente mantiene così il pieno controllo sul dato, utilizzando il knowledge graph come base per la generazione ed imponendo vincoli specifici sul processo, in funzione degli obiettivi della simulazione.

7.1.4 - Generazione dei dati sintetici

Un modulo Python di conversione trasforma la query SPARQL, prodotta a partire dalla coorte selezionata e dai vincoli modificati, in un insieme di SHACL shapes che formalizzano vincoli

strutturali, cardinalità, relazioni tra entità, tipi di dato e dipendenze semantiche del dominio selezionato.
Lo SHACL viene generato automaticamente a partire dalla query e dall'estrazione semantica del knowledge graph tramite interrogazioni SPARQL. Sebbene RDFGraphGen sia un framework *domain-agnostic,* nel caso considerato i vincoli SHACL sono derivati dall'ontologia e dai dati del knowledge graph, rendendo la generazione strettamente *domain-specific*. In particolare, i valori ammissibili per nodi e proprietà non vengono introdotti in modo casuale, ma sono ricavati mediante query SPARQL eseguite sulla coorte selezionata, vincolando la generazione ai valori effettivamente osservati nel knowledge graph. Lo SHACL generato costituisce così una specifica generativa del dataset sintetico, non un semplice schema di validazione.

Su questa base, RDFGraphGen produce il dataset RDF/Turtle controllandone la dimensione tramite un parametro quantitativo che definisce il numero di istanze da creare. Il file RDF ottenuto da questo processo viene successivamente sottoposto a una fase di normalizzazione e trasformazione, finalizzata a garantirne la piena compatibilità con il modello ontologico del knowledge graph CVL. Tale fase comprende operazioni di riallineamento dei namespace, rimozione dei metadati non pertinenti, rinomina deterministica degli individui e generazione delle etichette, così da rendere il grafo sintetico pienamente integrabile nell'ecosistema informativo della piattaforma.

Inoltre, poiché RDFGraphGen assegna i valori delle proprietà in modo non distribuzionale, attraverso una generazione non guidata da frequenze statistiche reali, il dataset sintetico prodotto viene sottoposto a una fase di *post-processing* correttivo, nella quale i valori vengono riallineati alle distribuzioni ricavate tramite interrogazioni SPARQL sul knowledge graph di riferimento.

Il dataset sintetico viene quindi caricato in un repository, dove può essere utilizzato come base per attività di test, simulazione, validazione e sperimentazione avanzata.

7.1.5 - Validazione e riproducibilità dei dati sintetici

Come prima forma di controllo, l'utente può visualizzare la query SPARQL generata e il file SHACL derivato dalla query a fini di verifica e debugging del processo generativo. Un'ulteriore validazione è ottenuta sia tramite la visualizzazione diretta del file RDF/Turtle prodotto, sia tramite una vista strutturata che mostra il numero di istanze per classe e di occorrenze di proprietà generate, rendendo verificabile la struttura del dataset sintetico.

Figura 7.2 - Statistiche del dataset sintetico generato

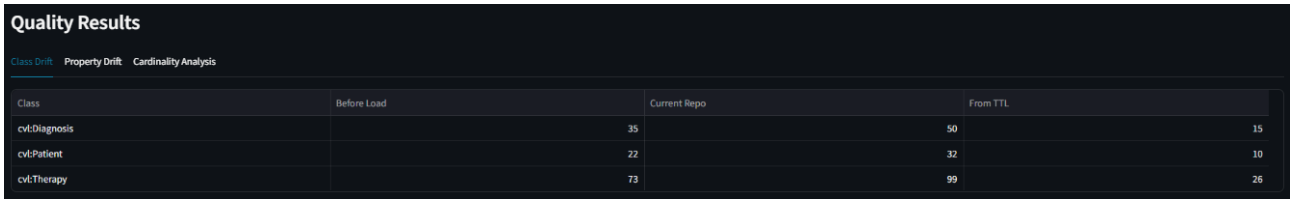
Un successivo controllo è realizzato tramite l'utilizzo di pySHACL, che esegue l'analisi di conformità del grafo sintetico rispetto alle specifiche SHACL generate. Poiché i dati sono generati a partire dalle SHACL shapes, la struttura ontologica del dataset sintetico risulta coerente con quella del modello semantico di riferimento.

L'output della validazione effettuata da pySHACL viene successivamente utilizzato come input per SHACL Dashboard, un tool dedicato alla visualizzazione e all'analisi dei risultati di validazione SHACL. SHACL Dashboard consente di esplorare in modo strutturato i report di validazione, analizzando le violazioni per classe e proprietà e ispezionando i nodi RDF coinvolti, supportando così le attività di verifica qualitativa e di debugging semantico del dataset sintetico.

Figura 7.3 - SHACL Dashboard

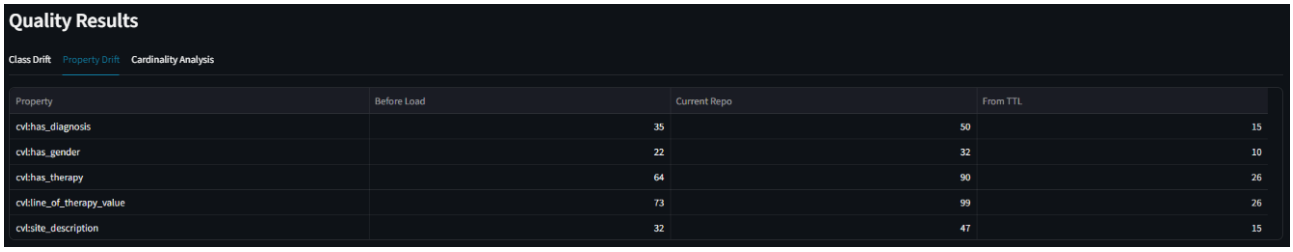
Al caricamento del file in un repository, nuovo o già esistente, vengono generati report di confronto, nei quali sono visualizzate il numero di istanze per ogni classe presenti nel repository prima del

caricamento, quelle presenti dopo il caricamento e quelle introdotte dal file TTL. La stessa analisi viene effettuata per le proprietà.



Class	Before Load	Current Repo	From TTL
cvl:Diagnosis	35	50	15
cvl:Patient	22	32	10
cvl:Therapy	73	99	26

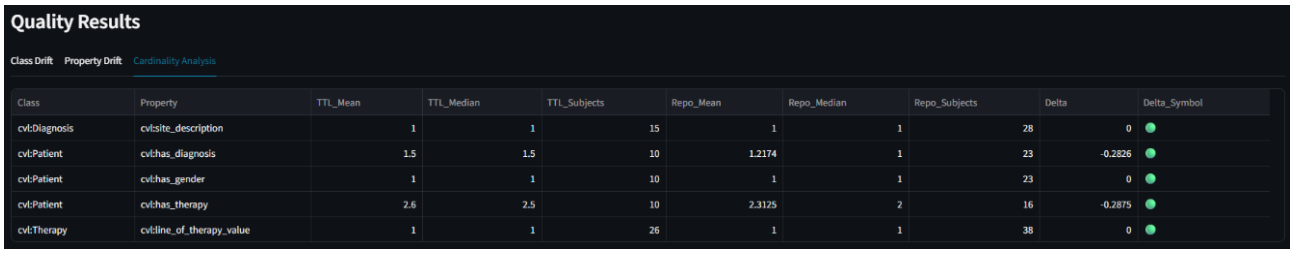
Figura 7.4 - Confronto delle istanze di ciascuna classe pre/post caricamento del dataset sintetico



Property	Before Load	Current Repo	From TTL
cvl:has_diagnosis	35	50	15
cvl:has_gender	22	32	10
cvl:has_therapy	64	90	26
cvl:line_of_therapy_value	73	99	26
cvl:site_description	32	47	15

Figura 7.5 - Confronto occorrenze delle proprietà pre/post caricamento del dataset sintetico

Viene inoltre effettuato, per ciascuna proprietà, un confronto tra i valori di media e mediana del numero di occorrenze per soggetto nel file TTL generato e nel repository di partenza, al fine di verificare la coerenza delle cardinalità tra dataset sintetico e grafo reale.



Class	Property	TTL_Mean	TTL_Median	TTL_Subjects	Repo_Mean	Repo_Median	Repo_Subjects	Delta	Delta_Symbol
cvl:Diagnosis	cvl:site_description	1	1	15	1	1	28	0	
cvl:Patient	cvl:has_diagnosis	1.5	1.5	10	1.2174	1	23	-0.2826	
cvl:Patient	cvl:has_gender	1	1	10	1	1	23	0	
cvl:Patient	cvl:has_therapy	2.6	2.5	10	2.3125	2	16	-0.2875	
cvl:Therapy	cvl:line_of_therapy_value	1	1	26	1	1	38	0	

Figura 7.6 - Confronto delle cardinalità delle proprietà nel file TTL generato e nel repository di origine

Infine, viene generato un report finale che costituisce il documento di tracciabilità e controllo dell'intero processo di generazione dei dati sintetici. Esso registra l'autore, la data e la durata dell'esecuzione, fornisce una sintesi quantitativa del dataset prodotto (numero di istanze per classe),

documenta lo stato dei processi di validazione semantica e verifica la coerenza strutturale delle cardinalità osservate rispetto alle eventuali regole definite. Inoltre, il report specifica il repository di destinazione in cui il dataset sintetico viene caricato e reso disponibile. In questo modo, il report assume valore di audit semantico, assicurando trasparenza, verificabilità e riproducibilità del processo di generazione.

La riproducibilità è garantita anche dalla possibilità di salvare la configurazione (coorte selezionata, vincoli, configurazioni), all'interno di profili persistenti, ricaricabili successivamente per ottenere gli stessi risultati. Vengono inoltre sempre salvati la query SPARQL di origine, il file SHACL, il file RDF/Turtle prodotto e il report finale, garantendo la completa tracciabilità del processo.

Per quanto riguarda la validazione della privacy, aspetto fondamentale nei dati clinici, essa risulta supportata dal fatto che i dati di origine sono anonimizzati e la generazione dei valori avviene in modo casuale tramite RDFGraphGen. Proprio per la natura casuale della generazione, l'approccio non preserva in modo nativo le correlazioni tra i dati e, a differenza dei modelli generativi basati su machine learning, non prevede una fase di addestramento sul dataset originario. Per questi motivi, il rischio di re-identificazione e di replicazione di record reali risulta significativamente ridotto.

7.2 - Esempio applicativo: generazione di un dataset sintetico a partire da una coorte

Di seguito è presentato un esempio di esecuzione del processo di generazione dei dati sintetici all'interno della piattaforma, a partire dalla selezione di una coorte di interesse tramite la pagina di interfaccia *Filters*. Viene illustrato l'intero flusso operativo, che dalla definizione della coorte conduce alla generazione delle SHACL shapes, fino alla produzione, validazione e tracciabilità del dataset sintetico.

7.2.1 - Definizione della coorte e selezione tramite interfaccia Filters

La coorte di riferimento è costituita da pazienti di genere femminile con diagnosi oncologica a sede "mammella femminile" o "ovaio", associate a percorsi terapeutici in qualunque linea di trattamento. Tale coorte è stata selezionata come caso esemplificativo per la dimostrazione *end-to-end* del processo di generazione dei dati sintetici.

Nella pagina *Filters*, a seguito della selezione dei filtri, viene visualizzata una tabella contenente l'intero insieme dei risultati della query, ovvero tutti i pazienti che presentano le caratteristiche selezionate.

Attraverso il comando *Configure Synthetic Gen*, l'utente accede quindi alla pagina dedicata alla generazione vera e propria dei dati sintetici.

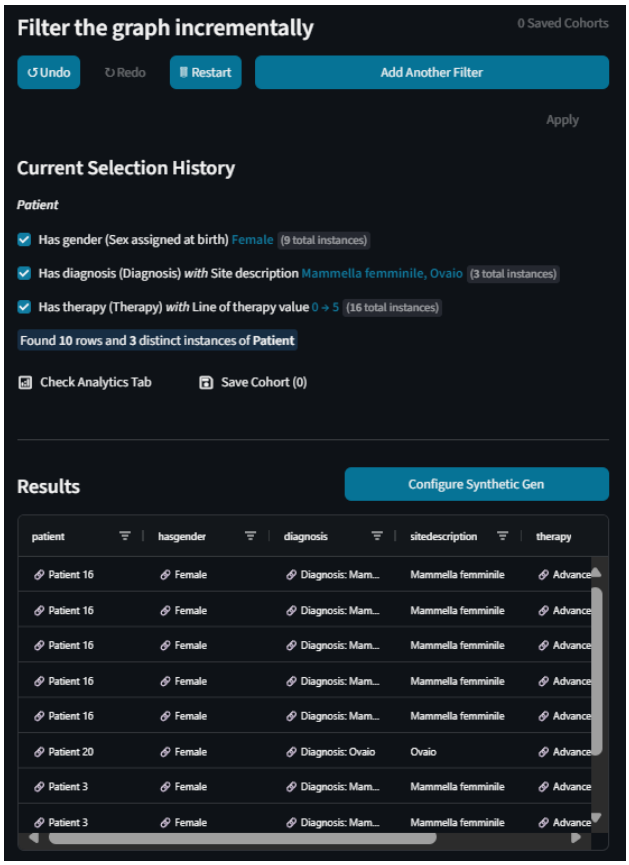


Figura 7.7 - Pagina di selezione dei filtri

7.2.2 - Configurazione del processo di generazione dei dati sintetici

Nella pagina di generazione dei dati sintetici, l'utente può caricare un profilo di configurazione precedentemente salvato tramite l'opzione *Load Profile*, oppure procedere alla costruzione incrementale di una nuova configurazione a partire dalla coorte selezionata. I campi *author name* e *reason* sono dedicati alla tracciabilità del processo. Attraverso l'opzione *scan database statistics*, la piattaforma restituisce le statistiche descrittive della coorte selezionata, includendo distribuzioni di frequenza, minimo, massimo e mediana del numero di occorrenze per ciascuna proprietà della coorte.

Su questa base informativa, l'utente può intervenire attivamente sul processo generativo, modificando i vincoli di cardinalità (min e max) e deselezionando specifiche relazioni, che verranno conseguentemente rimosse dalla generazione del dataset sintetico.

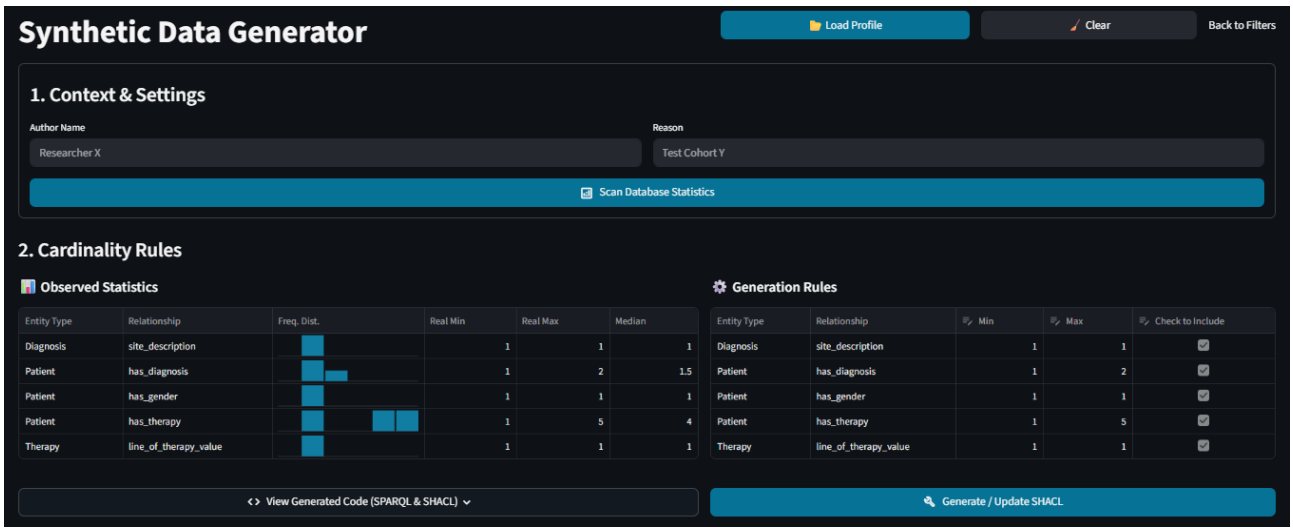


Figura 7.8 - Visualizzazione statistiche e modifica dei vincoli

7.2.3 - Generazione delle SHACL shapes a partire dalla coorte

La pressione del comando *Generate/Update SHACL* determina la generazione automatica delle SHACL shapes a partire dalla query SPARQL e dai dati estratti direttamente dal grafo tramite la stessa. Le shapes ottenute fungono da specifica semantica per la generazione del dataset sintetico. La query SPARQL di origine e il file SHACL derivato sono visualizzabili a fini di ispezione e debug.

7.2.4 - Generazione del dataset sintetico RDF

Nella fase di esecuzione del processo generativo, l'utente imposta il parametro quantitativo di generazione, in questo caso il numero di istanze di pazienti da creare, che funge da fattore di scala dell'intero dataset sintetico. L'avvio del processo tramite il comando *Launch Generation* attiva l'esecuzione di RDFGraphGen sulla base delle SHACL shapes generate, producendo il grafo RDF sintetico.

La piattaforma consente la visualizzazione del file RDF/Turtle generato e fornisce una sintesi strutturata dei risultati della generazione mostrando il numero di istanze create per ciascuna classe e

il numero di occorrenze generate per ciascuna proprietà. Nel caso d'uso analizzato, risultano generati 10 pazienti, 26 terapie e 15 diagnosi, con la creazione di 26 relazioni cvl:line_of_therapy_value, 26 cvl:has_therapy, 15 cvl:site_description, 15 cvl:has_diagnosis e 10 cvl:has_gender. Tali informazioni consentono una prima verifica quantitativa della coerenza del dataset sintetico rispetto alla configurazione definita.

Figura 7.9 - Generazione dei dati sintetici

7.2.5 - Validazione SHACL e integrazione nel repository GraphDB

Prima dell'integrazione, il dataset sintetico viene sottoposto a una fase di validazione SHACL, finalizzata a verificarne la conformità alle specifiche semantiche e strutturali del modello. Cliccando su *SHACL Dashboard*, l'utente viene reindirizzato alla dashboard dedicata, dalla quale è possibile consultare in modo interattivo i risultati della validazione.

Successivamente, il grafo validato può essere caricato in un repository GraphDB, selezionando un repository esistente dall'elenco disponibile oppure creandone uno nuovo tramite inserimento diretto del nome. Il caricamento è accompagnato dalla generazione automatica di report di confronto (*comparison reports*) e di un report finale di processo, che documenta l'intero workflow generativo, assicurando tracciabilità e riproducibilità dell'operazione.

Attraverso il report di confronto è possibile visualizzare, per ciascuna classe, il numero di istanze già presenti nel repository di partenza (nel caso specifico 35 diagnosi, 22 pazienti e 73 terapie), il numero di istanze presenti dopo il caricamento del file (50 diagnosi, 32 pazienti e 99 terapie) e, di conseguenza, le istanze effettivamente importate con il file RDF/Turtle generato. Un'analoga analisi viene effettuata per le proprietà. Per ciascuna proprietà vengono inoltre calcolati e confrontati i valori di media e mediana del numero di occorrenze di proprietà per soggetto nel file TTL generato e nel

repository di riferimento; il valore di delta rappresenta la differenza tra le medie, consentendo una verifica della coerenza delle cardinalità tra grafo sintetico e grafo reale.

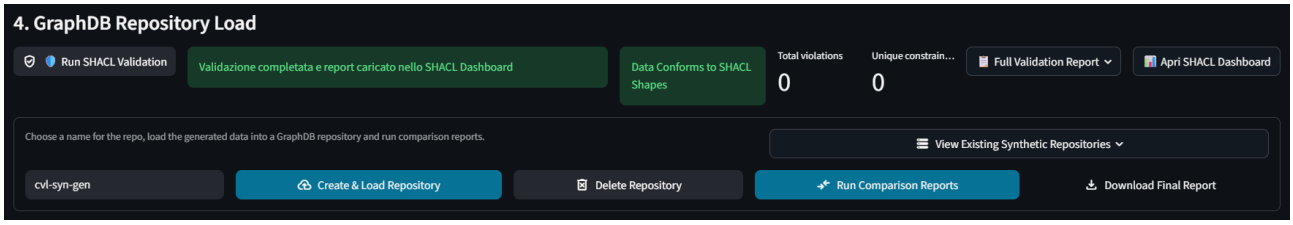


Figura 7.10 - Validazione SHACL e caricamento dei dati sintetici in un repository GraphDB

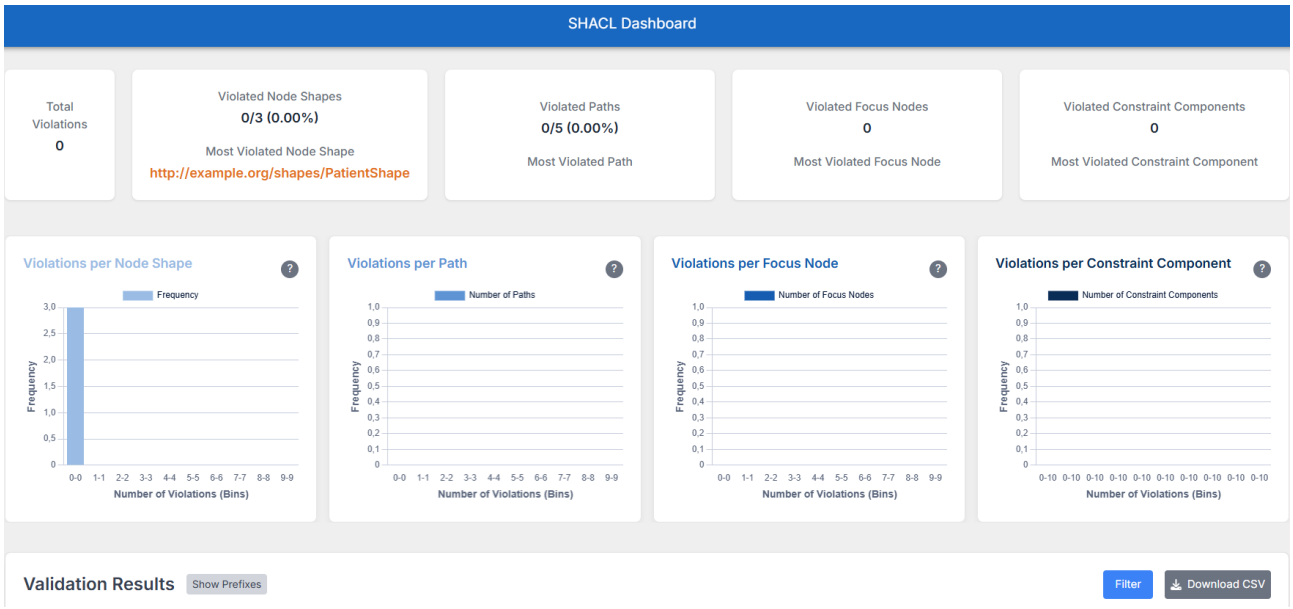


Figura 7.11 - SHACL Dashboard

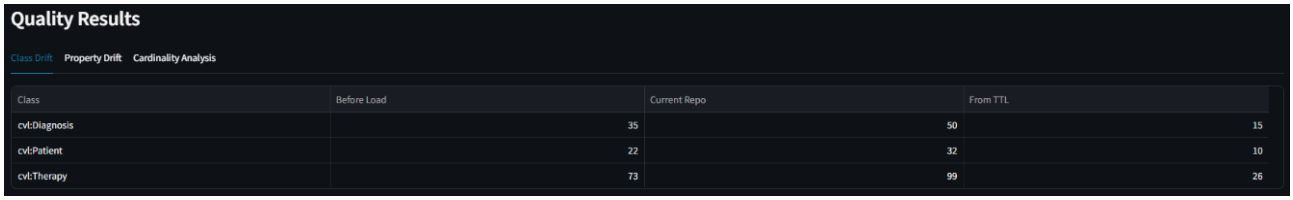
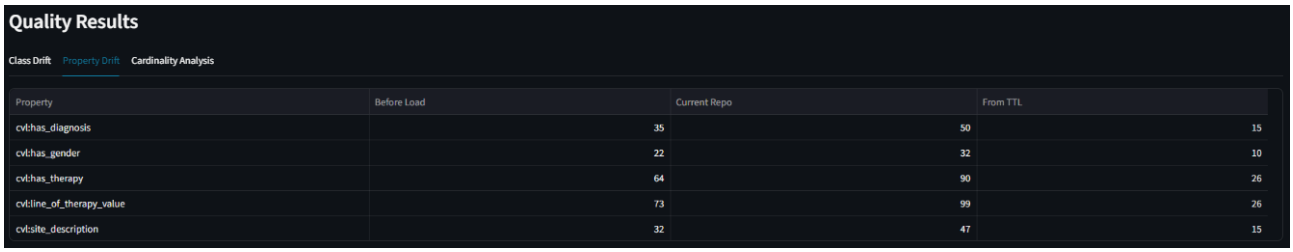
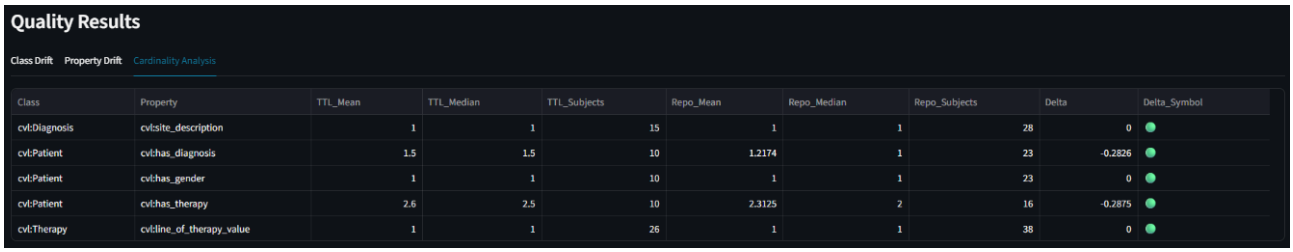


Figura 7.12 - Analisi comparativa della numerosità delle istanze di classe nel repository GraphDB prima e dopo il caricamento



Property	Before Load	Current Repo	From TTL
cvl:has_diagnosis	35	50	15
cvl:has_gender	22	32	10
cvl:has_therapy	64	90	26
cvl:line_of_therapy_value	73	99	26
cvl:site_description	32	47	15

Figura 7.13 - Analisi comparativa del numero di occorrenze delle proprietà nel repository GraphDB prima e dopo il caricamento dei dati



Class	Property	TTL_Mean	TTL_Median	TTL_Subjects	Repo_Mean	Repo_Median	Repo_Subjects	Delta	Delta_Symbol
cvl:Diagnosis	cvl:site_description	1	1	15	1	1	28	0	
cvl:Patient	cvl:has_diagnosis	1.5	1.5	10	1.2174	1	23	-0.2826	
cvl:Patient	cvl:has_gender	1	1	10	1	1	23	0	
cvl:Patient	cvl:has_therapy	2.6	2.5	10	2.3125	2	16	-0.2875	
cvl:Therapy	cvl:line_of_therapy_value	1	1	26	1	1	38	0	

Figura 7.14 - Valutazione del numero di occorrenze delle proprietà mediante confronto statistico tra dataset sintetico e dataset originale

7.2.6 - Output del processo e tracciabilità

A chiusura del caso d'uso vengono mostrati i file prodotti dal processo di generazione, dalla query SPARQL rappresentativa della coorte di interesse, alle SHACL shapes, fino all'output RDF/Turtle del dataset sintetico. Questa sezione rende esplicita la sequenza logica e operativa che conduce dalla definizione semantica della coorte alla produzione dei dati sintetici.

```
SELECT ?patient ?label_patient ?hasgender ?label_hasgender ?diagnosis ?label_diagnosis ?sitedescription ?therapy ?label_therapy ?lineoftherapyvalue WHERE {
?patient a <http://irst.emr.it/cvl/Patient> .
OPTIONAL { ?patient rdfs:label ?label_patient . FILTER (lang(?label_patient) = "en" || lang(?label_patient) = "") }
?patient <http://irst.emr.it/cvl/has_gender> ?hasgender .
FILTER(?hasgender IN (<http://irst.emr.it/cvl/Female>))
OPTIONAL { ?hasgender rdfs:label ?label_hasgender . FILTER (lang(?label_hasgender) = "en" || lang(?label_hasgender) = "") }
?patient <http://irst.emr.it/cvl/has_diagnosis> ?diagnosis .
?diagnosis a <http://irst.emr.it/cvl/Diagnosis> .
OPTIONAL { ?diagnosis rdfs:label ?label_diagnosis . FILTER (lang(?label_diagnosis) = "en" || lang(?label_diagnosis) = "") }
?diagnosis <http://irst.emr.it/cvl/site_description> ?sitedescription .
FILTER(?sitedescription IN ("Mammella femminile", "Ovaio"))
?patient <http://irst.emr.it/cvl/has_therapy> ?therapy .
?therapy a <http://irst.emr.it/cvl/Therapy> .
OPTIONAL { ?therapy rdfs:label ?label_therapy . FILTER (lang(?label_therapy) = "en" || lang(?label_therapy) = "") }
?therapy <http://irst.emr.it/cvl/line_of_therapy_value> ?lineoftherapyvalue .
FILTER(?lineoftherapyvalue >= 0 && ?lineoftherapyvalue <= 5)
}
```

Figura 7.15 - Query SPARQL rappresentativa della coorte di pazienti di genere femminile con diagnosi a sede "mammella femminile" o "ovaio", associate a percorsi terapeutici in qualsiasi linea di trattamento

```
ex:DiagnosisShape
    a sh:NodeShape ;
    sh:targetClass <http://irst.emr.it/cvl/Diagnosis> ;

    sh:property [
        sh:path <http://irst.emr.it/cvl/site_description> ;
        sh:minCount 1 ;
        sh:maxCount 1 ;
        sh:in ( "Mammella femminile" "Ovaio" ) ;
    ] .

ex:PatientShape
    a sh:NodeShape ;
    sh:targetClass <http://irst.emr.it/cvl/Patient> ;

    sh:property [
        sh:path <http://irst.emr.it/cvl/has_diagnosis> ;
        sh:minCount 1 ;
        sh:maxCount 2 ;
        sh:node ex:DiagnosisShape ;
    ] ;

    sh:property [
        sh:path <http://irst.emr.it/cvl/has_gender> ;
        sh:minCount 1 ;
        sh:maxCount 1 ;
        sh:in ( <http://irst.emr.it/cvl/Female> ) ;
    ] ;

    sh:property [
        sh:path <http://irst.emr.it/cvl/has_therapy> ;
        sh:minCount 1 ;
        sh:maxCount 5 ;
        sh:node ex:TherapyShape ;
    ] .

ex:TherapyShape
    a sh:NodeShape ;
    sh:targetClass <http://irst.emr.it/cvl/Therapy> ;

    sh:property [
        sh:path <http://irst.emr.it/cvl/line_of_therapy_value> ;
        sh:minCount 1 ;
        sh:maxCount 1 ;
        sh:in ( 0 1 2 3 4 5 ) ;
    ] .
```

Figura 7.16 - SHACL shapes generate a partire dalla query SPARQL

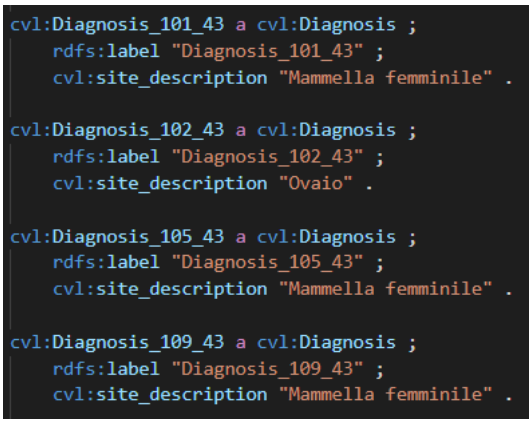
```
@prefix cvl: <http://irst.emr.it/cvl/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

cvl:Patient_100_43 a cvl:Patient ;
    rdfs:label "Patient_100_43" ;
    cvl:has_diagnosis cvl:Diagnosis_101_43,
        cvl:Diagnosis_102_43 ;
    cvl:has_gender cvl:Female ;
    cvl:has_therapy cvl:Therapy_103_43 .

cvl:Patient_104_43 a cvl:Patient ;
    rdfs:label "Patient_104_43" ;
    cvl:has_diagnosis cvl:Diagnosis_105_43 ;
    cvl:has_gender cvl:Female ;
    cvl:has_therapy cvl:Therapy_106_43,
        cvl:Therapy_107_43 .

cvl:Patient_108_43 a cvl:Patient ;
    rdfs:label "Patient_108_43" ;
    cvl:has_diagnosis cvl:Diagnosis_109_43,
        cvl:Diagnosis_110_43 ;
    cvl:has_gender cvl:Female ;
    cvl:has_therapy cvl:Therapy_111_43,
        cvl:Therapy_112_43,
        cvl:Therapy_113_43 .
```

Figura 7.17 - Estratto RDF/Turtle dei pazienti sintetici con rispettive relazioni



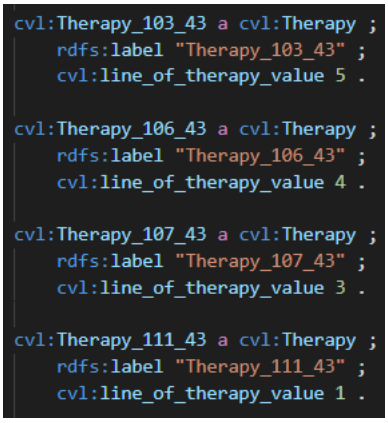
```
cvl:Diagnosis_101_43 a cvl:Diagnosis ;
    rdfs:label "Diagnosis_101_43" ;
    cvl:site_description "Mammella femminile" .

cvl:Diagnosis_102_43 a cvl:Diagnosis ;
    rdfs:label "Diagnosis_102_43" ;
    cvl:site_description "Ovaio" .

cvl:Diagnosis_105_43 a cvl:Diagnosis ;
    rdfs:label "Diagnosis_105_43" ;
    cvl:site_description "Mammella femminile" .

cvl:Diagnosis_109_43 a cvl:Diagnosis ;
    rdfs:label "Diagnosis_109_43" ;
    cvl:site_description "Mammella femminile" .
```

Figura 7.18 - Estratto RDF/Turtle delle diagnosi sintetiche con sede oncologica associata



```
cvl:Therapy_103_43 a cvl:Therapy ;
    rdfs:label "Therapy_103_43" ;
    cvl:line_of_therapy_value 5 .

cvl:Therapy_106_43 a cvl:Therapy ;
    rdfs:label "Therapy_106_43" ;
    cvl:line_of_therapy_value 4 .

cvl:Therapy_107_43 a cvl:Therapy ;
    rdfs:label "Therapy_107_43" ;
    cvl:line_of_therapy_value 3 .

cvl:Therapy_111_43 a cvl:Therapy ;
    rdfs:label "Therapy_111_43" ;
    cvl:line_of_therapy_value 1 .
```

Figura 7.19 - Estratto RDF/Turtle delle terapie sintetiche con linea di trattamento associata

CONCLUSIONI E SVILUPPI FUTURI

Il lavoro di tesi ha portato alla progettazione e all'implementazione di un processo di generazione di dati sintetici integrato all'interno della piattaforma Cancer Virtual Lab. La piattaforma dispone ora di una sezione operativa per la generazione, la validazione e il salvataggio di dataset sintetici in repository dedicati, rendendoli immediatamente riutilizzabili nel flusso applicativo complessivo. I dataset sintetici, generati a partire dai dati reali, risultano interrogabili e concretamente utilizzabili da clinici e ricercatori per attività di ricerca e per lo studio dell'evoluzione dei percorsi assistenziali. L'integrazione con le funzionalità della piattaforma consente infatti di accedere ai dati sintetici tramite strumenti di interrogazione avanzata, inclusa l'interazione con un agente di intelligenza artificiale, e di supportare analisi descrittive, diagnostiche, predittive e prescrittive delle coorti di pazienti.

Per quanto riguarda gli sviluppi futuri, una direzione di evoluzione fondamentale riguarda l'introduzione di meccanismi di coerenza clinica nella generazione dei dati, superando l'attuale assegnazione prevalentemente casuale dei valori, pur compatibili con i vincoli strutturali e semantici del modello. A titolo esemplificativo, sarà necessario garantire che a una determinata diagnosi corrispondano terapie e farmaci clinicamente appropriati e coerenti con la patologia di riferimento e che, analogamente, un paziente in uno specifico stadio di malattia venga associato a trattamenti adeguati alla gravità della condizione. Tale esigenza risulta strettamente connessa al tema della correlazione tra variabili cliniche, elemento essenziale affinché il dato sintetico sia non solo formalmente valido, ma anche plausibile dal punto di vista medico.

Un ulteriore sviluppo rilevante potrebbe consistere nell'adozione di generatori alternativi rispetto a RDFGraphGen, basati su modelli generativi in grado di preservare in modo più fedele la distribuzione statistica dei dati reali. In tale prospettiva, l'integrazione di approcci ispirati a modelli come OntoCGAN potrebbe consentire di ottenere dataset che riproducano non soltanto i vincoli strutturali, ma anche i pattern quantitativi osservabili nei dati originali.

Parallelamente, risulta necessario introdurre un livello di validazione più avanzato, finalizzato a misurare in modo oggettivo la qualità del dataset sintetico. In particolare, potranno essere integrate query SPARQL dedicate alla valutazione dell'utility, verificando la capacità dei dati generati di supportare interrogazioni e analisi analoghe a quelle effettuabili su dati reali. Sul piano della fidelity, invece, potranno essere applicate analisi statistiche mirate a confrontare la distribuzione e la correlazione dei dati, al fine di individuare eventuali deviazioni significative e garantire una maggiore aderenza ai comportamenti osservabili nel dataset di riferimento.

Nel complesso, tali sviluppi consentiranno di rendere la generazione dei dati sintetici non soltanto strutturalmente e semanticamente corretta, ma anche clinicamente plausibile, incrementando in modo significativo il valore applicativo della piattaforma per la ricerca e per il supporto alle decisioni cliniche.

BIBLIOGRAFIA

[1] H. Knublauch and D. Kontokostas, “Shapes Constraint Language (SHACL),” W3C Recommendation, 2017.

[2] M. Jovanovik, M. Vecovska, M. Jakubowski, and K. Hose, “RDFGraphGen: An RDF graph generator based on SHACL shapes,” 2025.

[3] C. Sun and M. Dumontier, “Generating unseen diseases patient data using ontology enhanced generative adversarial networks,” 2025.

[4] Y. Zhu, Y. Du, Y. Wang, Y. Xu, J. Zhang, Q. Liu, and S. Wu, “A survey on graph generation: Methods and application,” 2022.

[5] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, and S. Auer, “Quality assessment for linked data: A survey,” 2016.

[6] M. Färber, F. Bartscherer, C. Menne, and A. Rettinger, “Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO,” 2018.

[7] D. Kontokostas, P. Westphal, S. Auer, S. Hellmann, J. Lehmann, R. Cornelissen, and A. Zaveri, “Test-driven evaluation of linked data quality,” 2014.

[8] J. E. Labra Gayo, E. Prud’hommeaux, I. Boneva, and D. Kontokostas, “Validating RDF Data,” Morgan & Claypool Publishers, 2017.

[9] G. Aluç, O. Hartig, M. T. Özsu, and K. Daudjee, “Diversified stress testing of RDF data management systems,” 2014.

[10] C. Bizer and A. Schultz, “The Berlin SPARQL benchmark,” 2009.

[11] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava, “Resisting structural re-identification in anonymized social networks,” 2016.

[12] R. Gonçalves, P. Ray, B. Soper, J. Stevens, L. Coyle, and A. Sales, “Generation and evaluation of synthetic patient data,” 2020.

[13] A. Carbonaro, L. Giorgetti, A. Pagliarani, and N. Gentili, “Operationalizing semantic access to clinical knowledge with CVL-KG: A multi-layered framework for interoperability and decision support,” 2025.

[14] A. Carbonaro, L. Giorgetti, L. Ridolfi, R. Pasolini, A. Pagliarani, M. Cavallucci, A. Andalò, L. Del Gaudio, P. De Angelis, R. Vespignani, and N. Gentili, “From raw data to research-ready: A FHIR-based transformation pipeline in a real-world oncology setting,” 2025.