



**Politecnico
di Torino**

Politecnico di Torino

Corso di Laurea Magistrale in Ingegneria Elettrica

A.a. 2023/2024

Sessione di Laurea Novembre/Dicembre 2024

**L'Applicazione del BIM nella Gestione del
sistema di messa a terra
e del conduttore di protezione (PE)
della Torre della Regione Piemonte**

Relatori:

Prof. Ing. Di Leo Paolo

Prof. Ing. Del Giudice Matteo

Prof. Ing. Zucco Michele

Ing. Fratto Andrea

Candidato:

Bertone Alessandro

ABSTRACT (Italiano)

Questa tesi esplora l'applicazione del Building Information Modeling (BIM) nella gestione e manutenzione del sistema di protezione della Torre della Regione Piemonte a Torino, un grattacielo inaugurato nel 2022 e situato nella zona Lingotto a Torino. Con particolare attenzione al sistema TN dell'edificio, lo studio analizza sia l'impianto di messa a terra che il conduttore di protezione (PE), elementi fondamentali per la sicurezza elettrica del grattacielo. Attraverso un modello "As Built" sviluppato in Autodesk Revit, il lavoro comprende una rassegna degli standard BIM e delle normative sugli impianti elettrici, seguita da un'analisi approfondita della metodologia di modellazione dei componenti e dell'implementazione di una struttura gerarchica per la gestione delle informazioni. Di particolare rilevanza è lo sviluppo di uno script Dynamo per il calcolo automatizzato dell'impedenza dell'anello di guasto, che permette di analizzare il percorso completo dai dispositivi fino ai quadri a monte, considerando sia il circuito di alimentazione che quello di protezione. La tesi esamina inoltre l'integrazione del BIM nel Facility Management, evidenziando i benefici per la manutenzione predittiva e l'efficienza operativa. Il lavoro si conclude con l'implementazione di soluzioni di realtà virtuale e aumentata per la visualizzazione dell'intero sistema di protezione, aprendo nuove prospettive per il monitoraggio e la manutenzione delle infrastrutture elettriche.

ABSTRACT (English)

This thesis explores the application of Building Information Modeling (BIM) in the management and maintenance of the protection system of the Piedmont Region Tower in Turin, a skyscraper inaugurated in 2022 and located in the Lingotto area of Turin. With a particular focus on the building's TN system, the study analyzes both the grounding system and the protective conductor (PE), essential elements for the electrical safety of the skyscraper. Through an "As Built" model developed in Autodesk Revit, the work includes a review of BIM standards and electrical system regulations, followed by an in-depth analysis of the component modeling methodology and the implementation of a hierarchical structure for information management. Particularly noteworthy is the development of a Dynamo script for the automated calculation of fault loop impedance, which allows a complete analysis of the pathway from devices to upstream panels, taking into account both the power and protection circuits. The thesis also examines BIM integration in Facility Management, highlighting the benefits for predictive maintenance and operational efficiency. The work concludes with the implementation of virtual and augmented reality solutions for the visualization of the entire protection system, opening new perspectives for the monitoring and maintenance of electrical infrastructures.

Indice

Indice delle figure.....	9
1. BIM: Building Information Modeling.....	13
1.1 Definizione.....	13
1.2 Leggi e normative.....	17
1.2.1 Anni 2000.....	17
1.2.2 Anni 2010.....	17
1.2.3 Anni 2014 e seguenti.....	17
1.3 La torre della Regione Piemonte.....	23
1.3.1 Sistema di alimentazione del complesso.....	25
2. Impianto di terra.....	28
2.1 Cenni teorici e sicurezza elettrica.....	28
2.1.1 Progettazione secondo la norma.....	31
2.1.2 Misura dell'anello di guasto.....	35
2.2 Il caso specifico della torre.....	38
3. Metodologia "As Built".....	44
3.1 Piattaforma di sviluppo.....	44
3.2 Modellazione componenti.....	45
3.2.1 Maglia di terra.....	48
3.2.2 Picchetto.....	49
3.2.3 Morsetto di interconnessione.....	50
3.2.4 Collettori di terra.....	51
3.2.5 Quadro elettrico.....	53
3.2.6 Blindosbarre, torrette e corpi illuminanti.....	55
3.3 Modellazione sistemi elettrici.....	55
3.3.1 Browser dei sistemi.....	55
3.3.2 Parametro di appartenenza impianto di terra.....	59
3.3.3 Parametro circuito di terra.....	60
3.3.4 Filtri di vista.....	61
3.3.5 Abachi.....	63
4. Gestione e manutenzione.....	64
4.1 Generalità.....	64
4.2 Verifica degli impianti di terra.....	65
4.3 Esempio applicativo di calcolo dell'impedenza dell'anello di guasto.....	69
4.4 Risultati numerici.....	83
5. Visualizzazione.....	86

5.1 Esportazione da Revit	86
5.2 Importazione parametri da Revit.....	89
5.2.1 Codice LettoreCSV.cs.....	89
5.2.2 Codice KeyValuePair.cs	90
5.2.3 Codice DynamicParameters.cs.....	91
5.2.4 Codice DynamicParametersEditor.cs.....	92
5.2.5 Codice ColliderAssigner.cs.....	93
5.2.6 Codice CameraController.cs	94
5.2.7 Codice TabletUI.cs.....	94
5.2.8 Codice ObjectClickHandler.cs.....	95
5.3 Visualizzazione in realtà virtuale	97
6. Conclusione e sviluppi futuri	99
Appendice 1	101
Appendice 2	108
Appendice 3 (script lettura circuiti)	110
Appendice 4 (script rilevazione lunghezza).....	112
Appendice 5 (script rilevazione nome quadro).....	113
Appendice 6 (script dizionario sezione).....	114
Appendice 7 (script calcolo impedenza).....	116
Appendice 8	117
Appendice 9	119
Appendice 10	121
Appendice 11	124
Appendice 12	125
Appendice 13	126
Appendice 14	127
Appendice 15	129
Bibliografia e sitografia	132

Indice delle figure

Figura 1: ciclo di vita degli edifici	13
Figura 2: Curva di MacLeamy	14
Figura 3: schema concettuale formato IFC	16
Figura 4: schema requisiti informativi secondo la ISO 19650.....	19
Figura 5: livelli di maturità BIM secondo la UNI 11337-1.....	20
Figura 6: Dimensioni del BIM	22
Figura 7: esterno della Torre (scatto personale del 29/04/2024).....	23
Figura 8: locazione cabina di ricezione MT	25
Figura 9: schema di massima rete di distribuzione (realizzazione personale).....	26
Figura 10: tensioni di contatto e di passo [21]	29
Figura 11: tipi di resistenza e relativo potenziale [30].....	30
Figura 12: tensione di contatto limite (https://www.overtec.it/articoli/impianti-elettrici/sostituita-la-norma-cei-11-1-dalle-en-61636-1-ed-en-50522).....	30
Figura 13: schema ridotto verifica impianti di terra [30].....	32
Figura 14: schema completo verifica impianti di terra [21].....	33
Figura 15: schema del percorso della corrente di guasto [21].....	34
Figura 16: loop tester https://www.asita.com/prodotto/strumento-integrato-prova-differenziali-e-loop-test-as5025/	36
Figura 17: diagramma vettoriale differenza delle tensioni [31].....	37
Figura 18: schema impianto di terra del complesso "Torre" (realizzazione personale).....	38
Figura 19: tipi di morsetti di interconnessione.....	39
Figura 20: corda di rame	39
Figura 21: picchetto.....	40
Figura 22: pozzetto in calcestruzzo	40
Figura 23: collettore cabina MT (scatto personale del 29/04/2024)	40
Figura 24: collettore locale BT (scatto personale del 29/04/2024)	41
Figura 25: distribuzione verticale in cavo (a sinistra) e blindosbarra (a destra) (scatto personale del 29/04/2024)	41
Figura 26: collettore locale tecnico Nord (scatto personale del 29/04/2024)	42
Figura 27: sbarra equipotenziale quadro elettrico (sbarra in rame) (scatto personale del 29/04/2024)	43
Figura 28: panoramica dell'interfaccia di Revit 2019	44

Figura 29: schema della gerarchia degli oggetti (realizzazione personale)	45
Figura 30: modello del piano tipo	47
Figura 31: modello della cabina MT e locale BT	47
Figura 32: in alto, estrusione su percorso del file .dwg; a destra, vista in pianta della maglia di terra	48
Figura 33: browser di sistema del circuito di alimentazione O-3 (a sinistra) e del suo circuito di terra (a destra)	56
Figura 34: definizioni tensione (impostazioni elettriche)	57
Figura 35: sistemi di distribuzione (impostazioni elettriche).....	58
Figura 36: scelta sistema di distribuzione per il collettore.....	58
Figura 37: creazione parametro condiviso "Connesso a terra"	59
Figura 38: associazione parametro di progetto "Connesso a terra"	59
Figura 39: creazione parametro condiviso "Circuito di terra"	60
Figura 40: associazione parametro di progetto "Circuito di terra"	60
Figura 41: creazione regola filtro T term O-1	61
Figura 42: creazione regola filtro T term O-3.....	62
Figura 43: visualizzazione filtro T term O-1	62
Figura 44:visualizzazione filtro T term O-3	63
Figura 45: blocco selezione elemento di partenza	70
Figura 46: macroblocco "Analisi dei circuiti"	71
Figura 47: lettura circuiti.....	72
Figura 48: scelta circuito.....	72
Figura 49: rilevazione lunghezza	73
Figura 50: rilevazione nome quadro di partenza.....	73
Figura 51: conversione nome oggetto.....	73
Figura 52: macroblocco "Calcolo impedenza"	74
Figura 53: scelta parametri per ricavare i valori specifici.....	75
Figura 54: calcolo impedenza	75
Figura 55: estrapolazione valore impedenza totale.....	76
Figura 56: macroblocco "Calcolo tratto trasformatore-collettore MT"	77
Figura 57: macroblocco "Impedenza del trasformatore"	79
Figura 58: pannello delle proprietà torretta con valore dell'impedenza abbinato	80
Figura 59: macroblocco "Somma condizionale".....	81
Figura 60: finestra Lettore Dynamo.....	82

Figura 61: Plugin VR-READY	86
Figura 62: Barra degli strumenti di VR-READY.....	87
Figura 63: Finestra principale di Unity con modello importato.....	88
Figura 64: visore Meta Quest 3 e controller.....	97
Figura 65: navigazione nel modello virtuale e interazione con puntatore rosso.....	98
Figura 66: comparsa del pannello delle proprietà modificabili.....	98

1. BIM: Building Information Modeling

1.1 Definizione

Con l'acronimo BIM (Building Information Modeling) non si intende un software o un programma specifico di progettazione, bensì una metodologia operativa che consenta di realizzare un modello virtuale di una qualsiasi costruzione, da un piccolo oggetto fino a grandi complessi industriali e opere pubbliche.

Il BIM non restituisce soltanto una rappresentazione grafica 3D, ma associa ad ogni singola entità del modello una serie di dati e informazioni riguardanti ad esempio il materiale, le peculiarità geometriche, le caratteristiche energetiche e impiantistiche, che accompagneranno il modello dalla sua prima concezione fino alla demolizione.

Questa metodologia permette una profonda e completa sinergia tra esperti del settore e maestranze, sia in fase di progetto, che di realizzazione e manutenzione per tutto il ciclo vita, aumentando la produttività, l'efficienza, la qualità e la sostenibilità.

Una delle potenzialità permesse da questo metodo è senza dubbio quello di poter realizzare in corso d'opera e nello stesso ambiente di sviluppo abachi, computi metrici, analisi energetiche ed economiche solo per citarne alcune; con un semplice CAD (Computer-Aided Design) ad esempio, tutto ciò deve essere per forza realizzato all'esterno dell'ambiente di sviluppo con software aggiuntivi o plugin di dubbia provenienza, facendo lievitare i costi per le licenze aggiuntive e aumentando i rischi di incompatibilità di versioni dei software in uso tra i vari esperti.

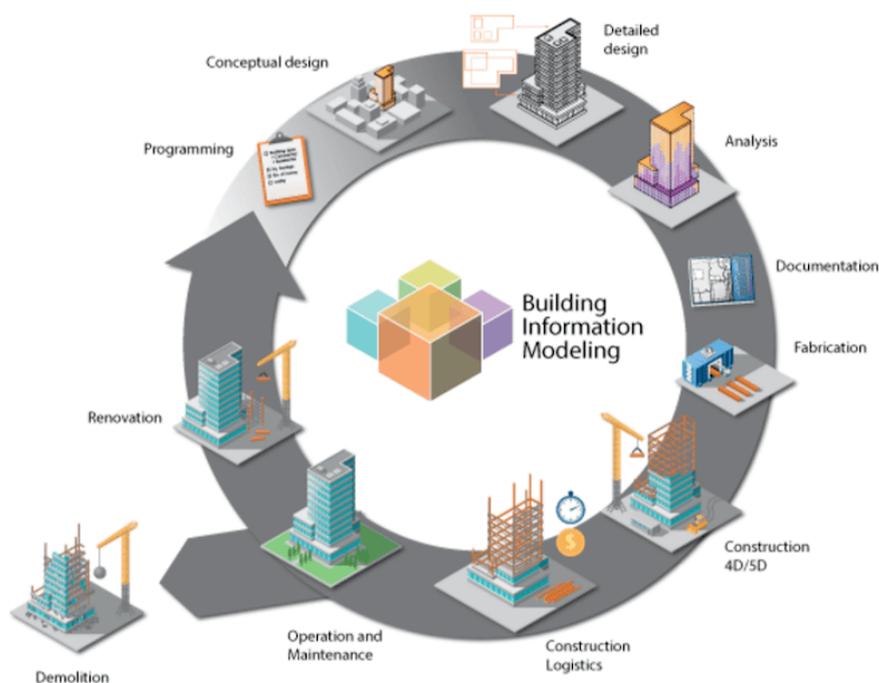


Figura 1: ciclo di vita degli edifici

Nella Figura 1 è possibile apprezzare l'intero ciclo di vita della costruzione, pienamente gestita dalla metodologia Bim; si possono individuare le seguenti fasi:

- programmazione;
- progettazione (che si articola nelle fasi di design concettuale, design dettagliato, analisi e documentazione);
- realizzazione (comprendente la fabbricazione, costruzione in loco, logistica);
- manutenzione e gestione;
- rinnovamento o demolizione. [1]

Il Bim modifica profondamente il modo di lavorare, spostando il focus dalla rappresentazione grafica al dato informativo, che diventa cuore pulsante dell'intero progetto: il secondo non sostituisce mai il primo, che resta parte integrante della realizzazione, contribuendo a realizzare il modello virtuale della costruzione.

Una delle sfide per chi si avvicina a questa modalità è quella di trovare il giusto equilibrio sulla quantità di dati, che devono essere selezionati attentamente per permettere un livello informativo adeguato al contesto e alla fase di realizzazione dell'opera ma allo stesso tempo evitando di appesantire inutilmente il modello.

Il processo di progettazione non è più di tipo "sequenziale" come nel CAD, con le attività di ogni singolo esperto condotta separatamente e poi ricomposta in un secondo momento, ma tutte le professionalità coinvolte portano in parallelo il proprio lavoro, abbracciando questo nuovo tipo di approccio, condiviso e collaborativo, nonché aperto e soprattutto integrato.

A tal proposito, è interessante analizzare la curva di MacLeamy in Figura 2, che propone l'effetto/sforzo/costo in funzione della fase progettuale:

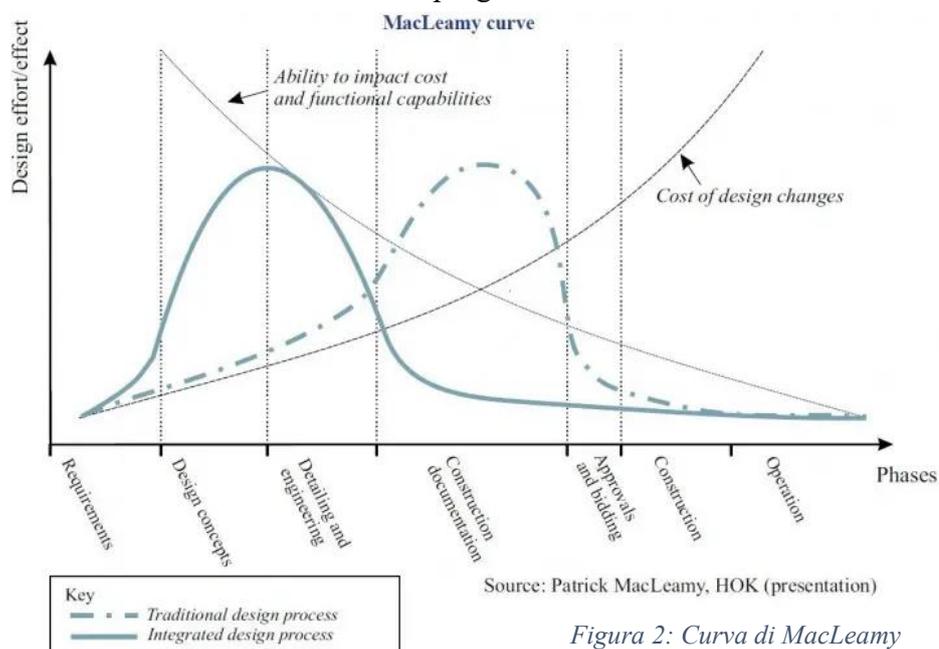


Figura 2: Curva di MacLeamy

Il BIM (linea continua azzurra nel grafico) anticipa rispetto la richiesta di sforzi in termini di tempo e denaro nelle fasi iniziali della progettazione, laddove i costi per le variazioni progettuali sono ancora contenuti e resta alta la capacità di influenzare i costi e le performance; al contempo, un approccio tradizionale come il CAD (linea tratteggiata azzurra) presenta un picco spostato a destra, che significa affrontare i problemi come ad esempio un'interferenza tra impianti in tempi successivi, quando è più difficile e oneroso porvi rimedio. [2]

Per mantenere alto il livello di interoperabilità e condivisione di cui si è parlato nei precedenti paragrafi, si è optato per utilizzare uno standard internazionale ovvero openBIM, elaborato da Building Smart International, che garantisce un flusso di lavoro digitale basato su formati aperti e privi di legami commerciali con i software.

In questo modo è possibile condividere senza intoppi tutto il set informativo attraverso l'intera filiera di progettazione, standardizzando il processo di trasmissione e garantendo i principi di:

- Interoperabilità
- Formato Aperto
- Affidabilità
- Collaborazione
- Flessibilità
- Sostenibilità [3].

Nello specifico, l'IFC (Industry Foundation Classes), assieme a BCF, COBie, CityGML, gbXML e altri, risulta il formato aperto più utilizzato, in grado di organizzare attraverso standard e corrispondenze univoche in una struttura ordinata le seguenti componenti di un modello BIM [4]:

- identità e semantica: identificativo univoco che riferisca nome, oggetto e funzione;
- caratteristiche e attributi: le informazioni riguardanti le proprietà chimico-fisiche dell'oggetto;
- relazioni in essere:
 - tra gli oggetti, quindi ad esempio le connessioni a livello strutturale e sistemico;
 - tra i processi di analisi, come report sui consumi energetici o sulla gestione degli impianti;
 - tra le azioni installative e manutentive;
 - tra le professionalità che collaborano alla stesura del progetto.

L'IFC è importantissimo perché permette il trasferimento delle geometrie e delle informazioni senza modificare minimamente la struttura complessiva o delle parti (Figura 3).

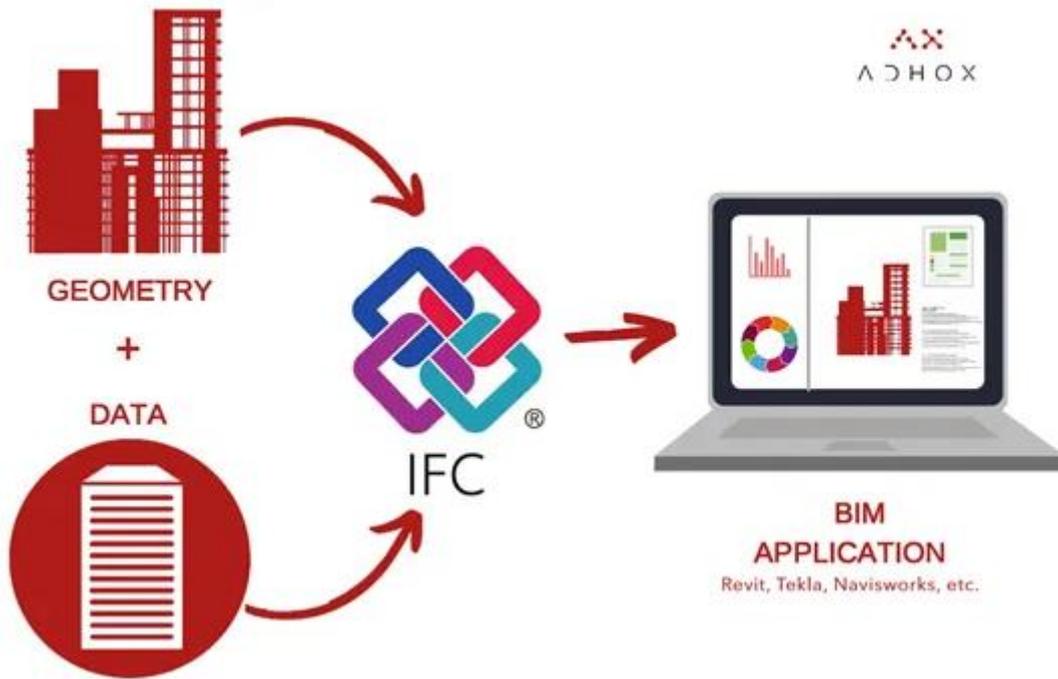


Figura 3: schema concettuale formato IFC

1.2 Leggi e normative

1.2.1 Anni 2000

I primi a porsi il problema di una standardizzazione sono stati gli Stati Uniti che, tramite il National Institute of Building Sciences (NIBS), hanno sviluppato il National BIM Standard (oggi alla sua quarta versione [5]) per promuovere l'uso del BIM nel settore delle costruzioni.

Inoltre, diverse agenzie governative, tra cui il General Services Administration (GSA), hanno reso obbligatorio l'uso del BIM per progetti pubblici, contribuendo a diffondere l'adozione del BIM a livello federale. La GSA, nel 2003, è stata la prima a richiedere l'uso del BIM per la progettazione di edifici federali, segnando l'inizio dell'integrazione di standard per le opere pubbliche.

In un secondo momento anche altri settori si sono adeguati, come quello medico in cui il Veterans Affairs (VA) ha reso obbligatorio il BIM per la progettazione di strutture sanitarie e infrastrutture correlate e quello militare, con l'interessamento da parte del Department of Defense (DoD) e il U.S. Army Corps of Engineers (USACE).

1.2.2 Anni 2010

Negli anni successivi anche il Regno Unito ha iniziato ad adeguarsi, introducendo formalmente il BIM con il UK Government Construction Strategy del 2011. Questo documento stabiliva che entro il 2016 sarebbe stato obbligatorio l'uso del BIM Level 2 per tutti i progetti pubblici per aumentare l'efficienza e ridurre i costi [6].

Degne di nota sono state la serie di specifiche PAS 1192, attualmente non disponibile per la consultazione perché sostituita dalla BS EN ISO 19650 [7] nel 2019.

Assieme a questo paese troviamo anche la Norvegia in cui si è fatto promotore l'ente pubblico Statsbygg che dal 2010 ha richiesto l'uso del BIM per i progetti pubblici, posizionandosi come uno dei primi Paesi ad adottare il BIM in maniera sistematica [8].

Di pari passo ha agito la Finlandia grazie all'iniziativa COBIM (Common BIM Requirements), promossa dal governo: dal 2007, l'uso del BIM è diventato obbligatorio per i progetti di infrastrutture pubbliche gestiti da enti come il Senate Properties, il principale gestore delle costruzioni pubbliche finlandesi [9].

1.2.3 Anni 2014 e seguenti

È solo nel 2014 che a livello europeo viene adottata una direttiva comune, la 2014/24/UE, nata con lo scopo di regolamentare gli appalti pubblici all'interno dell'Unione Europea: è al suo

interno che si cita la possibilità della richiesta da parte degli Stati membri di richiedere alle stazioni appaltanti l'utilizzo di strumenti di simulazione elettronica o similari per la progettazione [10].

A livello italiano, se ne tratta approfonditamente nel Nuovo Codice degli Appalti (D. Lgs. 50/2016) [11], ma solo con il Decreto Baratonò (D.M. 560/2017) lo si attua, definendo “le modalità e i tempi di progressiva introduzione, da parte delle stazioni appaltanti, delle amministrazioni concedenti e degli operatori economici, dell'obbligatorietà dei metodi e degli strumenti elettronici specifici, quali quelli di modellazione per l'edilizia e le infrastrutture, nelle fasi di progettazione, costruzione e gestione delle opere e relative verifiche” [12].

Sono seguite deroghe e modifiche, come il D.M. 312/2021 [13] e il Nuovo codice degli appalti (D.Lgs. 36/2023) [14], dai quali si sono delineati i seguenti scaglioni per l'obbligatorietà del BIM:

- Dal 1° Gennaio 2019 per lavori con importo maggiore o uguale a 100 milioni di euro;
- Dal 1° Gennaio 2020 per lavori con importo maggiore o uguale a 50 milioni di euro;
- Dal 1° Gennaio 2022 per lavori con importo maggiore o uguale a 15 milioni di euro;
- Dal 1° Gennaio 2023 per lavori con importo maggiore alla soglia indicata dall'articolo 35 del codice dei contratti pubblici (5,35 milioni di euro);
- Dal 1° Gennaio 2025 per lavori con importo maggiore o uguale a 1 milione di euro.

Nel nuovo codice degli appalti trovano posto due allegati importanti: l'I.7 si occupa della gestione tramite il BIM del capitolato informativo, l'I.9 richiama le norme UNI EN ISO 19650 e UNI 11337 [15]. La prima, come già anticipato precedentemente, deriva dalle PAS britanniche ed è diventata nel corso degli anni la norma di riferimento per tutti gli Stati; divisa in cinque parti, si occupa della gestione delle informazioni tramite il BIM, dividendo i requisiti in 3 gruppi e altrettante sottocategorie:

- Requisiti dei soggetti interessati
 - OIR (Organization Information Requirements) – Requisiti Informativi della pianificazione;
 - PIR (Project Information Requirements) – Requisiti Informativi del progetto;
- Requisiti informativi dell'incarico
 - AIR (Asset Information Requirements) – Requisiti Informativi dell'asset;
 - EIR (Exchange Information Requirements) – Requisiti di scambio delle informazioni;
- Deliverables informativi
 - PIM (Project Information Model) – Modello Informativo del progetto;
 - AIM (Asset Information Model) – Modello Informativo dell'asset [16].

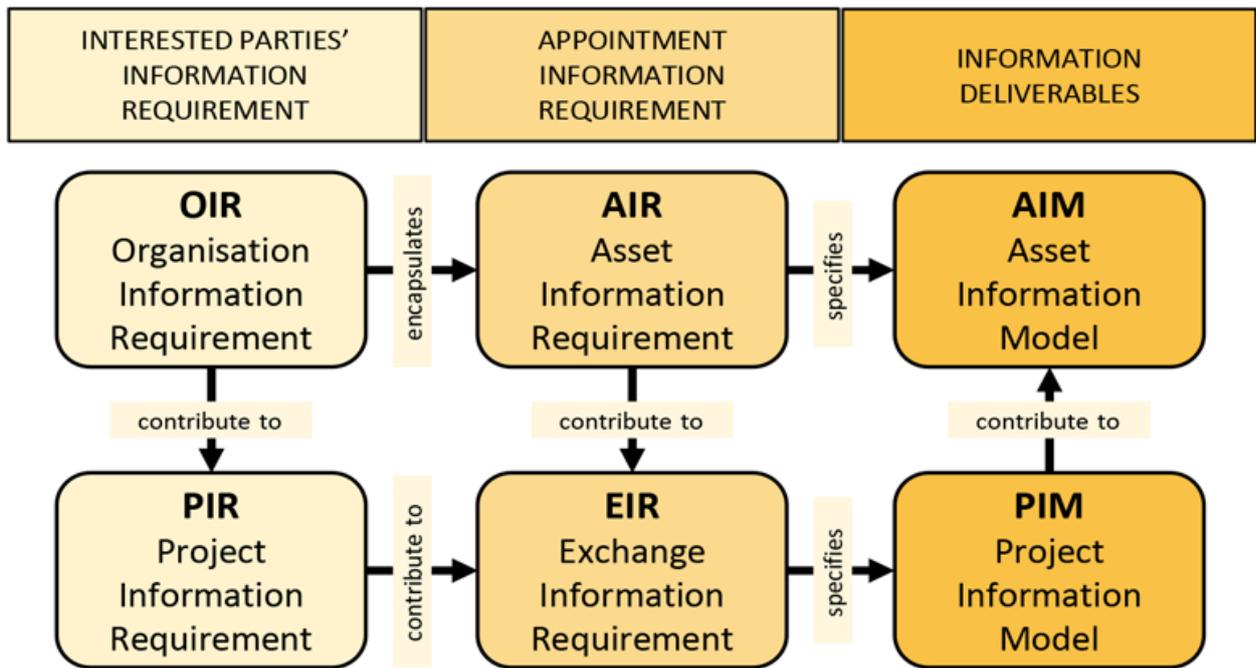


Figura 4: schema requisiti informativi secondo la ISO 19650

La norma appena descritta è essenziale per capire il significato di requisito informativo (o information requirement-IR) che altro non è che un dato che diventa informazione poiché fornito alla persona giusta al momento giusto. Per generare i requisiti, che devono poi essere trasformati in attributi del modello, si usa spesso il metodo delle “Plain Language Questions” [17], ovvero quello di utilizzare delle domande poste utilizzando un linguaggio non tecnico, utile per far dialogare facilmente due figure come il committente e il Bim Manager.

Per fare un esempio collegato alla gestione degli impianti di terra e quindi a questa trattazione, ci si è posti la seguente domanda:

“Come posso gestire la manutenzione degli impianti di terra?”

Da ciò è iniziata la stesura di un elenco di verifiche da fare legate agli impianti elettrici che è sfociata poi nella scheda di manutenzione descritta al capitolo 4.2 e proposta in versione integrale all’appendice 1.

La seconda norma introdotta prima, ovvero la UNI 11337, con le sue sette parti, risulta essere il “proseguimento nazionale” della norma europea. Al suo interno, nello specifico all’interno della parte 1, si fa la distinzione tra elaborati e modelli e si delineano cinque livelli di maturità digitale, in ordine crescente dal modello privo di controparte digitale al modello caratterizzato da un completo ventaglio di informazione e dati.

Nella Figura 5, viene proposta una visione schematica dei livelli, con una breve descrizione di ciascuno di essi [18].

Livello di Maturità (UNI 11337-1)	Livello 0	Livello 1	Livello 2	Livello 3	Livello 4
Esigenze individuate dalla norma	Trasferimento di contenuti informativi tramite elaborati informativi non digitali	Trasferimento di contenuti informativi attraverso elaborati informativi digitali e non	Trasferimento di contenuti informativi prevalentemente attraverso modelli informativi grafici, eventualmente accompagnati da elaborati informativi grafici digitali	Modelli informativi grafici ed elaborati comunicano attraverso schede informative digitali di prodotto e di processo	Trasferimento di contenuti informativi attraverso modelli informativi (virtualizzabili in senso grafico, documentale, multimediale)

Figura 5: livelli di maturità BIM secondo la UNI 11337-1

Nella parte 4 vengono invece definiti i livelli di approfondimento delle informazioni associate al modello 3D dell'opera realizzate e sono il pilastro fondamentale per definire il grado di sviluppo di un progetto. I vari LOD si susseguono in una scala gerarchica crescente, partendo da un livello in cui l'oggetto, nel suo stato embrionale, consiste in un simbolo, ai livelli più alti, in cui l'oggetto è sempre più arricchito di dettagli e informazioni.

Sulla falsa riga del caso americano o britannico, l'acronimo LOD significa letteralmente "Level of Definition" ovvero Livello di definizione ed è caratterizzato dalla fusione di altri due acronimi: LOG (Level Of Geometry) e LOI (Level Of Information). Nel caso italiano, la classificazione avviene sulle categorie di oggetti e la scala gerarchica viene univocamente gestita grazie alle lettere dell'alfabeto (contrariamente a quanto usato in America, ovvero le centinaia, oppure in Regno Unito con le unità).

La scala gerarchica è la seguente:

- LOD A (oggetto simbolico);
- LOD B (oggetto generico);
- LOD C (oggetto definito);
- LOD D (oggetto dettagliato);
- LOD E (oggetto specifico);
- LOD F (oggetto eseguito);
- LOD G (oggetto aggiornato).

Il concetto di LOIN (Level Of Information Need) nasce successivamente per superare il concetto di LOD e spostare l'attenzione dalle caratteristiche dell'oggetto alla necessità di informazioni per soddisfare le esigenze del professionista in una determinata fase del progetto.

Il LOIN si divide a sua volta in due parti:

1. Prerequisiti, che sono quattro e definiscono quante e quali informazioni devono essere incluse in ogni oggetto, in base a:
 - scopo (perché?);
 - scadenze (quando?);
 - attori coinvolti (chi?);
 - organizzazione in una o più strutture di scomposizione (cosa?).
2. Livello di fabbisogno informativo, che combina tre tipi di informazioni:
 - geometriche, espresse utilizzando forma, grandezza, dimensione e posizione;
 - alfanumeriche, espresse utilizzando caratteri, cifre, simboli, ecc.;
 - documentazione, cioè l'insieme di documenti relativi a un soggetto indicato.

Questi concetti verranno poi ripresi successivamente ed adattati al caso specifico degli impianti oggetto di questa trattazione.

La parte 6 della norma UNI 11337 introduce e approfondisce il concetto di dimensione del BIM, passo fondamentale con cui ci scontra andando oltre la mera progettazione geometrica. Le categorie sono le seguenti:

- BIM 3D: Modellazione BIM parametrica, tridimensionale, che, come detto precedentemente coniuga il disegno geometrico con le informazioni sottoforma di parametri;
- BIM 4D: La programmazione temporale, che ha come vantaggi l'ottimizzazione della pianificazione dei tempi tra i vari partecipanti alla progettazione e costruzione, tenendo sotto controllo i progressi ed evitando conflitti durante la costruzione dovuti ad esempio a modifiche in corso d'opera;
- BIM 5D: L'analisi dei costi, utile per controllare i progressi dei costi nel tempo, aumentando precisione e prevedibilità delle stime dei costi. Con questa dimensione diventano disponibili l'estrazione e l'analisi delle spese, assieme alla simulazione di scenari conseguenti alle modifiche.
- BIM 6D: La gestione operativa del bene, che include tutta la parte di gestione e manutenzione durante tutto il ciclo vita. Si tiene conto delle attività, lo stato dei componenti, le specifiche, i manuali di manutenzione / installazione / gestione, le garanzie ecc.
- BIM 7D: Sostenibilità di tipo ambientale, economico e sociale: facendo previsioni già in fase di progetto è possibile arrivare alla realizzazione con un edificio pienamente ottimizzato a livello energetico e ambientale.

In fase di discussione vi sono poi altre tre dimensioni, ovvero:

- BIM 8D: Sicurezza in cantiere, sia in fase di progettazione che esecuzione dei lavori, agevolando il CSE (Coordinatore per la sicurezza in fase di esecuzione dei lavori) nel controllo del rispetto del D.Lgs. 81/2008 e del PSC (piano di sicurezza e coordinamento);
- BIM 9D: Costruzione snella, ottimizzando le fasi realizzative dell'opera, una gestione efficiente delle risorse e una minimizzazione degli sprechi: un esempio è quello di evitare anticipi o ritardi sulla fornitura dei materiali che causerebbero, rispettivamente, ingombro inutile in cantiere o allungamento dei tempi di consegna.
- BIM 10D: Costruzione Industrializzata, inserendo nuove tecnologie, dati fisici, commerciali o ambientali per incrementare la produttività. Un caso esemplificativo è quello di un'organizzazione attenta delle lavorazioni da parte del direttore dei lavori, che, tenuto conto della compatibilità con le condizioni meteorologiche, sappia se e con quale attività procedere in ogni istante, riducendo al minimo i tempi morti [19].

Dimensioni secondo la UNI 11337							In fase di dibattito		
1D	2D	3D	4D	5D	6D	7D	8D	9D	10D
Organizzazione	Modellazione bidimensionale	Modellazione tridimensionale	Gestione temporale	Gestione economica	Ciclo di vita e manutenzione	Sostenibilità	Sicurezza	Costruzione snella	Costruzione industrializzata

Figura 6: Dimensioni del BIM

1.3 La torre della Regione Piemonte

Il caso studio oggetto di questa trattazione è il grattacielo della Regione Piemonte, inaugurato nel 2022 e situato al centro del triangolo con ai vertici la stazione ferroviaria Lingotto, l'Oval e la futura Città della Salute.

Al Politecnico di Torino è stata affidata la realizzazione di un modello con metodologia BIM, ed è intenzione di questa tesi proseguire il lavoro fin qui fatto, con particolare attenzione agli impianti elettrici e nello specifico l'impianto di terra.



Figura 7: esterno della Torre (scatto personale del 29/04/2024)

Il complesso della Regione è suddiviso in cinque edifici ovvero: Torre, Interrati torre, Centro servizi, Interrati parcheggi e Promenade (passeggiata di collegamento tra la Torre e la Stazione).

Il modello è stato realizzato secondo una struttura federata, ovvero un file per ogni edificio, a loro volta suddivisi in sette discipline ossia: Architettonico, Strutturale, Meccanica, Elettrica, Idrico, Antincendio, Arredi.

Sono state redatte delle linee guida dal gruppo di ricerca responsabile Drawing To The Future [20], secondo cui, per motivi di gestione, ciascun file non deve superare la dimensione massima di 200 MB e seguire la seguente codifica:

Progetto_Edificio_Disciplina_Tipologia_Livello iniziale_Livello finale.

All'interno di ogni progetto le famiglie sono definite dal seguente nome in codice:

- Codice Progetto, (3 lettere);
- Codice Edificio, (2 lettere);
- Codice Disciplina, (3 lettere);
- Codice Famiglia, (2 lettere).

I tipi, che sono le alternative disponibili all'interno di ciascuna famiglia, vengono invece univocamente indicate così:

- Codice Funzione, (2 lettere);
- Codice Caratteristica, (alfanumerico).

A completare la definizione di ogni istanza, ovvero di ogni singolo elemento presente nel progetto, si aggiungono 14 parametri comuni presenti sempre, indipendentemente dalla disciplina di appartenenza. Uno di questi è l'identificativo che, seguendo la sintassi richiesta dalla guida, è il seguente:

CodiceFamiglia_CodiceTipo_Livello_Numero progressivo

È così che nel seguente codice univoco si può estrapolare quanto detto fino ad ora:

TRP_TO_ELE_QS_AL_O_LF04_00081

- TRP: Codice Progetto, "Torre Regione Piemonte";
- TO: Codice Edificio, "Torre";
- ELE: Codice Disciplina, "Elettrica";
- QS: Codice Famiglia, "Quadro Servizi";
- AL_O: Codice Tipo, "Alimentazione Ordinaria";
- LF04: Livello, "Quarto Livello Fuori terra";
- 00081: Numero Progressivo.

1.3.1 Sistema di alimentazione del complesso

L'intero complesso viene alimentato dal distributore attraverso una cabina di ricezione sita su Via Farigliano, angolo Via Nizza: per motivi di ridondanza i punti di fornitura sono due, ciascuno con il proprio quadro di consegna, uno principale e l'altro di riserva.

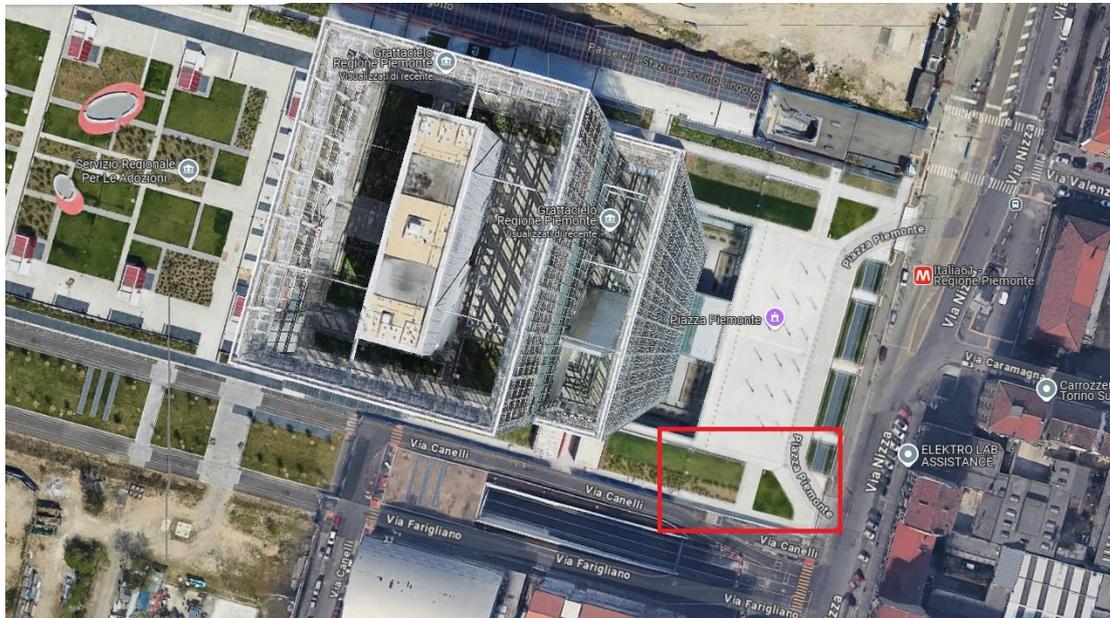


Figura 8: locazione cabina di ricezione MT

Da questo punto si snoda una rete in Media Tensione a 22 kV, con configurazione ad anello aperto: questa scelta si rivela la migliore nel caso vi sia un guasto sulla rete interna. Infatti, una volta individuato il punto problematico, è possibile aprire l'anello a monte di esso e chiuderlo subito a valle, così da circoscrivere la zona e contro alimentare le cabine elettriche attraverso il tronco di rete sano, minimizzando i disagi. Risulta molto utile anche in vista della manutenzione per evitare fuori servizi o fermi impianto: tutto ciò implica però costi di realizzazione maggiori dovuti all'estensione della rete e complessità in termini di protezioni.

Nello specifico sono presenti tre locali cabine di ricezione e smistamento (le prime due dedicate ai punti di consegna e la terza per la connessione su unica sbarra dei due ingressi) più sei cabine di trasformazione lungo l'anello. Queste sono:

1. LOCALE CABINA MT/BT AUTORIMESSA MT-03;
2. LOCALE CABINA MT/BT IMP. FLUIDOMECCANICI MT-02;
3. LOCALE CABINA MT/BT IMP. ANTINCEDIO MT-ANT;
4. LOCALE CABINA MT/BT CENTRO SERVIZI MT-04;
5. LOCALE CABINA MT/BT TORRE MT-01;
6. LOCALE CABINA MT/BT EMERGENZA MT-E.

Segue uno schema di massima della struttura ad anello in Figura 9, dove si può notare la configurazione entra-esce di ciascuna cabina.

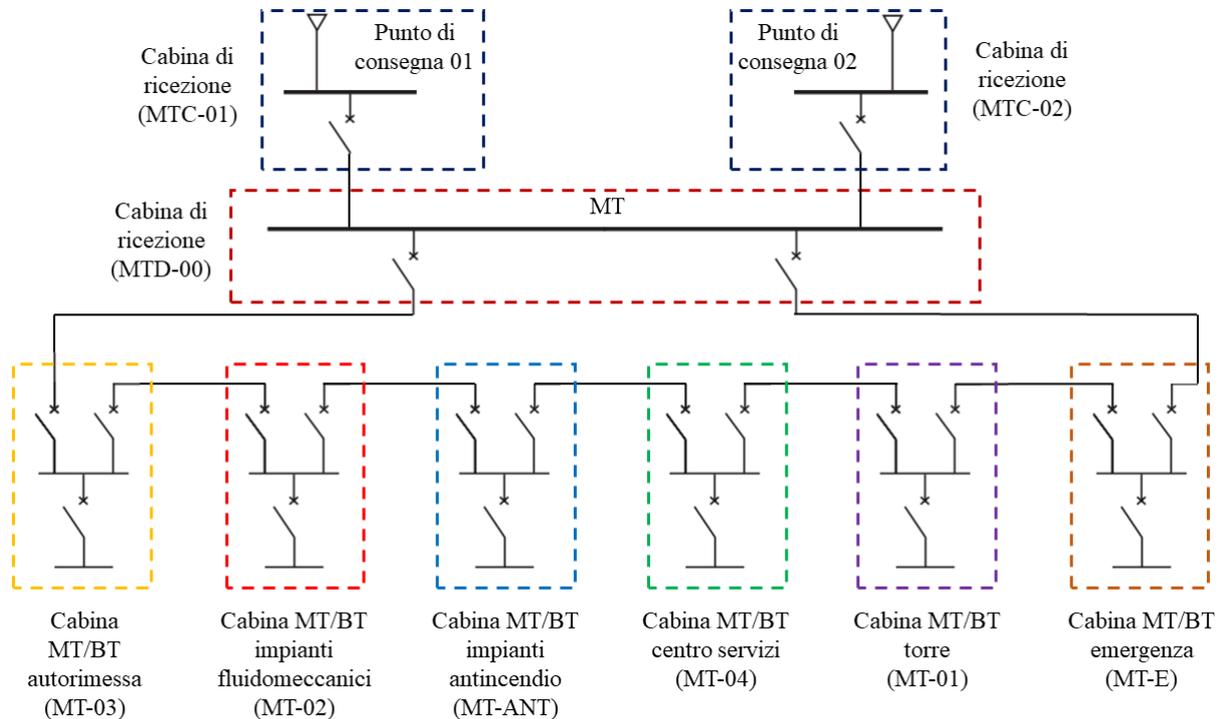


Figura 9: schema di massima rete di distribuzione (elaborazione personale)

Siccome il fine di questa opera è quello di analizzare l'impianto di terra relativo alle apparecchiature dell'edificio torre, nello specifico quelle del piano 4, ci si limiterà alla descrizione delle cabine interessate, ovvero quella di media tensione denominata "Cabina MT/BT torre (MT-01)" e del locale bassa tensione dello stesso edificio.

La prima presenta cinque trasformatori, uno dedicato alla sezione ordinaria ($P_n=2.5$ MVA, $V_{1n}/V_{2n}=22/0.4$ kV, $V_{cc\%}=6\%$) e quattro a quella privilegiata, due con gli stessi dati dell'ordinario (di cui uno di riserva, il TR.3) e altri due più piccoli ($P_n=0.5$ MVA, $V_{1n}/V_{2n}=22/0.4$ kV, $V_{cc\%}=6\%$).

Nella seconda si trovano i vari quadri in bassa tensione:

- QGBT.1: quadro principale di distribuzione, presenta al suo interno la sbarra di ordinaria (O) e privilegiata (P15), separate da un congiuntore normalmente aperto. In condizioni di malfunzionamento o manutenzione, una serie di congiuntori vengono opportunamente manovrati per la riconfigurazione della rete;
- QGBT.2: per la sezione "SAFETY", alimentato dai due trasformatori più piccoli della cabina di media tensione della torre. Esso si occupa di alimentare la sezione volgarmente detta "sicura o di continuità" SAFETY A e SAFETY B, ovvero quella relativa a due UPS e relativi quadri:
 - UPS.5A (120 kVA, 120 min.) e QGBT.5;

- UPS.5B (120 kVA, 120 min.) e QGBT.6;

Ad essi afferiscono i circuiti relativi all'illuminazione di sicurezza, rivelazione incendi, antintrusione, controllo accessi e BMS (Building Management System);

- due quadri alimentati dalla sbarra di privilegiata, deputati entrambi a diverse sezioni di alimentazione “sicura o di continuità”:
 - QGBT.3 alimentato dai due UPS.3A e UPS.3B (800 kVA, 15 min.) per la sezione “CONTINUITÀ”;
 - QGBT.4 alimentato dall'UPS.4 (160 kVA, 60 min.) per la sezione “SECURITY”;
- diversi quadri per le operazioni di commutazione, per realizzare più riconfigurazioni:
 - QE.T.COMM.1;
 - QE.T.COMM.2;
 - QE.T.COMM.3;
 - QE.T.COMM.4;
 - QT.T.P45.02 (QE.T.COMM.5);
 - QT.T.P37.03 (QE.T.COMM.6);
 - QT.T.P42.01-2 (QE.T.COMM.7)
 - QE.T.COMM.7;
 - QE.T.COMM.8.

Sono inoltre presenti due gruppi elettrogeni, ciascuno da 2 MVA 400 V: sono collegati alla cabina di emergenza e la loro tensione viene elevata a 22 kV tramite due trasformatori innalzatori con $P_n=2$ MVA, $V_{1n}/V_{2n}=22/0.4$ kV, $V_{cc}\%=6\%$.

L'impianto di terra segue abbastanza fedelmente il percorso dei conduttori di alimentazione, inclusa la distribuzione verticale tra i piani effettuata tramite blindosbarra: ve ne sono tre in totale, installate in un apposito cavedio, una per ogni tipologia di alimentazione (ordinaria, privilegiata e sicura). Al loro interno vi sono cinque sbarre, tre per le fasi, una per il neutro e una per la terra. Ad ogni piano vi è una derivazione tramite scatola: di lì si procede in cavo fino al quadro di piano di riferimento e successivamente, per quanto riguarda le utenze a pavimento in torretta, la distribuzione avviene tramite blindosbarre nel pavimento; per le altre attrezzature elettriche il percorso avviene totalmente in cavo.

La struttura dettagliata dell'impianto di terra avverrà nel prossimo capitolo, dopo un approfondimento teorico su detto impianto.

2. Impianto di terra

2.1 Cenni teorici e sicurezza elettrica

Prima di vedere nel dettaglio la struttura dell'impianto di terra in esame, è doveroso fare un approfondimento su alcuni concetti teorici fondamentali [21].

Innanzitutto la prima grande distinzione è quella tra:

- la *messa a terra di protezione*: è necessaria per limitare le tensioni pericolose agenti su chiunque possa venir a contatto con le masse, ovvero parti metalliche di impianti o apparecchi utilizzatori normalmente non in tensione ma che possono divenirlo in seguito ad un guasto; inoltre favoriscono l'intervento dei dispositivi di protezione. In questo gruppo si annoverano:
 - gli impianti di protezione contro le scariche atmosferiche;
 - gli impianti relativi ai dispositivi contro l'accumulo di cariche elettrostatiche;
 - gli impianti relativi ad apparecchiature elettroniche munite di filtri.
- la *messa a terra di funzionamento*: come ad esempio quella di mettere a terra il centro stella dei sistemi elettrici AT, per un esercizio più regolare;
- la *messa a terra per lavori*: di carattere provvisorio, per permettere l'accesso in sicurezza a parti di impianto normalmente in tensione e messe fuori servizio per lavori.

Dallo studio degli andamenti del potenziale sulla superficie del terreno, che viene tralasciata per non appesantire troppo la trattazione, si deduce che un impianto di terra è tanto migliore quanto più rende uniforme tale distribuzione.

A questo punto è necessario introdurre dei parametri fondamentali:

- la *tensione di contatto* U_T : è la tensione a cui è sottoposto un uomo in contatto con un elemento accessibile andato in tensione (tra elemento in tensione e piedi dell'uomo vi è 1 metro);
- la *tensione di passo* U_S : è la tensione che può essere applicata ai piedi di una persona (distanza convenzionale di 1 metro) che transiti su un suolo percorso da corrente;
- la *tensione di contatto a vuoto* U_{ST} o U_{vT} : è la tensione (misurata con un voltmetro a resistenza interna pressochè infinita) fra l'elemento metallico in tensione e un punto del terreno, posti ad un metro di distanza l'uno dall'altro;
- la *tensione di passo a vuoto* U_{SS} : la tensione che si misurerebbe tra due punti del terreno posti alla distanza di un metro, utilizzando un voltmetro uguale a quello sopra;

- la *corrente di guasto a terra* I_F : massima corrente di guasto verso terra dell'impianto elettrico;
- la *corrente di terra* I_E : frazione della corrente di guasto a terra I_F che l'impianto di terra disperde nel terreno;
- la *tensione totale di terra* U_E : tensione stabilita tra un impianto di terra che sta disperdendo la corrente di terra I_E e un punto sufficientemente lontano assunto a potenziale zero;
- la *resistenza totale* R_{TOT} : data dalla somma della resistenza del corpo umano R_B , della resistenza delle scarpe ($1\text{ k}\Omega$) e della resistenza verso terra della persona (assunta pari a $1,5\rho_s$, con ρ_s pari alla resistività superficiale del suolo);
- la *resistenza di terra* R_E di un impianto di terra: calcolata tra il dispersore e la terra di riferimento e uguale a $\frac{U_E}{I_E}$.

Nelle immagini seguenti è possibile visualizzare graficamente alcune delle grandezze introdotte:

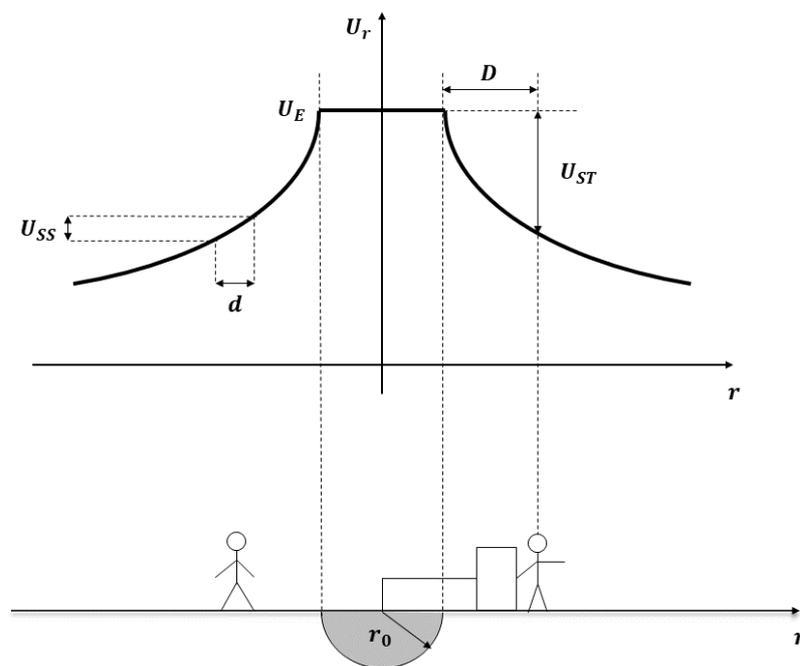


Figura 10: tensioni di contatto e di passo [21]

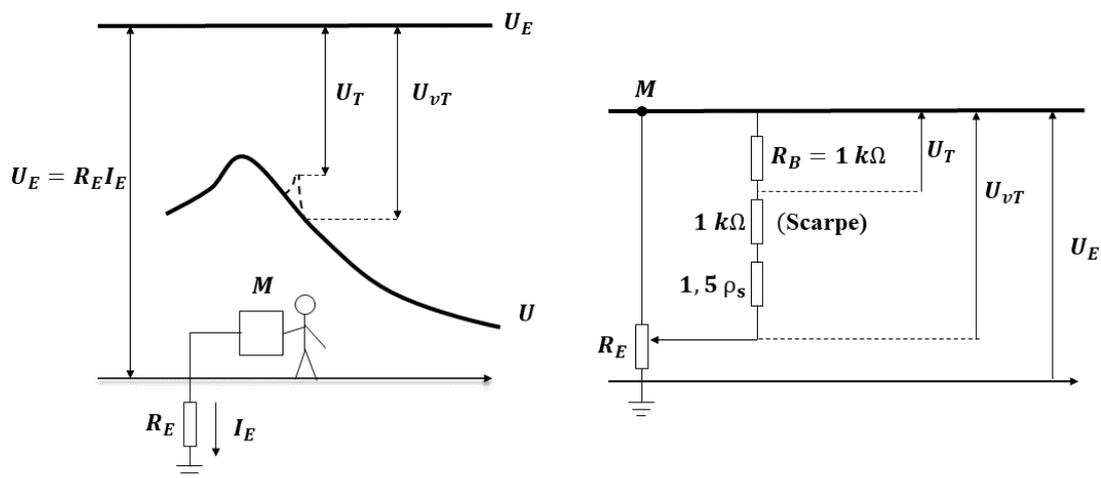


Figura 11: tipi di resistenza e relativo potenziale [30]

Consultando la norma è possibile individuare la curva della tensione di contatto ammissibile U_{Tp} in funzione della durata del guasto t_f . Nella figura 12 vi è il confronto tra la curva mostrata nella vecchia CEI 11-1 e quella nella norma vigente attualmente ovvero la CEI EN 50522 (Classificazione CEI 99-3) [22]:

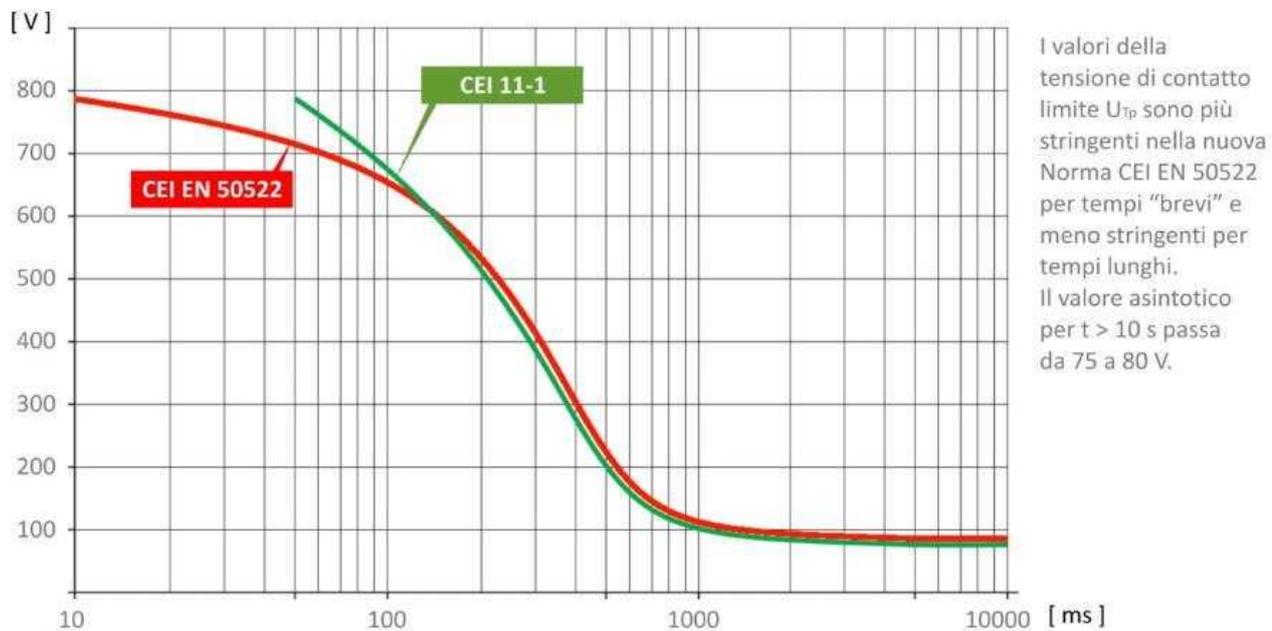


Figura 12: tensione di contatto limite
(<https://www.overtec.it/articoli/impianti-elettrici/sostituita-la-norma-cei-11-1-dalle-en-61636-1-ed-en-50522>)

2.1.1 Progettazione secondo la norma

Relativamente ad un guasto in media tensione, un impianto di terra viene considerato sicuro se la tensione di contatto che si stabilisce in un qualsiasi punto di esso (considerato nella sua interezza, MT+BT) non supera la tensione di contatto ammissibile U_{Tp} .

Secondo la CEI 99-3, che ragiona sulla resistenza di terra R_E essendo un parametro più facilmente misurabile, è sufficiente che venga verificata tale condizione:

$$R_E \leq \frac{U_{Tp}}{I_E} \quad (1)$$

Questo poiché se

$$U_E = R_E \cdot I_E \leq U_{Tp} \quad (2)$$

l'impianto di terra garantisce per forza la sicurezza, essendo che

$$U_T \leq U_E \quad (3)$$

Nel caso in cui la (1) non fosse verificata, prima di considerare l'impianto non adeguato, è necessario misurare le tensioni di contatto (U_T): se

$$U_T \leq U_{Tp} \quad (4)$$

allora l'impianto di terra può ritenersi idoneo.

Nel caso specifico in cui l'impianto sia costituito da una rete di terra magliata su tutta l'area della cabina e dello stabilimento, la condizione di adeguatezza da rispettare è

$$U_E \leq 2U_{Tp} \quad (5)$$

Se non lo fosse nemmeno così, si utilizza una condizione più permissiva ovvero

$$U_E \leq 4U_{Tp} \quad (6)$$

che se fosse verificata richiederebbe l'adozione di alcuni provvedimenti (i cosiddetti provvedimenti M) atti ad aumentare il potenziale, o la resistività superficiale, del suolo: in questo modo si riesce a riportare la tensione di contatto entro i limiti di sicurezza.

Si tiene a precisare che nel caso in cui l'impianto di terra venisse collegato ad un impianto di terra globale, ovvero ad un sistema di impianti di terra zonali (ad esempio tutti gli impianti delle cabine di trasformazione della zona interconnesse tramite gli schermi dei cavi di media tensione), non sarebbe necessario tutto l'iter di dimensionamento, poiché verrebbe considerato sicuro per definizione: è compito del distributore informare l'utente della disponibilità di tale impianto.

Il seguente diagramma di flusso riassume i passi fin qui descritti:

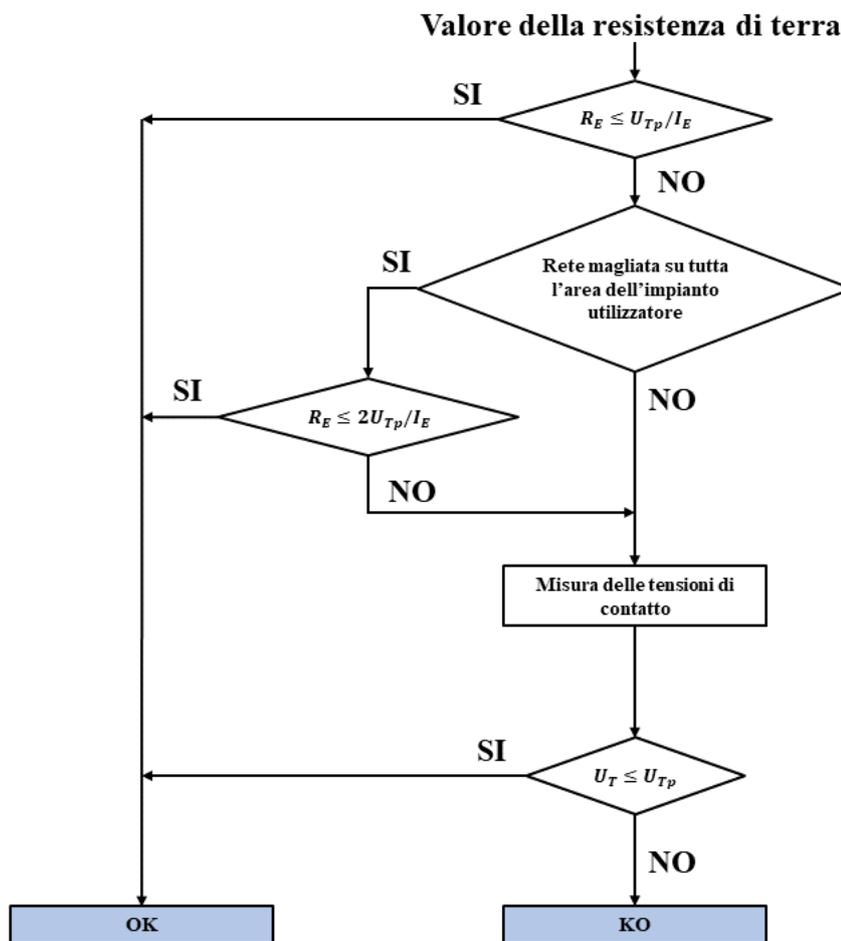


Figura 13: schema ridotto verifica impianti di terra [30]

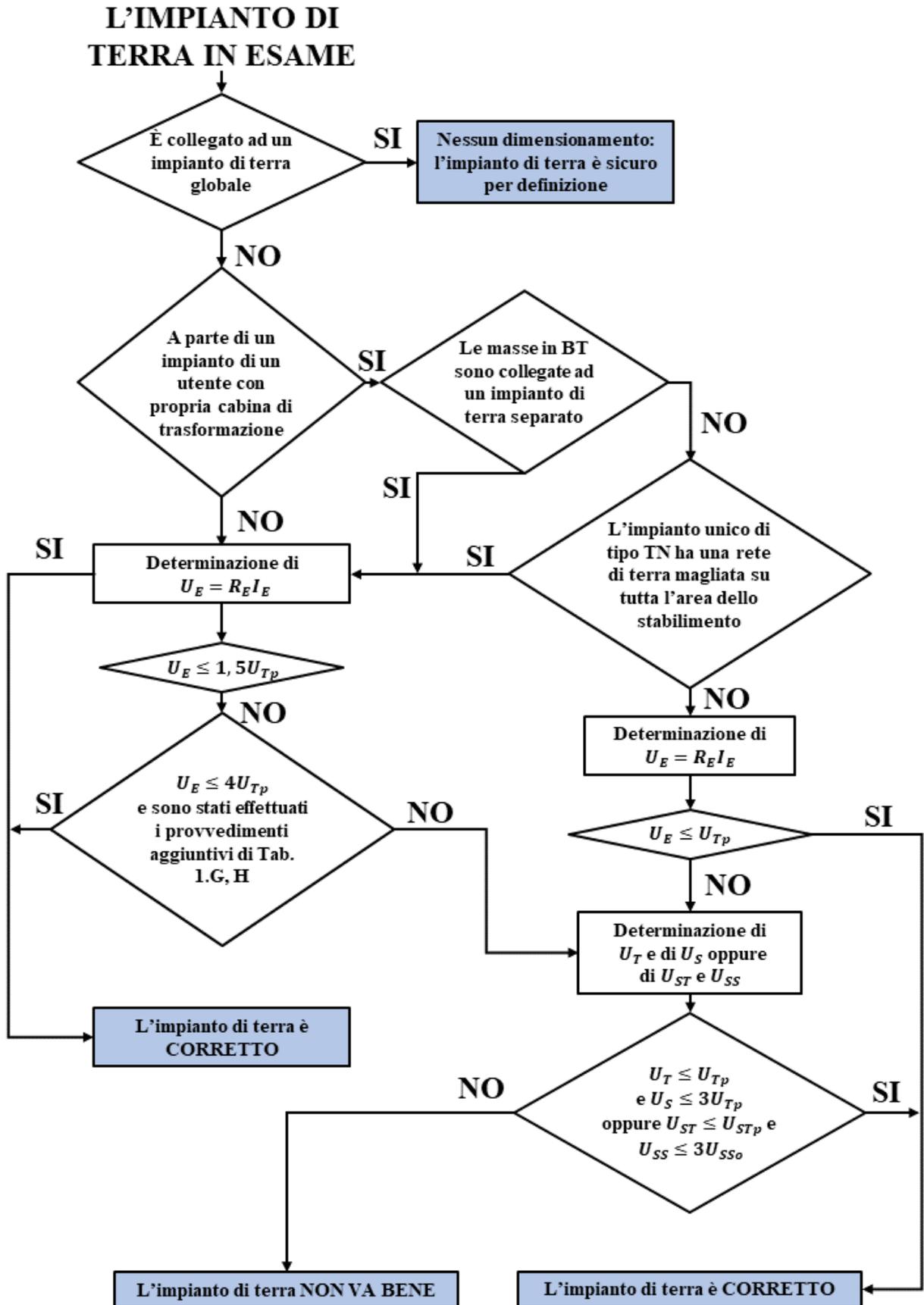


Figura 14: schema completo verifica impianti di terra [21]

Secondo la CEI 0-16 [23], tutte le cabine connesse alla rete pubblica di media tensione dopo il 01/09/2008 devono essere progettate idoneamente affinché rispettino contemporaneamente le seguenti condizioni:

- corrente di guasto a terra (I_F) e tempo di estinzione del guasto (t_f) comunicati dal distributore;
- corrente di guasto a terra di 40 A (15 kV) o di 50 A (20 kV) e durata del guasto maggiore di 10 s (o altri valori in base alla tensione nominale).

Questo affinché l'impianto sia in grado di essere alimentato sia da una rete a neutro compensato che isolato. Inoltre, va detto che le protezioni di terra (51N ovvero il relè omopolare e la 67N ovvero la direzionale di terra) servono ad evitare che un qualsiasi guasto lato utente faccia intervenire le protezioni di linea del distributore, causando la disalimentazione di tutti gli utenti serviti da tale linea.

Per quanto riguarda invece un guasto lato bassa tensione (come ad esempio un cedimento dell'isolamento in grado di mettere in tensione una massa) è necessario notare che la corrente di guasto non viene drenata dal sistema disperdente, bensì circola in un anello costituito dal conduttore di fase interessato dal guasto e dal conduttore di protezione.

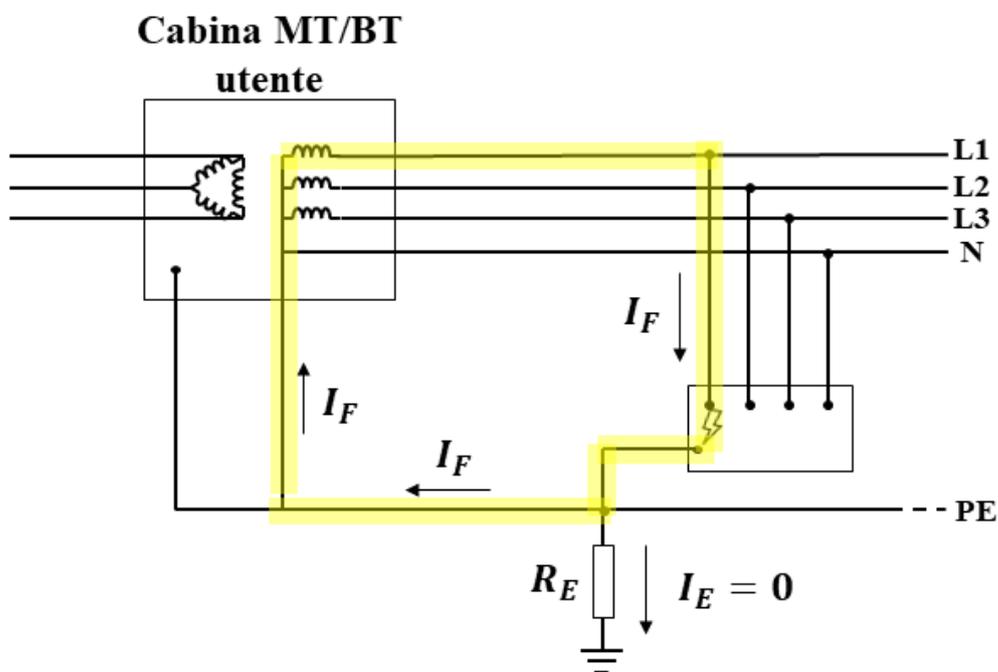


Figura 15: schema del percorso della corrente di guasto [21]

È per questo motivo che non serve lavorare sulla resistenza di terra R_E , ma sulla limitazione dell'impedenza dell'anello di guasto Z_S , data dalla somma delle impedenze in serie del conduttore di fase fino al punto di guasto, dal conduttore di protezione PE e dalla fase del trasformatore, affinché venga verificata la seguente condizione:

$$Z_S \leq \frac{U_O}{I_a} \quad (7)$$

dove U_O è la tensione di fase e I_a è la corrente d'intervento del dispositivo di protezione. Secondo la norma CEI 64-8 la I_a può assumere diversi valori:

- $I_a=I_{diff}$ nel caso della presenza di un interruttore differenziale;
- I_a =la corrente per cui l'interruttore può intervenire entro 5 secondi, per un circuito di distribuzione o terminale alimentante un apparecchio fisso;
- I_a =la corrente per cui l'interruttore può intervenire entro il tempo convenzionale a seconda della tensione nominale del circuito (ad esempio 0,4 secondi per ambienti ordinari e 0,2 per ambienti particolari a 230 V), per un circuito terminale (CEI 64-8 [24] Art. 413.3.5).

Per i circuiti di distribuzione, spesso si ricorre ad un criterio semplificativo:

$$I_m \approx 10I_n \quad (8)$$

con I_m soglia di intervento magnetico dell'interruttore automatico. Se la (8) è verificata, a maggior ragione lo sarà anche per I_a , viceversa sarà necessario entrare con il rapporto $\frac{U_O}{Z_S}$ in ascisse nella curva corrente tempo e verificare se la I_a supera tale rapporto o meno.

Per il calcolo dell'impedenza ci si mette nella condizione più svantaggiosa, ovvero per ciascuna linea di alimentazione si calcola il rapporto $\frac{U_O}{Z_S}$ scegliendo il percorso più lungo (Z_S maggiore significa un valore di rapporto minore possibile e quindi una probabilità maggiore di non verificare la disuguaglianza $I_{rif} < \frac{U_O}{Z_S}$).

2.1.2 Misura dell'anello di guasto

Come già anticipato, l'impedenza dell'anello di guasto (Z_S), nel caso di un guasto franco a massa in un sistema TN, si calcola sommando le impedenze del conduttore di fase, del conduttore di protezione (PE) e dell'avvolgimento secondario del trasformatore (CEI 64-8/6 art 61.3.6.3 all.B, art B.2; CEI 64-14 2.3.2.3).

L'impedenza si misura con particolari strumenti, chiamati "loop tester" (dall'inglese loop che significa anello): essi sfruttano la caduta di tensione sulla resistenza di prova al loro interno,

in abbinata ad un voltmetro e ad un dispositivo di elaborazione dati. Nell'immagine seguente è possibile osservare le fattezze di questo strumento.



Figura 16: loop tester <https://www.asita.com/prodotto/strumento-integrato-prova-differenziali-e-loop-test-as5025/>

Questa resistenza di prova viene inserita grazie ad un sistema di commutazione elettronica per un tempo che oscilla intorno ai 20 ms per evitare che falsi la misura; gli strumenti di ultima generazione effettuano più misure in rapida sequenza, mediando sui risultati.

Il voltmetro è richiesto che sia molto sensibile per poter apprezzare anche le più piccole variazioni di tensione quando vi è passaggio di stato da vuoto (U_0) a carico (U_P); nelle situazioni in cui l'impedenza sia molto bassa, ovvero in prossimità di trasformatori di elevata potenza (reattanza non trascurabile rispetto alla resistenza), le correnti di prova possono raggiungere alcune centinaia di ampere, affinché la caduta di tensione sia rilevabile.

Va detto che, per limitare gli errori di misura, gli strumenti dovrebbero avere la separazione tra circuito voltmetrico e amperometrico, utilizzando quando possibile lo schema di inserzione a quattro fili; una volta misurate correnti e tensioni, lo strumento ne calcola la resistenza o l'impedenza in Ohm. Per i primi, viene utilizzata questa equazione che sfrutta la differenza algebrica tra i moduli delle tensioni:

$$R_S = \frac{|\underline{U}_0| - |\underline{U}_P|}{|\underline{I}_P|} \quad (9)$$

Per i secondi, utilizzabili in posizioni prossime al trasformatore, vale invece questa relazione che a numeratore presenta la differenza vettoriale tra le tensioni:

$$Z_S = \frac{|\underline{U}_0 - \underline{U}_P|}{|\underline{I}_P|} \quad (10)$$

Si ricorda che nel caso specifico di calcolo nell'ambito di un sistema TN, la risoluzione di questo strumento dovrebbe essere di almeno 0,01 Ω .

Nella figura seguente, è rappresentata graficamente la differenza delle due tensioni nel caso specifico in cui appunto la componente reattiva dell'impedenza non sia trascurabile rispetto a quella resistiva.

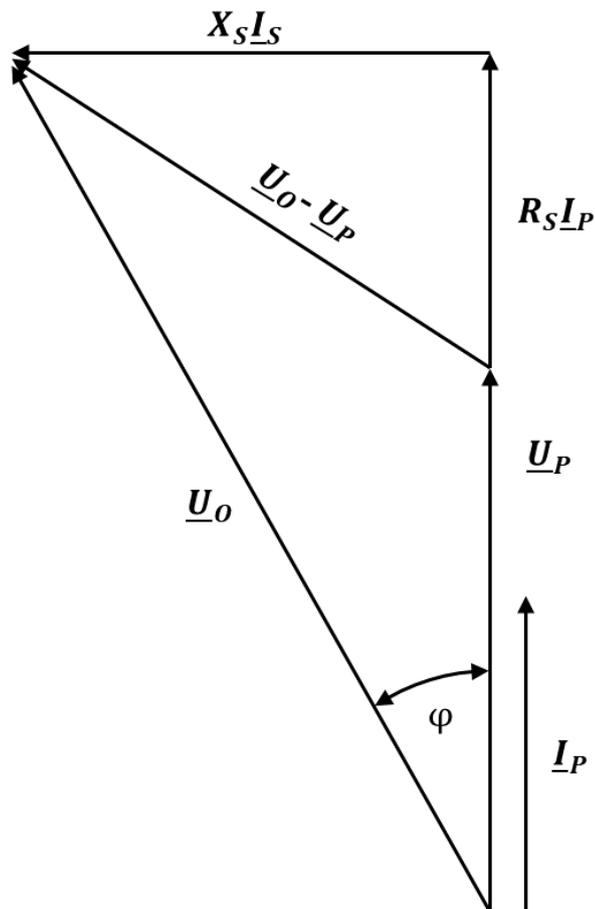


Figura 17: diagramma vettoriale differenza delle tensioni [31]

2.2 Il caso specifico della torre

La torre della regione presenta un impianto elettrico di tipo TN-S: la T indica lo stato del neutro, in questo caso messo a terra, la N indica lo stato delle masse degli utilizzatori, in questo caso collegate allo stesso impianto di terra del centro stella del trasformatore, mentre la S indica che il conduttore di neutro e quello di protezione (PE) sono separati.

Come anticipato nel capitolo 1, possiede la cabina di media tensione di proprietà ed è quindi responsabile dell'atterramento del neutro. In più, deve realizzare un adeguato impianto di messa a terra per l'intervento delle protezioni del distributore in caso di guasto lato media tensione e per la fornitura di un punto a potenziale nullo per le utenze in bassa tensione: in questo caso specifico i due impianti di terra sono riuniti in uno solo, nel rispetto della regola generale per cui conviene unirli e non separarli, ai fini dell'equipotenzialità.

A livello gerarchico la struttura dell'impianto di messa a terra, lato bassa tensione, da monte a valle, è così organizzato:

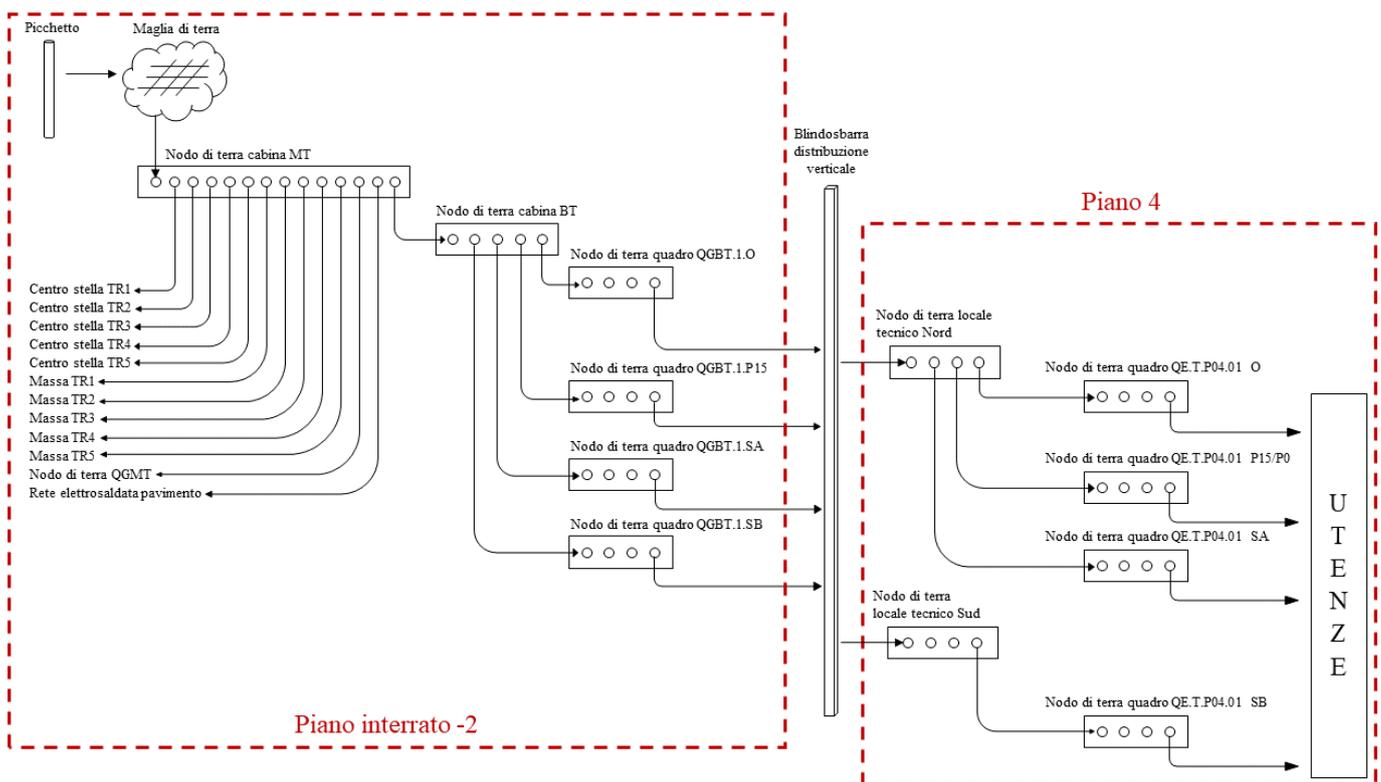


Figura 18: schema impianto di terra del complesso "Torre" (elaborazione personale)

- maglia di terra (dispersore): costituita da una corda di rame nuda immersa nelle fondamenta, di sezione 50 mm². L'interconnessione tra le varie corde avviene con speciali morsetti con corpo in rame e bulloneria in acciaio inox, dimensioni 60x60x3 mm, adatto per collegamenti di tondi Φ 10-24 mm, funi e corde fino a 78 mm²; l'interconnessione tra la maglia e i tondini dell'armatura avviene invece con morsetti di fissaggio con corpo e bulloneria in acciaio inox, adatto per collegamenti di tondi Φ 8-10 mm, per conduttori di sezione 16-50 mm².

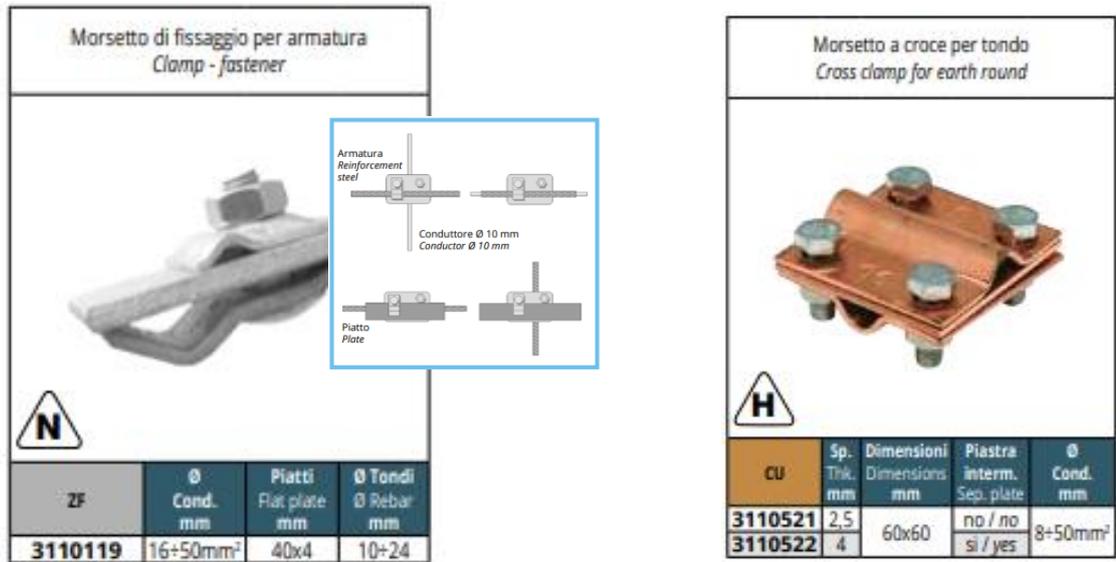


Figura 19: tipi di morsetti di interconnessione



Figura 20: corda di rame

- dispersore tubolare (picchetto): presente ai fini ispettivi, alloggiato in un pozzetto di cemento apribile e connesso alla maglia. Realizzato in acciaio zincato a caldo, lungo 1,5 metri;



Figura 22: pozzetto in calcestruzzo



Figura 21: picchetto

- collettore di terra MT: presente nella cabina MT della torre, è costituito da una barra in rame fissata al muro tramite isolatori. Presenta dei fori a cui imbullonare i capicorda a morsetto in ottone nichelato; ad esso afferiscono i centri stella dei cinque trasformatori e le loro masse, gli involucri delle celle di media tensione e la rete di protezione dei trasformatori. Da questo, il sistema continua verso il locale di bassa tensione;

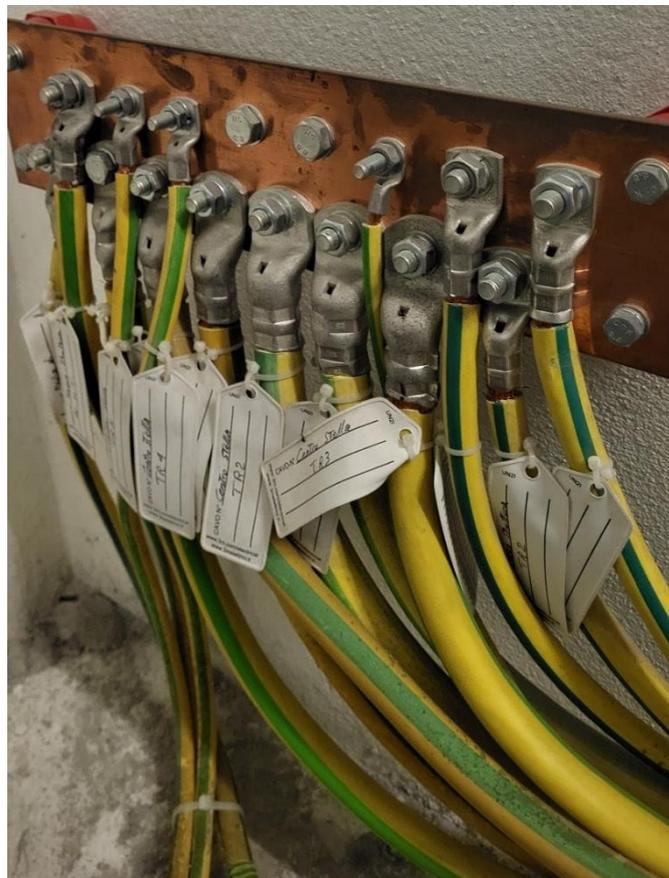


Figura 23: collettore cabina MT (scatto personale del 29/04/2024)

- collettore di terra BT: presente nel locale quadro BT, ad esso si collegano tutti i quadri generali di bassa tensione presenti in questo luogo. Possiede dei fori a cui imbullonare i capicorda a morsetto in ottone nichelato; da questo, il sistema prosegue con la connessione alla distribuzione verticale, attraverso il conduttore dedicato nella blindosbarra per la distribuzione verticale tra i piani;

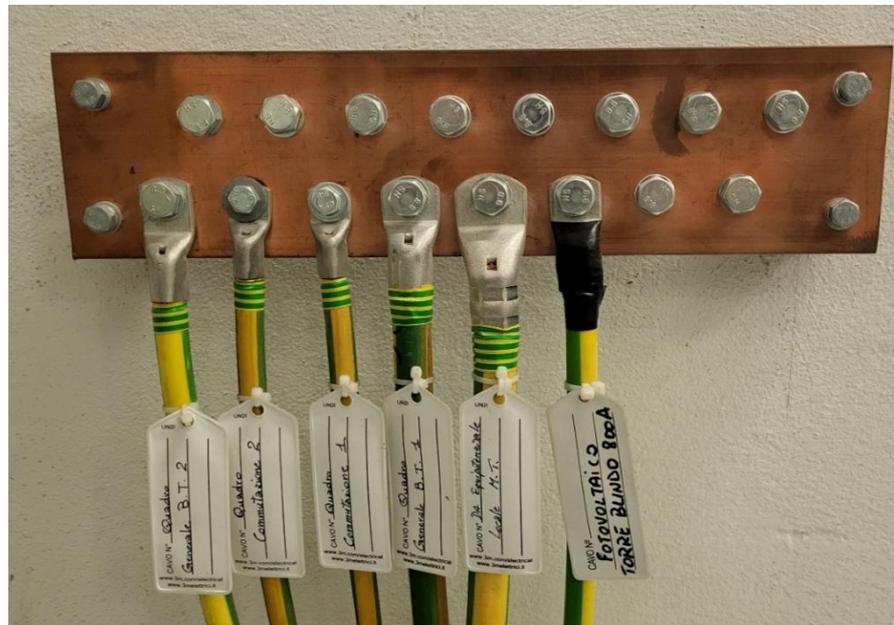


Figura 24: collettore locale BT (scatto personale del 29/04/2024)

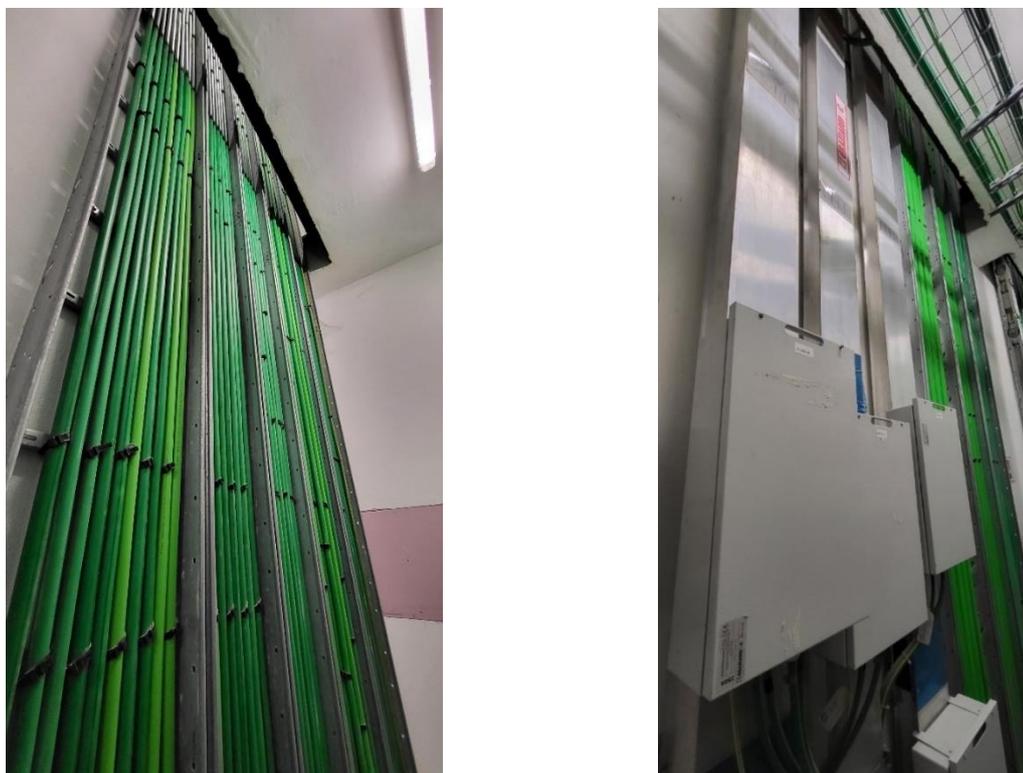


Figura 25: distribuzione verticale in cavo (a sinistra) e blindosbarra (a destra) (scatto personale del 29/04/2024)

- collettore di terra del locale tecnico LT: al piano 04 esistono due locali tecnici, uno a Nord e l'altro a Sud e di conseguenza altrettanti collettori. Il sistema si connette a quello a Nord considerandolo il principale di piano, mentre quello a Sud è come se fosse la sua estensione, figurando ad un livello inferiore nella scala gerarchica e nel browser dei sistemi. Ognuno di essi si connette ai quadri presenti nel proprio locale di appartenenza;



*Figura 26: collettore locale tecnico Nord
(scatto personale del 29/04/2024)*

- ciascun quadro contiene a sua volta un collettore di terra a cui sono collegati tutti i conduttori di protezione di ciascun circuito afferente; da questo punto il circuito prosegue in cavo tripolare connettendosi alle varie utenze terminali. Nell'immagine a fianco si intravede la barratura in rame costituente il nodo equipotenziale in questione. Si ricorda che, nel caso specifico delle torrette o comunque dei dispositivi che necessitano di diverse tipologie di alimentazione, i circuiti di messa a terra sono separati.

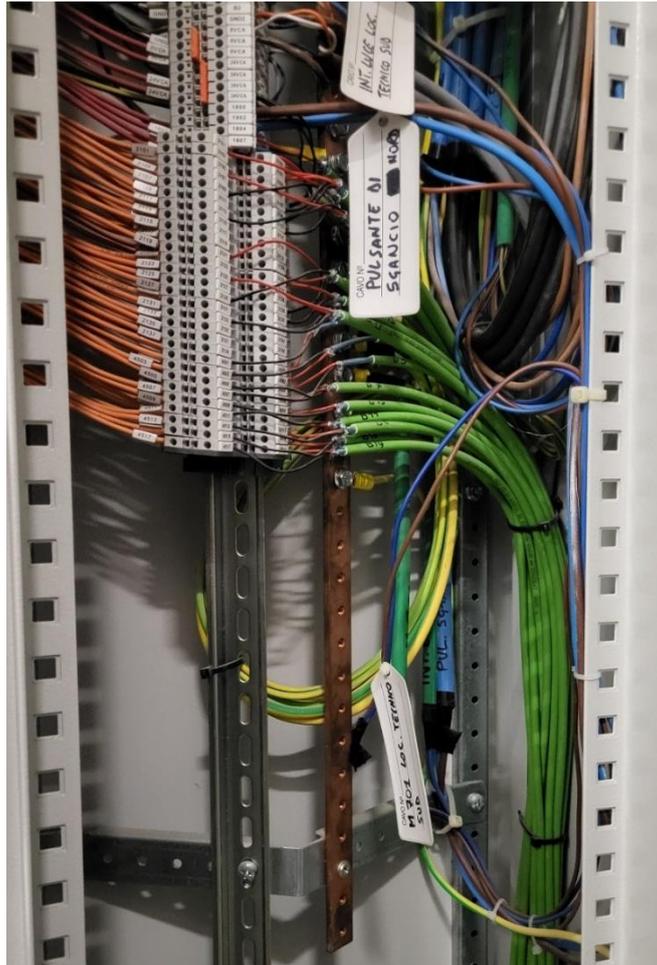


Figura 27: sbarra equipotenziale quadro elettrico (sbarra in rame) (scatto personale del 29/04/2024)

3. Metodologia “As Built”

3.1 Piattaforma di sviluppo

Come è stato detto nel primo capitolo, i formati di progetto non vincolano il progettista ad un software unico e predefinito. Per continuare sul solco tracciato da chi ci ha precedentemente lavorato e per evitare perdite di dati nel passaggio ad una versione più recente, si è preferito lavorare con il programma di “Bim mentoring” Autodesk Revit 2019. Ciò ha comportato alcune rinunce e limitazioni, risolte talvolta con escamotage, per giungere comunque ad un risultato soddisfacente: dette mancanze e difetti, verranno sottolineati e ripresi successivamente in sede specifica.

Senza entrare nello specifico, visto che non è il fine ultimo di questa trattazione, si descrive brevemente l’interfaccia utente del programma, in special modo quelle utilizzate per la restituzione dell’impianto di terra:

1. In alto si trova la barra degli strumenti che, come in tanti altri programmi, racchiude al suo interno i pulsanti per ogni comando associato ad una specifica azione;
2. in centro, si trova la vista di progetto, dove fisicamente è possibile modificare il modello e vedere graficamente la sua realizzazione;
3. sulla destra i vari browser, rispettivamente “di progetto”, nel quale vi è l’elenco completo delle viste tra cui scegliere e “di sistema”, un diagramma ad albero relativamente al sistema selezionato (nel nostro caso l’elettrico);
4. sulla sinistra, la finestra delle proprietà con i vari campi modificabili

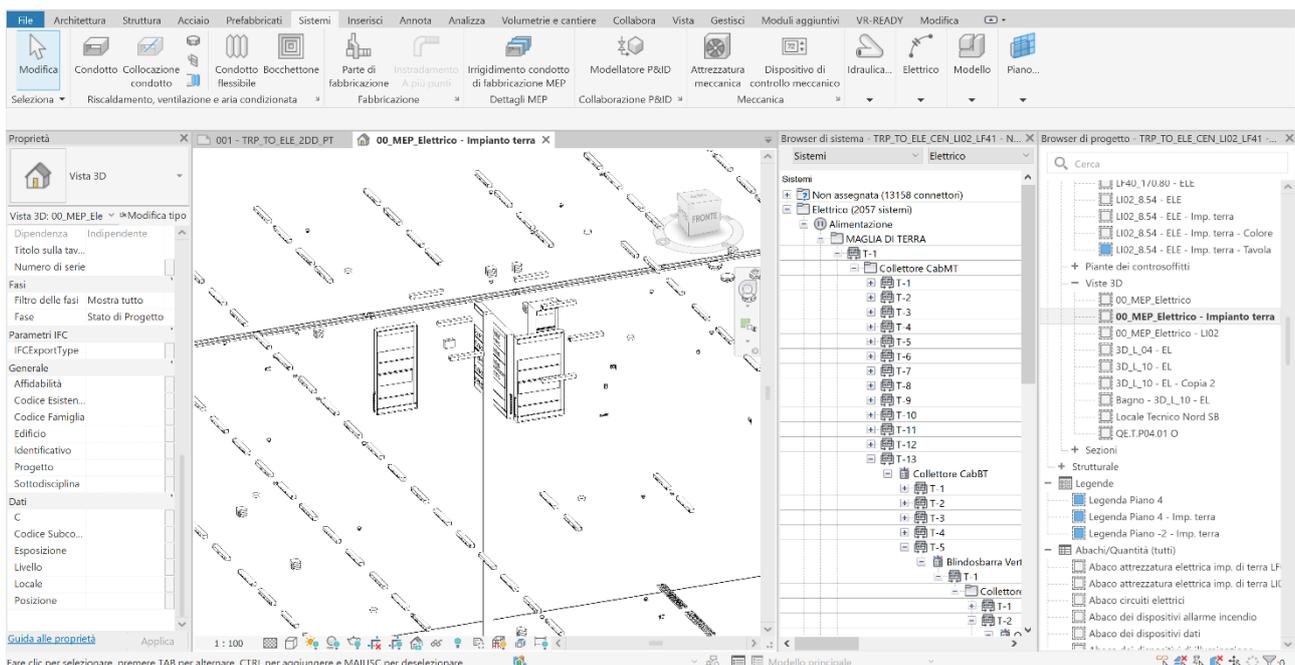


Figura 28: panoramica dell'interfaccia di Revit 2019

Le viste sono raggruppate in discipline: in questo caso si è deciso di visualizzare soltanto l'architettonico, l'elettrico, lo strutturale e il coordinamento, essenziale per il controllo delle interferenze tra i tre mondi: le altre discipline non sono state inserite per mantenere l'identità di progetto federato.

In ciascuna di esse si possono trovare a loro volta diverse sezioni, tra le quali viste in pianta (pavimenti e controsoffitti), sezioni e viste tridimensionali: il vero vantaggio di usare un programma di modellazione 3D come Revit è quello che le modifiche effettuate in ciascuna vista vengono ripecchiate nelle altre, essendo il modello uno unico.

3.2 Modellazione componenti

Il primo passo che è stato affrontato nella restituzione del modello dell'impianto di terra presente all'interno della torre è stato quello della modellazione dei componenti relativi a questo sistema. Per fare ciò si è analizzato attentamente il modello aggiornato, frutto del lavoro di precedenti studenti che si sono occupati di modellare l'impianto elettrico nella sua interezza, dai quadri di piano, alle torrette negli uffici, dalle blindosbarre sotto il pavimento alle passerelle e ai corpi illuminanti. Tutto ciò è stato inoltre collegato a livello di sistema, con la funzione "creazione circuiti".

Per realizzare i componenti presenti e quelli nuovi relativi all'impianto di terra, si è tenuta presente la struttura gerarchica degli oggetti all'interno di Revit rappresentata nella seguente immagine:



Figura 29: schema della gerarchia degli oggetti (realizzazione personale)

Le discipline sono ad esempio la strutturale, l'architettónica o l'impiantistica (elettrica nel nostro caso). Le categorie sono il sottogruppo che comprende elementi accomunati da determinate caratteristiche; nel nostro caso, le categorie della disciplina elettrica sono:

- Apparecchi elettrici
- Apparecchi per illuminazione
- Attrezzatura elettrica
- Cavi
- Dispositivi allarme incendio
- Dispositivi chiamata infermiera
- Dispositivi dati
- Dispositivi di comunicazione
- Dispositivi di illuminazione
- Dispositivi di sicurezza
- Dispositivi telefono
- Passerelle
- Raccordi passerella
- Raccordi tubo protettivo
- Tubi protettivi

Esse sono predefinite all'interno del software.

Scendendo di un livello, si trovano le famiglie, uno speciale raggruppamento che comprende tutti gli oggetti che possiedono una rappresentazione grafica simile (ad esempio la famiglia collettore di terra); va tenuto presente che esistono due tipi di famiglie, ovvero quelle di sistema, presenti di default in Revit che possono essere solo modificate e non cancellate, e quelle caricabili, ovvero create appositamente per il progetto nell'Editor dedicato, salvabili ed esportabili in altri progetti.

All'interno di ciascuna famiglia, vi è la divisione in tipi, basati ad esempio sulle dimensioni o sul numero dei posti (nel caso del collettore di terra, a 4,8,12 o 16 posti).

L'ultimo livello è quello dedicato alle istanze, ovvero i singoli oggetti inseriti nel progetto, ciascuno individuabile univocamente.

Fatta questa premessa, ci si è concentrati sul rilevamento degli elementi dell'impianto elettrico già modellati, valutando se e come siano stati collegati all'impianto di terra dell'edificio.

Come descritto nel capitolo precedente, l'impianto di messa a terra si estende dalle fondamenta all'utenza finale, il che ha reso inevitabile una commistione tra due file di progetto diversi, ovvero quello della torre che ospita il quarto piano (il piano tipo oggetto dello studio) e quello della corte interrata, ospitando la cabina elettrica della torre, punto nevralgico assieme al locale quadro generale BT. Una soluzione per ovviare a ciò ci sarebbe stata, ovvero quella di

utilizzare il codice padre per rendere interconnessi i due file lasciandoli separati, ma questo non avrebbe reso possibile la visualizzazione complessiva tramite i colori del filtro di vista, né tantomeno una successiva visualizzazione completa una volta esportato nel mondo virtuale.

Nelle due immagini seguenti sono riportate, nell'ordine, una porzione della vista 3D della cabina di trasformazione con a fianco il locale quadro elettrico generale BT e, poco più sotto, una panoramica generale del piano 4 sempre in 3D:

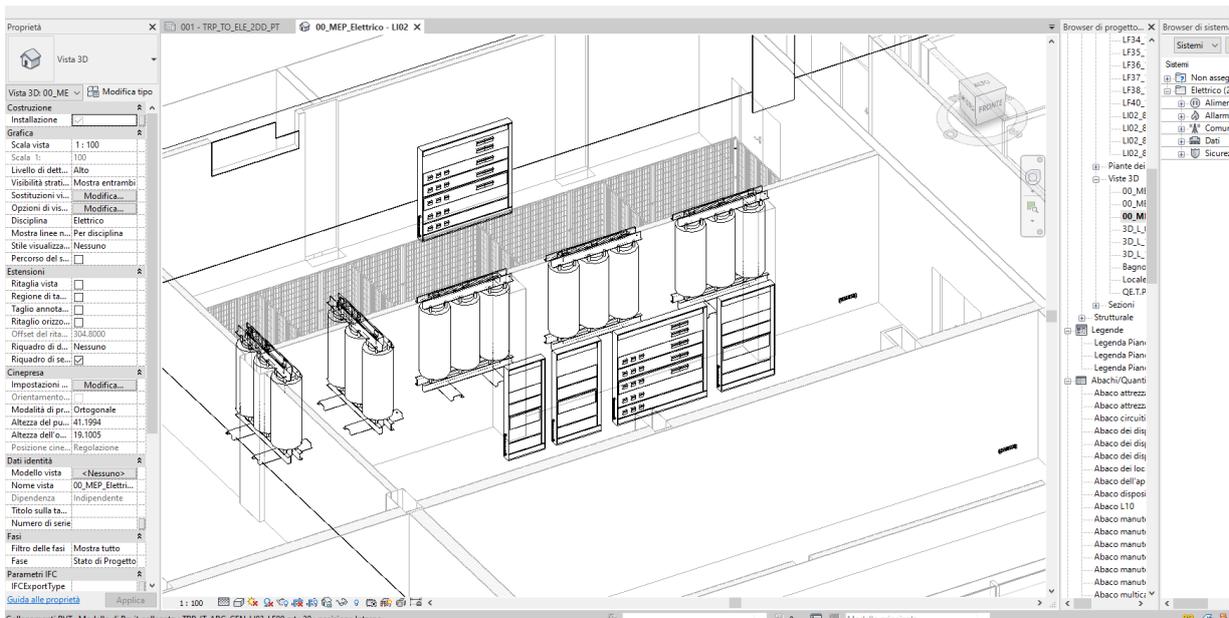


Figura 31: modello della cabina MT e locale BT

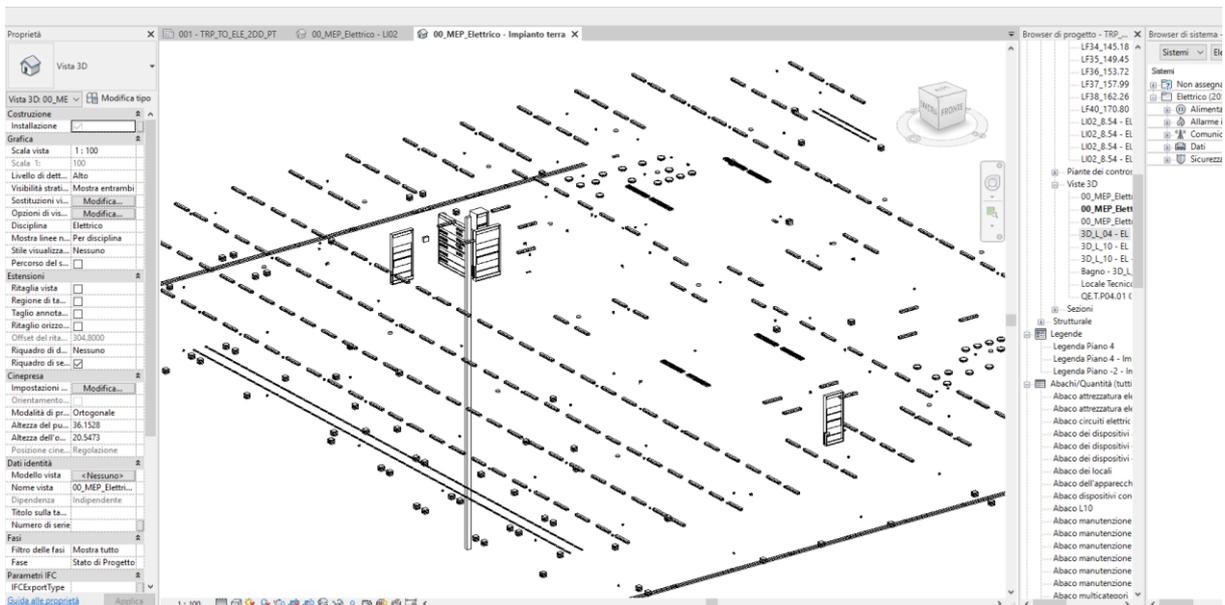


Figura 30: modello del piano tipo

I paragrafi che seguono trattano nello specifico ogni singolo elemento, modellato dal nuovo o modificato all'occorrenza, appartenente all'impianto di terra, che sia un semplice utilizzatore o un componente integrante. La presentazione seguirà una struttura gerarchica, da monte a valle.

3.2.1 Maglia di terra

Il primo componente modellato come famiglia caricabile è stato la maglia di terra: è stata realizzata tramite un'estrusione su percorso utilizzando come "base" il file .dwg 2D fornito dall'ufficio tecnico della torre. In esso è raffigurata la disposizione della corda di terra al secondo livello interrato e le sue relative informazioni alfanumeriche. Essendo un disegno in pianta realizzato in fase di progettazione e non più aggiornato dopo la realizzazione, sono presenti anche elementi che, in sede di sopralluogo, si sono rivelati installati in posti differenti: il modello realizzato su Revit ha corretto queste discrepanze. La famiglia non presenta un simbolo specifico o una rappresentazione differente a seconda del livello di dettaglio, pertanto risulta inutile una scheda LOD dedicata. Seguono due immagini relative alla realizzazione:

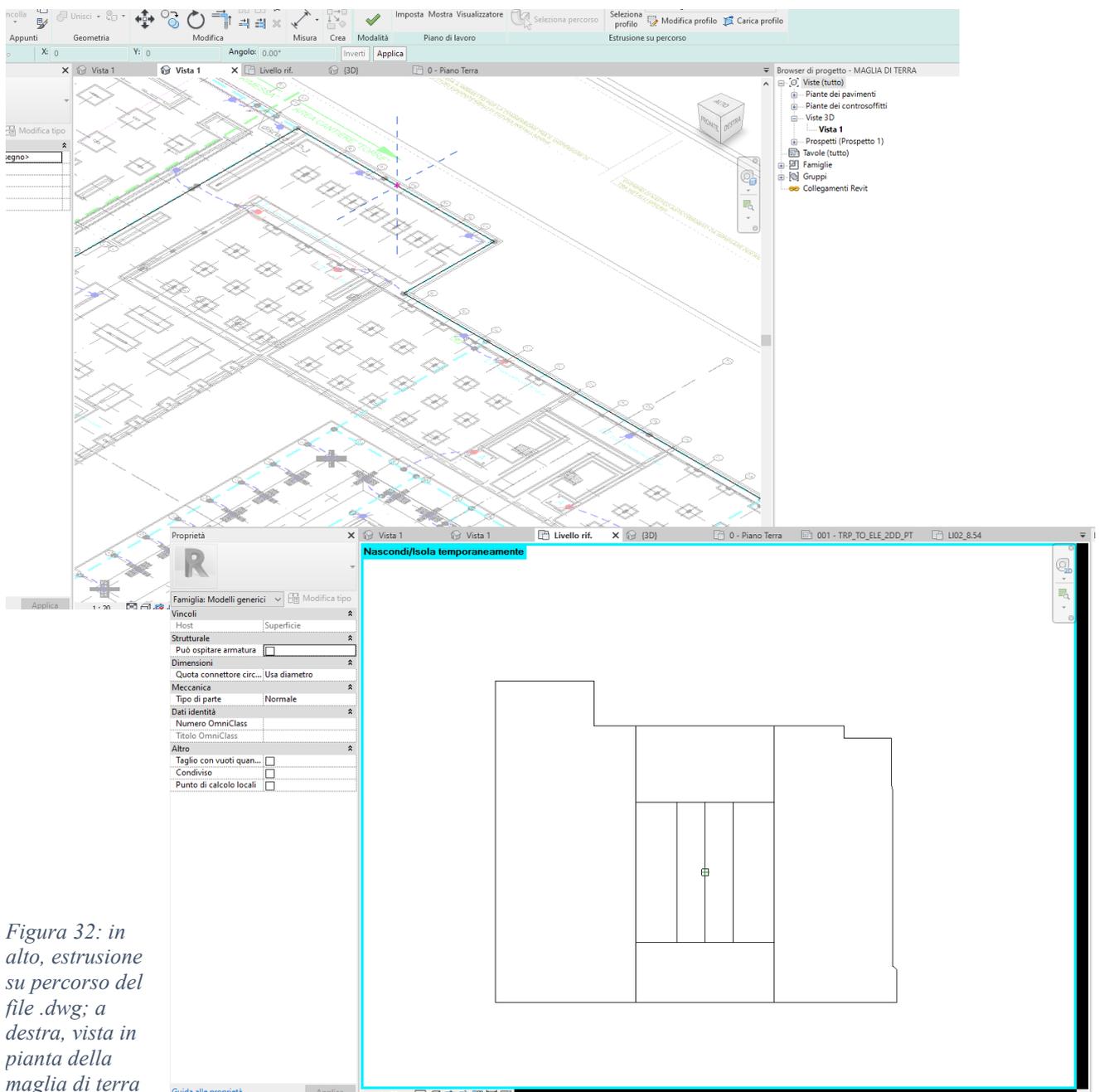


Figura 32: in alto, estrusione su percorso del file .dwg; a destra, vista in pianta della maglia di terra

3.2.2 Picchetto

Un elemento non cruciale in questo tipo di struttura di impianto di messa a terra ma essenziale per tutte le attività di monitoraggio e manutenzione nel tempo è sicuramente il picchetto. In strutture più semplici e in presenza di un'estensione ridotta del terreno, è utilizzato come unico elemento disperdente. Nel caso in esame è di tipo tubolare, in acciaio zincato a caldo e lungo 1,5 m: è immerso nel terreno e contenuto in un pozzetto per ispezione di dimensioni 60x60x90 cm completo di chiusino in ghisa ed è localizzato nel locale cabina MT della torre.

È stato realizzato come famiglia caricabile, nell'Editor famiglie, e questa è la sua scheda LOD:

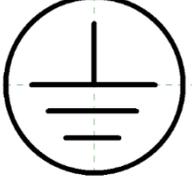
PICCHETTO	Rappresentazione	LoG	LoI
LOD A		rappresentazione grafica attraverso un sistema geometrico simbolico	caratteristiche quantitative e qualitative indicative
LOD B		entità virtualizzate graficamente come un sistema geometrico generico o una geometria d'ingombro	caratteristiche qualitative e quantitative approssimate
LOD D		entità virtualizzate graficamente come un sistema geometrico dettagliato	caratteristiche qualitative e quantitative specifiche di una pluralità definita di prodotti similari

Tabella 1: scheda LOD picchetto

3.2.3 Morsetto di interconnessione

Un altro componente cardine dell'impianto è senza dubbio il morsetto di interconnessione. Per poter avere due simboli 2D di colore diverso si è optato per la creazione di due famiglie separate:

- morsetto di interconnessione per punti di giunzione tra elementi del dispersore di terra: come parametro peculiare è stata inserita il diametro della corda;
- morsetto di interconnessione per punti di giunzione tra elementi del dispersore di terra e i ferri di armatura: identico al caso precedente con l'aggiunta anche del parametro del diametro del tondino.

Nella tabella seguente, la loro scheda LOD:

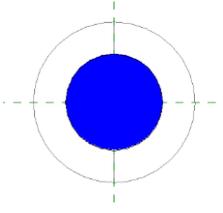
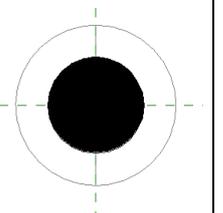
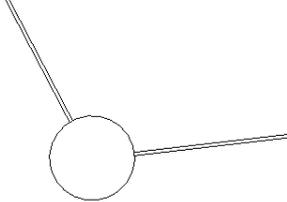
MORSETTI DI INTERCONN.	Rappresentazione		LoG	LoI
	tra elem. dispersore	tra dispersore e armatura		
LOD A			rappresentazione grafica attraverso un sistema geometrico simbolico	caratteristiche quantitative e qualitative indicative
LOD B			entità virtualizzate graficamente come un sistema geometrico generico o una geometria d'ingombro	caratteristiche qualitative e quantitative approssimate

Tabella 2: scheda LOD morsetti di interconnessione

3.2.4 Collettori di terra

I punti nevralgici dell'impianto sono sicuramente i collettori, ovvero piastre di rame a cui vengono imbullonati i capicorda dei conduttori di protezione in arrivo e in partenza. Anche loro sono stati modellati come famiglie caricabili e inizialmente concepite come famiglie ospitanti della famiglia bullone, anch'essa creata appositamente.

La famiglia collettore è una sola, con quattro tipi al suo interno a seconda che si abbiano 4, 8, 10 o 16 posti (fori) per i bulloni: questi ultimi sono stati modellati come famiglia esterna e poi annidata in quella dei collettori. Questo ha permesso la realizzazione di una famiglia parametrica, che permettesse di adattare il numero di bulloni e la loro posizione in base al tipo scelto.

Ciascun bullone nella realtà è collegabile a due capicorda, uno sul lato anteriore e l'altro sul lato posteriore: nell'ambiente virtuale si è cercato di mantenere questa modalità, ponendo un connettore elettrico su ciascuna delle due facce della vite di serraggio.

Per agevolare il manutentore e riuscire ad individuare univocamente la corrispondenza bullone-circuito di appartenenza, si è pensato in aggiunta di realizzare una famiglia annidata che comprendesse un numero e venisse posizionata sotto a ciascun bullone, cambiando cifra al variare del tipo di collettore.

Nel capitolo relativo ai sistemi, verrà spiegato nel dettaglio la logica sulla scelta della categoria di appartenenza e gli accorgimenti adottati per sopperire alle mancanze del software.

Con questa famiglia sono stati modellati tutti i collettori singoli fissati al muro dei vari locali tecnici e della cabina di trasformazione.

Il simbolo 2D è ispirato a quello della messa a terra generica normata ma colorata di verde e racchiusa da un quadrato, così da distinguerlo dai collettori relativi ai quadri elettrici e ai picchetti, che nelle tavole relative alle piante potrebbero confondersi.

Potendo inserire il simbolo soltanto nella vista in pianta ma avendo l'esigenza di posizionare il collettore in tante direzioni, è stato disegnato un simbolo identico in ogni vista (e associato alla categoria simboli di annotazione) così da non scalare con la pianta ma perdendo di fatto l'opzione di impedirgli di ruotare assieme all'oggetto: è un limite del software, che si è preferito sorvolare invece di creare una famiglia per ogni orientamento che avrebbe solo appesantito il modello e contribuito a far perdere l'essenza di una modellazione in ambiente BIM.

Nella pagina seguente, la scheda LOD relativa:

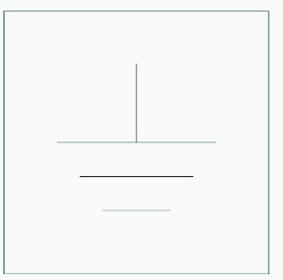
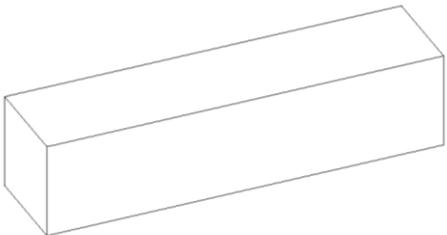
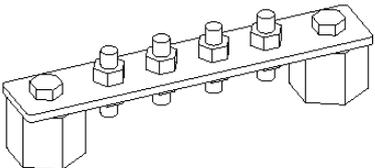
COLLETTORE	Rappresentazione	LoG	LoI
LOD A		<p> rappresentazione grafica attraverso un sistema geometrico simbolico </p>	<p> caratteristiche quantitative e qualitative indicative </p>
LOD B		<p> entità virtualizzate graficamente come un sistema geometrico generico o una geometria d'ingombro </p>	<p> caratteristiche qualitative e quantitative approssimate </p>
LOD D		<p> entità virtualizzate graficamente come un sistema geometrico dettagliato </p>	<p> caratteristiche qualitative e quantitative specifiche di una pluralità definita di prodotti simili </p>

Tabella 3: scheda LOD collettori

3.2.5 Quadro elettrico

Per quanto riguarda i quadri elettrici, ci si è limitati ad adattare il modello realizzato dai precedenti studenti, implementando la presenza di un collettore di terra.

Nella realtà, i quadri elettrici di zona o di piano presentano un nodo equipotenziale al loro interno, costituito da una barra di rame alloggiata nella maggior parte dei casi in un comparto separato dagli organi di manovra e protezione, solitamente nella sezione della morsettiera, con la propria portella chiusa a chiave. Inizialmente si è pensato di inserire la stessa famiglia del collettore di terra, annidata in quella del quadro: purtroppo però è stato necessario abbandonare quella strada sia per un motivo grafico, in quanto all'interno dell'oggetto quadro sarebbe stato impossibile visualizzarlo (sia in Revit che poi in Unity per la visualizzazione grafica), sia per un motivo di collegamento tra sistemi, in quanto la versione 2019 non prevede la possibilità di associare due sistemi di distribuzione elettrica diversi a due oggetti annidati.

Per questo motivo, ci si è orientati sull'opzione di apporre il collettore esternamente, su una parete del quadro, soluzione non corrispondente alla realtà ma ideale in un'ottica di visualizzazione 3D: chi andrà ad operare sul modello, dovrà essere debitamente informato su tale discrepanza.

Per quanto riguarda il simbolo 2D, si è optato per affiancare al simbolo normato del quadro elettrico, quello della messa a terra racchiuso da un cerchio verde: nelle classiche viste in pianta questa soluzione è poco usata, preferendola a didascalie o lunghe descrizioni nella legenda. La soluzione adottata risulta essere invece un richiamo immediato alla realtà, per distinguere in breve tempo se e quale quadro presenta un nodo equipotenziale dedicato.

La scheda LOD relativa è la seguente:

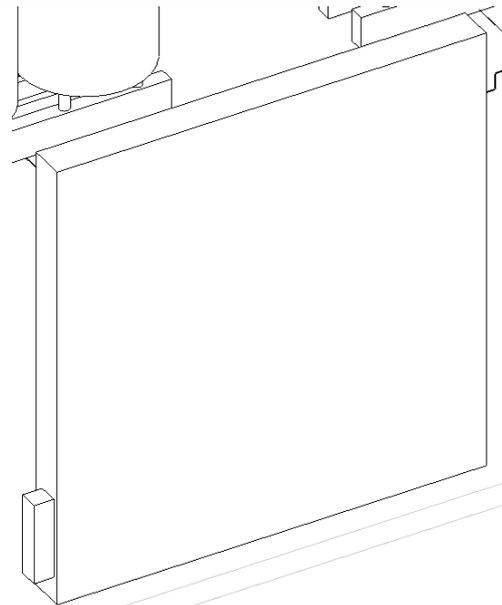
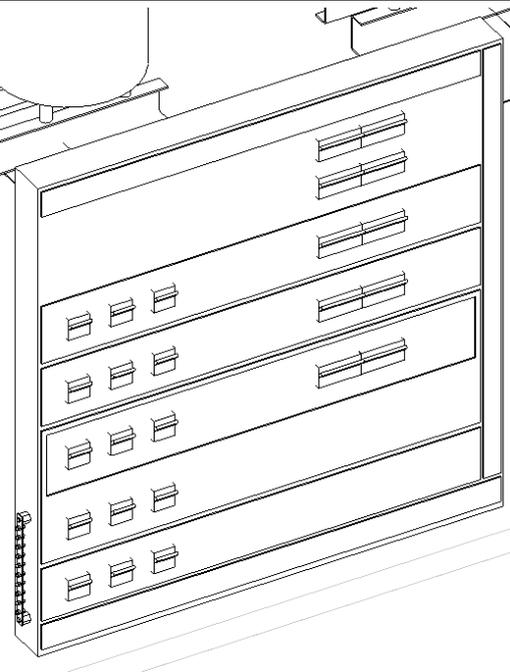
QUADRO ELETTRICO	Rappresentazione	LoG	LoI
LOD A		<p>rappresentazione grafica attraverso un sistema geometrico simbolico</p>	<p>caratteristiche quantitative e qualitative indicative</p>
LOD B		<p>entità virtualizzate graficamente come un sistema geometrico generico o una geometria d'ingombro</p>	<p>caratteristiche qualitative e quantitative approssimate</p>
LOD D		<p>entità virtualizzate graficamente come un sistema geometrico dettagliato</p>	<p>caratteristiche qualitative e quantitative specifiche di una pluralità definita di prodotti similari</p>

Tabella 4: scheda LOD quadri elettrici

3.2.6 Blindosbarre, torrette e corpi illuminanti

Per queste tre tipologie di dispositivi elettrici, ci si è limitati ad adattare i modelli già creati in precedenza da altri studenti. Nello specifico:

- blindosbarre per la distribuzione elettrica orizzontale sotto al pavimento del primo piano: erano già presenti i connettori elettrici per la parte di potenza, ne sono stati aggiunti altrettanti per la messa a terra. Il lavoro è stato compiuto sia sulle blindosbarre della distribuzione ordinaria che su quelle della privilegiata;
- torrette: sono delle famiglie caricabili contenenti quelle delle prese annidate. Avendo due connettori di potenza, uno per ciascuna tipologia di alimentazione, ne sono stati creati altrettanti per la messa a terra. In verità essendo l'impianto di terra unico, ne è stato utilizzato solo uno, mantenendo il secondo per completezza e in previsione di eventuali studi approfonditi futuri;
- corpi illuminanti: tutte le famiglie presenti dei corpi illuminanti sono stati provvisti di un connettore elettrico per la messa a terra, senza il raddoppio poiché ciascuna lampada viene alimentata da una sola tipologia di alimentazione. I circuiti di comando luce non sono interessati dalla messa a terra bensì solo l'elemento finale, proprio come nella realtà.

Una menzione speciale spetta alla blindosbarra per la distribuzione elettrica tra i piani, non modellata dagli studenti precedenti: essa è stata solo abbozzata ai fini della visualizzazione e dell'integrazione nel sistema di messa a terra.

3.3 Modellazione sistemi elettrici

3.3.1 Browser dei sistemi

Uno dei passi più importanti da eseguire per proseguire nell'analisi, è sicuramente quello della modellazione dell'impianto di messa a terra attraverso una delle famiglie integrate in Revit, ovvero quella dei Sistemi elettrici: questo procedimento è essenziale poiché permette di connettere a livello informativo tutti le componenti peculiari di ciascun impianto e poter così effettuare controlli, gestioni e rilevamenti.

Considerando il livello di dettaglio stabilito a priori, i conduttori di alimentazione non sono stati modellati fisicamente, in quanto farlo non avrebbe comportato un miglioramento significativo del modello.

Sulla scia di quanto fatto per la parte di alimentazione, si è fatto lo stesso per la parte del conduttore di protezione, sfruttando però la possibilità di nominare i circuiti tramite prefisso

separato da un trattino, in modo tale che la numerazione automatica di Revit associasse lo stesso numero tra il circuito di “andata” e di “ritorno” senza però causare un’omonimia che avrebbe generato ambiguità tra i sistemi.

Per fornire un esempio, i circuiti di alimentazione afferenti alla torretta con Revit ID 5378045 presentano rispettivamente il nominativo “O-3” e “P-3” (a seconda che l’alimentazione sia di tipo ordinario o privilegiato): di conseguenza si è deciso di utilizzare come prefisso per i circuiti di terra terminali della stessa torretta “T term O-3” e “T term P-3”, mentre per quelli intermedi il semplice prefisso T.

Per applicare il prefisso a determinati circuiti è necessario operare sull’apposita opzione posta tra le proprietà del quadro “padre” subito a monte.

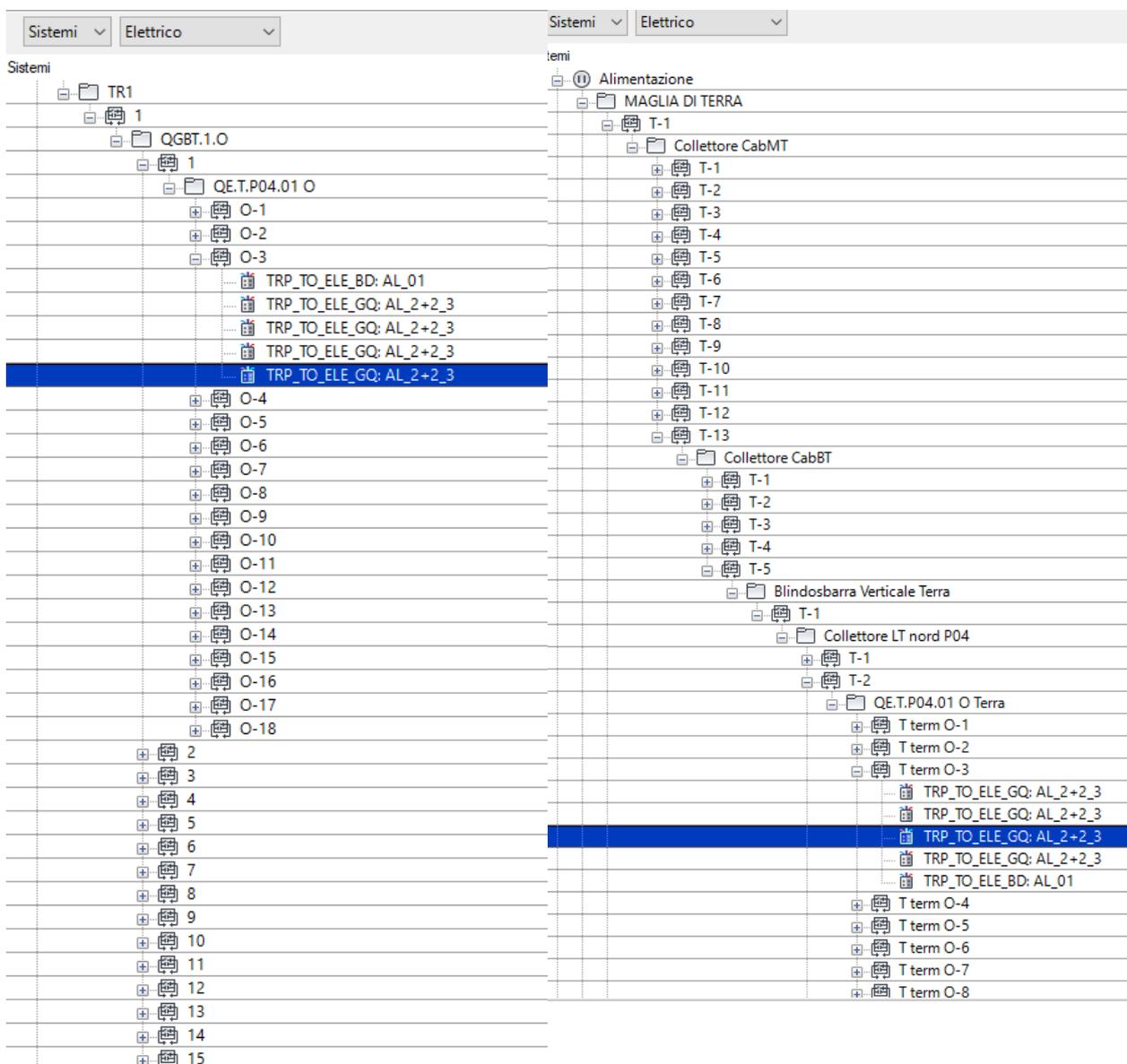


Figura 33: browser di sistema del circuito di alimentazione O-3 (a sinistra) e del suo circuito di terra (a destra)

L'ordine gerarchico è esattamente quello descritto nel capitolo precedente: ogni collettore di terra si comporta come un quadro elettrico del sistema di alimentazione, da cui hanno origine i vari circuiti discendenti verso gli elementi terminali dei circuiti. Per questa analogia con la parte di alimentazione, si è scelto per l'appunto di inquadrare tutti i collettori nella famiglia dei quadri elettrici, anche per sopperire ad una limitazione della versione 2019 di Revit che non ammette due sistemi di alimentazione diversi in termini di valore di tensione nominale coesistenti all'interno dello stesso pannello elettrico.

Ecco perché i collettori dei singoli quadri sono stati posti su un fianco dei vari quadri elettrici e non al loro interno, discostando il modello dalla realtà dei fatti.

Nel tab gestisci, le impostazioni elettriche sono state teatro di alcuni cambiamenti, allo scopo di associare all'impianto di messa a terra un suo sistema specifico: la prima modifica è stata quella di creare, nella sezione "Definizioni tensione", un nuovo valore di tensione chiamato semplicemente "0 V". Al valore è stato associato 0.00 V, a quello minimo 0.00 V mentre a quello massimo 230.00 V, così da comprendere sia il caso di normale funzionamento (tensione nulla) che quello di guasto monofase a terra (tensione massima).

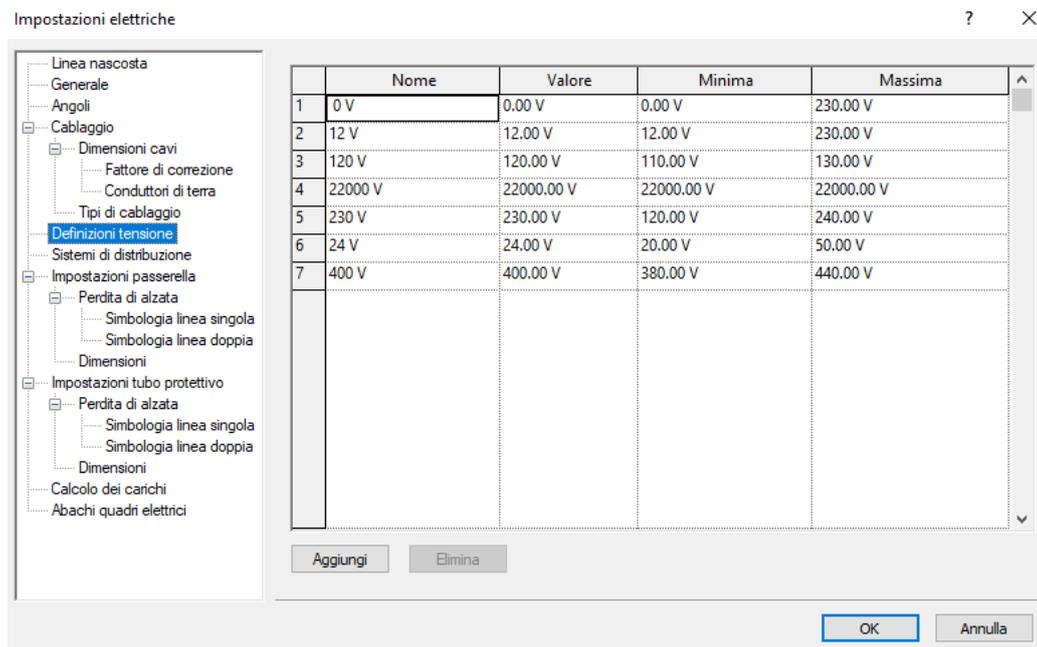


Figura 34: definizioni tensione (impostazioni elettriche)

Nelle stesse impostazioni, forti della nuova definizione di tensione, ci si è focalizzati sulla sezione “Sistemi di distribuzione”: qui, ne è stato definito uno nuovo denominato “Terra”, con tensione fase-terra di 0 V. Si segnala che, essendo quella della messa terra un’applicazione inusuale per un software come Revit che non è specifico per gli impianti elettrici, non è stato possibile assegnare un solo cavo nel campo relativo a questi ultimi.

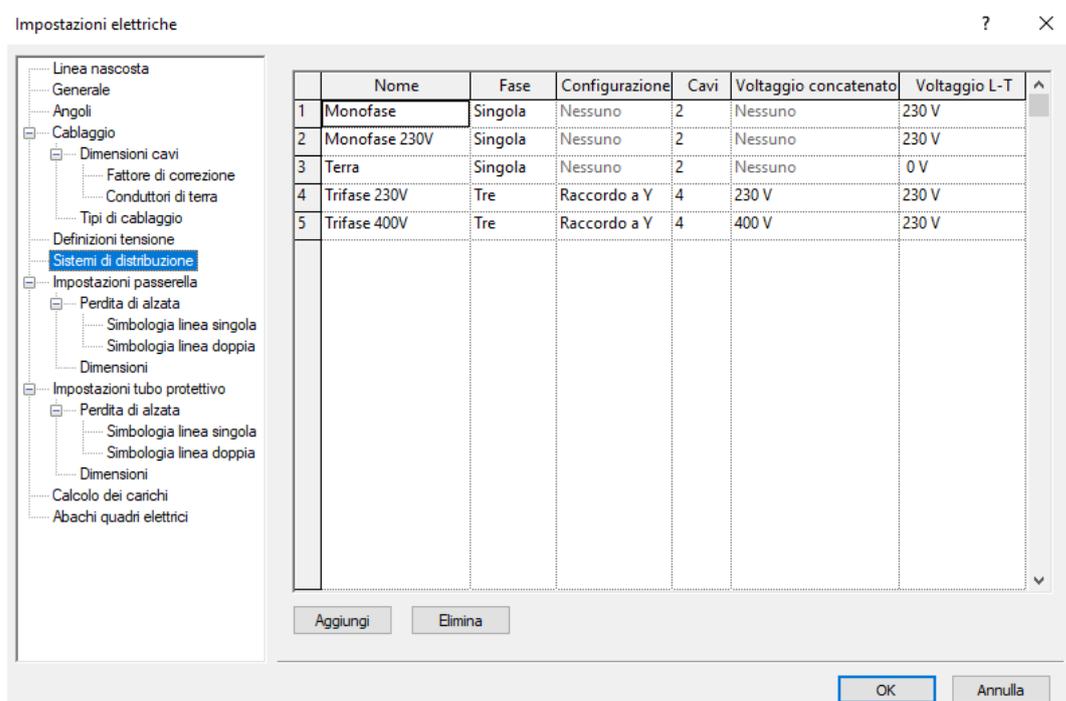


Figura 35: sistemi di distribuzione (impostazioni elettriche)

Grazie a queste modifiche, ciascun elemento connesso all’impianto di terra, tramite la modellazione delle singole famiglie, è stato provvisto di uno o più connettori elettrici (in aggiunta a quelli già presenti per l’alimentazione) e a cui è stato associato il valore 0 V. In questa maniera Revit ha riconosciuto ciascun elemento come compatibile con il sistema elettrico “Terra” creato precedentemente, facendolo comparire tra i sistemi di distribuzione disponibili nell’apposito menù a tendina:

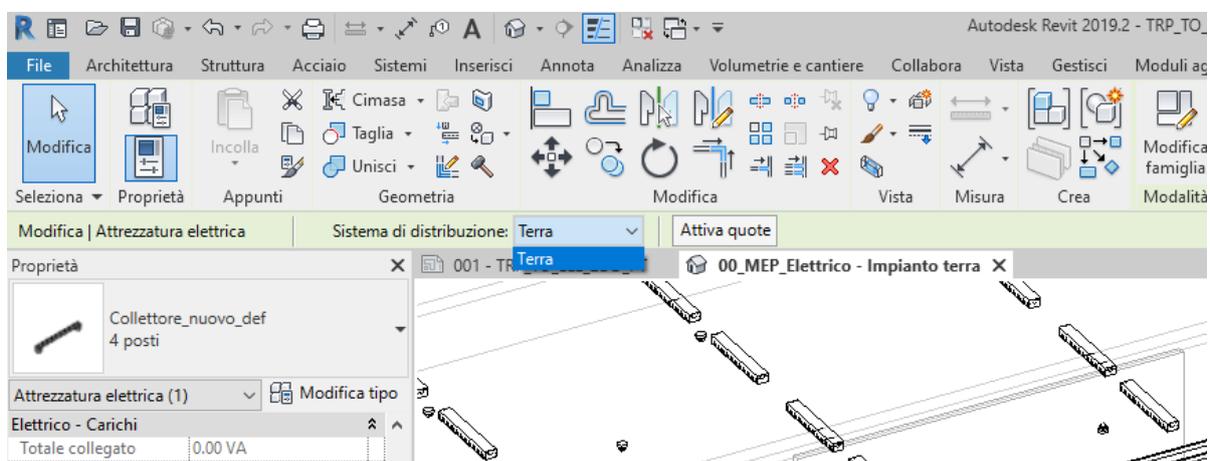


Figura 36: scelta sistema di distribuzione per il collettore

3.3.2 Parametro di appartenenza impianto di terra

Con l'intenzione di rendere modificabile l'appartenenza di ciascuna istanza del mondo elettrico, si è pensato di creare un parametro condiviso di tipo Sì/No, chiamato "Connesso all'impianto di terra": in questo modo, inserito come parametro di progetto nel modello, oltre a poter essere usato come discriminare per la creazione di abachi, diventa il primo criterio di selezione all'interno dei filtri di vista.

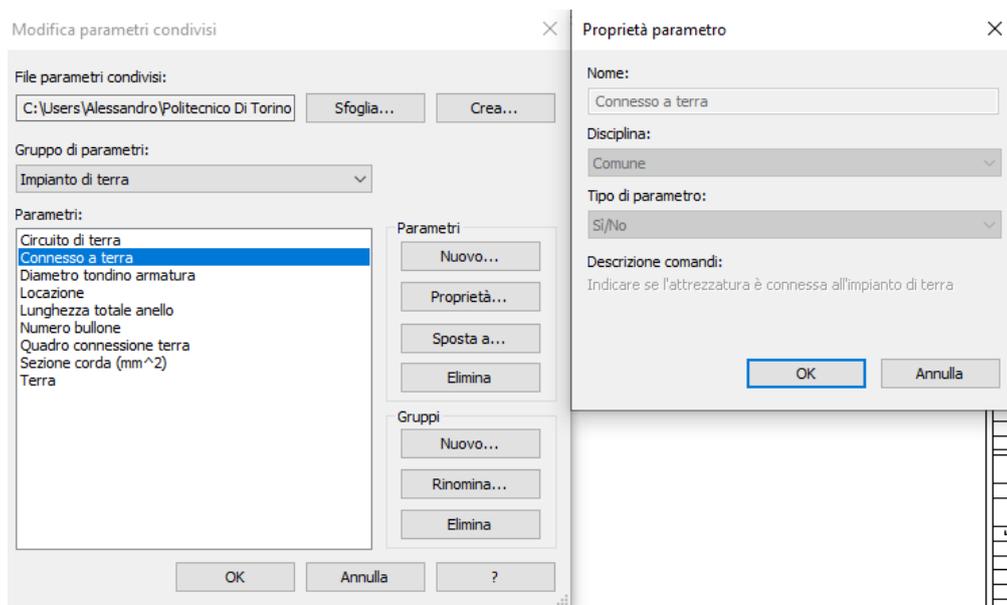


Figura 37: creazione parametro condiviso "Connesso a terra"

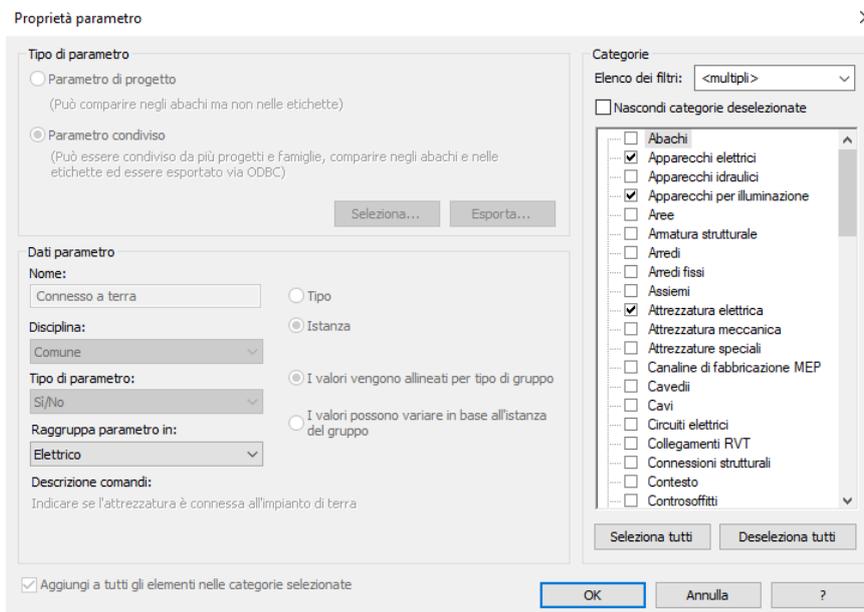


Figura 38: associazione parametro di progetto "Connesso a terra"

3.3.3 Parametro circuito di terra

Un altro parametro creato appositamente per i filtri di vista che verranno trattati nel capitolo successivo, risulta essere quello di tipo condiviso chiamato “Circuito di terra”: si tratta di un parametro di progetto di tipo testo, compilabile singolarmente per ciascuna istanza e in cui va riportato il nome del circuito di terra di appartenenza.

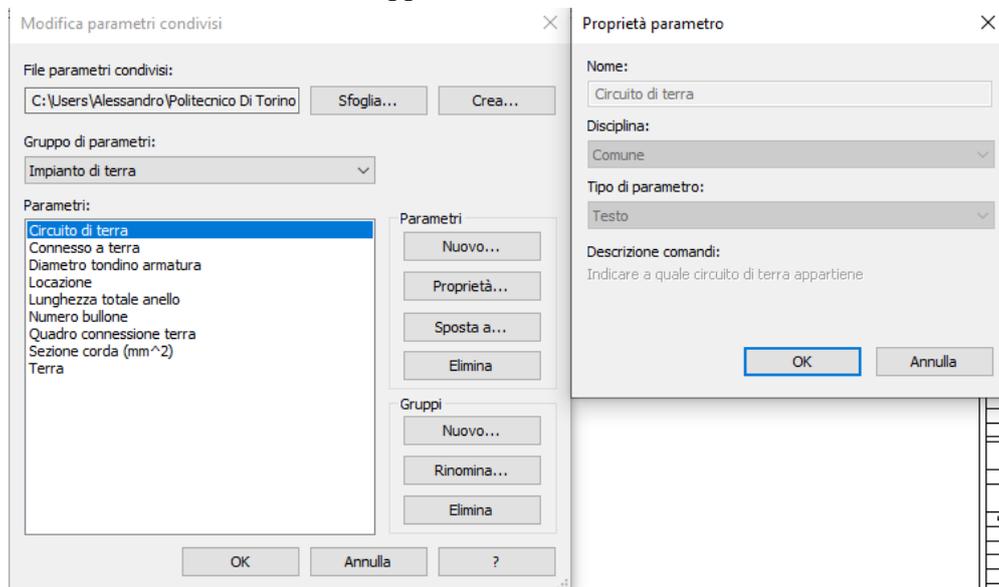


Figura 39: creazione parametro condiviso "Circuito di terra"

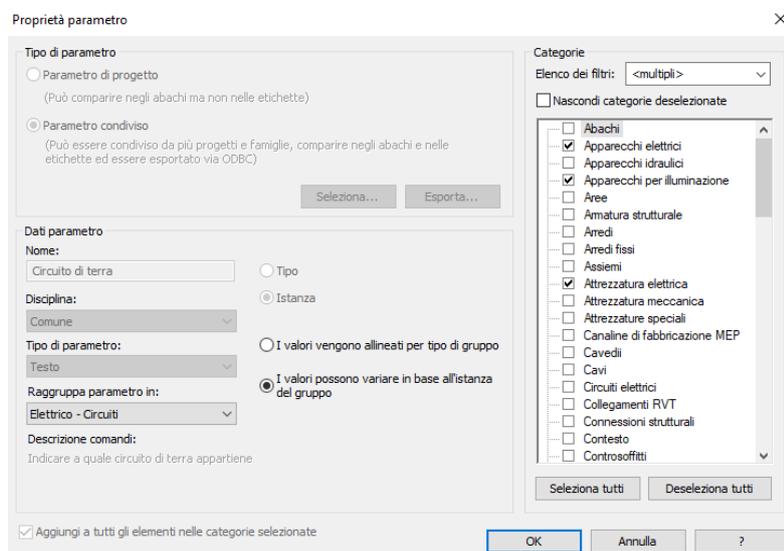


Figura 40: associazione parametro di progetto "Circuito di terra"

Sono stati quindi compilati i campi relativi a due circuiti di funzione differenti, uno per l'alimentazione di un gruppo di torrette e l'altro per una serie di lampade: l'intento è stato quello di esaminare due casi tipo, delineando una strategia realizzativa e lasciando la possibilità di estendere il ragionamento a tutti gli altri sistemi elettrici. È il caso quindi di “T term O-1” (illuminazione) e “T term O-3” (alimentazione delle torrette), che riprendono tale e quale la nomenclatura con prefissi ideata per identificare i circuiti elettrici nel browser dei sistemi.

3.3.4 Filtri di vista

I filtri di vista sono solo il primo tentativo di visualizzazione dell'impianto modellato e delle relative istanze all'interno di Revit: per lo scopo di questa trattazione sono stati creati dei filtri associati a più categorie, essendo il parametro relativo alla connessione a terra di tipo comune.

Sono stati quindi realizzati due filtri per i circuiti scelti nella sezione precedente, chiamati "Flag T term O-1" e "Flag T term O-3": la parola "flag" richiama il fatto che il principale criterio discriminante sia un parametro sì/no da spuntare, mentre "T term O-..." richiama la funzione e il numero del circuito a cui il filtro è associato; le categorie prescelte sono state quelle degli apparecchi elettrici, degli apparecchi per illuminazione e dell'attrezzatura elettrica.

Le regole di filtraggio sono pressochè identiche e si differenziano soltanto per il nome del circuito di terra: la struttura prevede un controllo primario se l'elemento sia collegato o meno all'impianto di terra (condizione di tipo "AND") e una serie annidata di condizioni di tipo "OR". Queste ultime condizioni sono complementari alla prima ed è sufficiente che ne sia vera una per attivare il filtro: si controlla il parametro del circuito di terra per evidenziare la parte terminale del circuito (condizione "uguale"), in alternativa si verifica il parametro "Identificativo", già presente nel modello al momento della presa in carico per questo progetto e contenente il nome in linguaggio comune dato ad ogni istanza dagli utenti. Si fa notare che con la condizione "contiene" si comprendono tutte i collettori dei quadri interessati dal collegamento in base alla tipologia di alimentazione, mentre con la condizione "uguale" si scremano solo le istanze che hanno proprio quell'identificativo in quanto si è consapevoli che la parte a monte di ogni circuito è in comune con tutti gli altri.

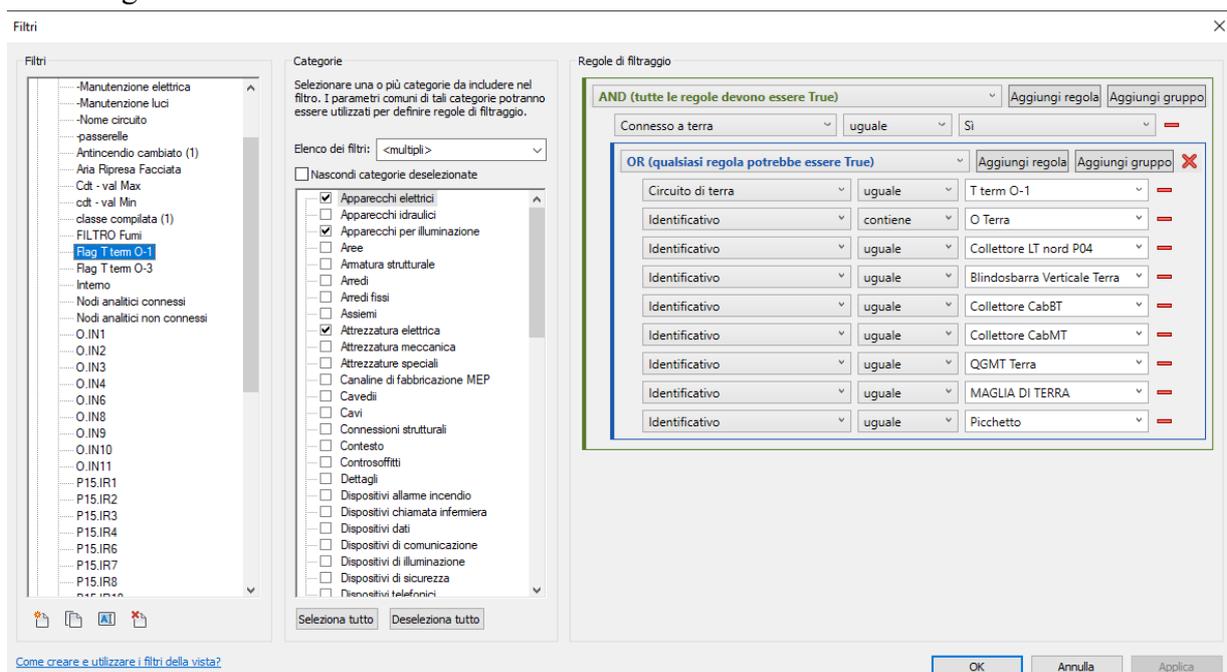


Figura 41: creazione regola filtro T term O-1

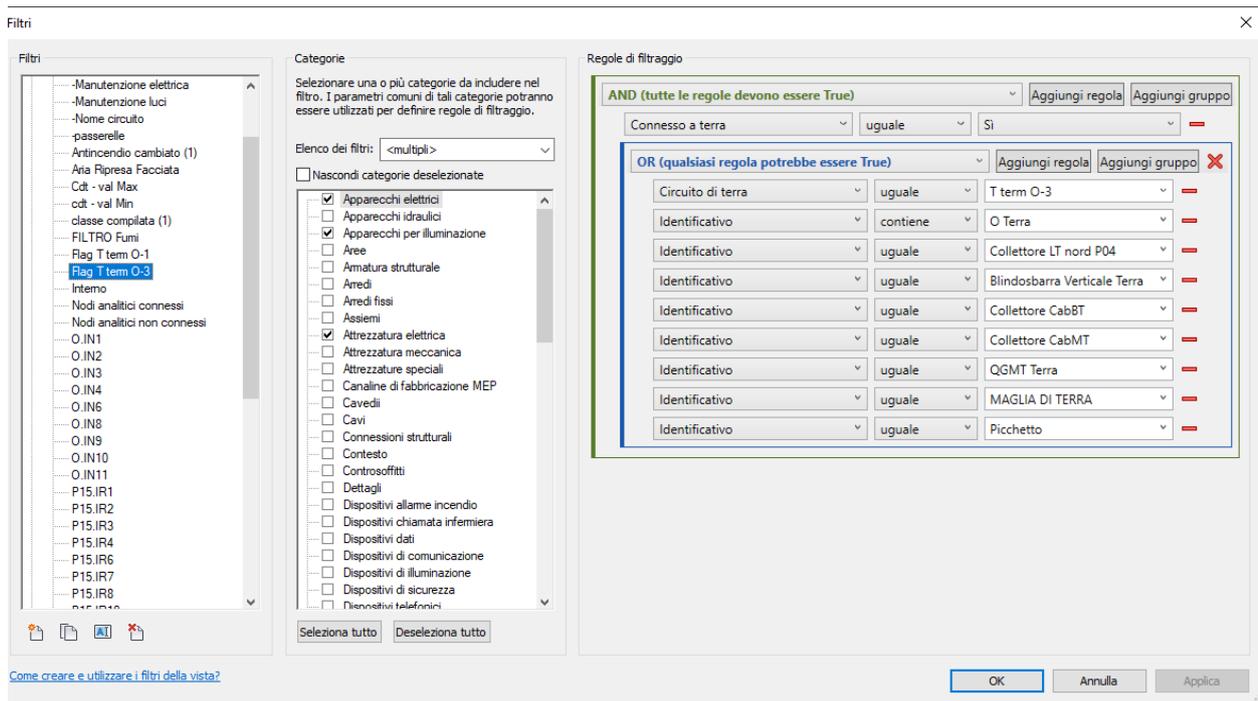


Figura 42: creazione regola filtro T term O-3

Questa metodologia potrebbe sembrare un po' macchinosa ma possiede un ottimo tasso di scalabilità per tutti gli altri circuiti del modello, potendo copiare il filtro e cambiando solo la seconda voce relativa al nome del "Circuito di terra".

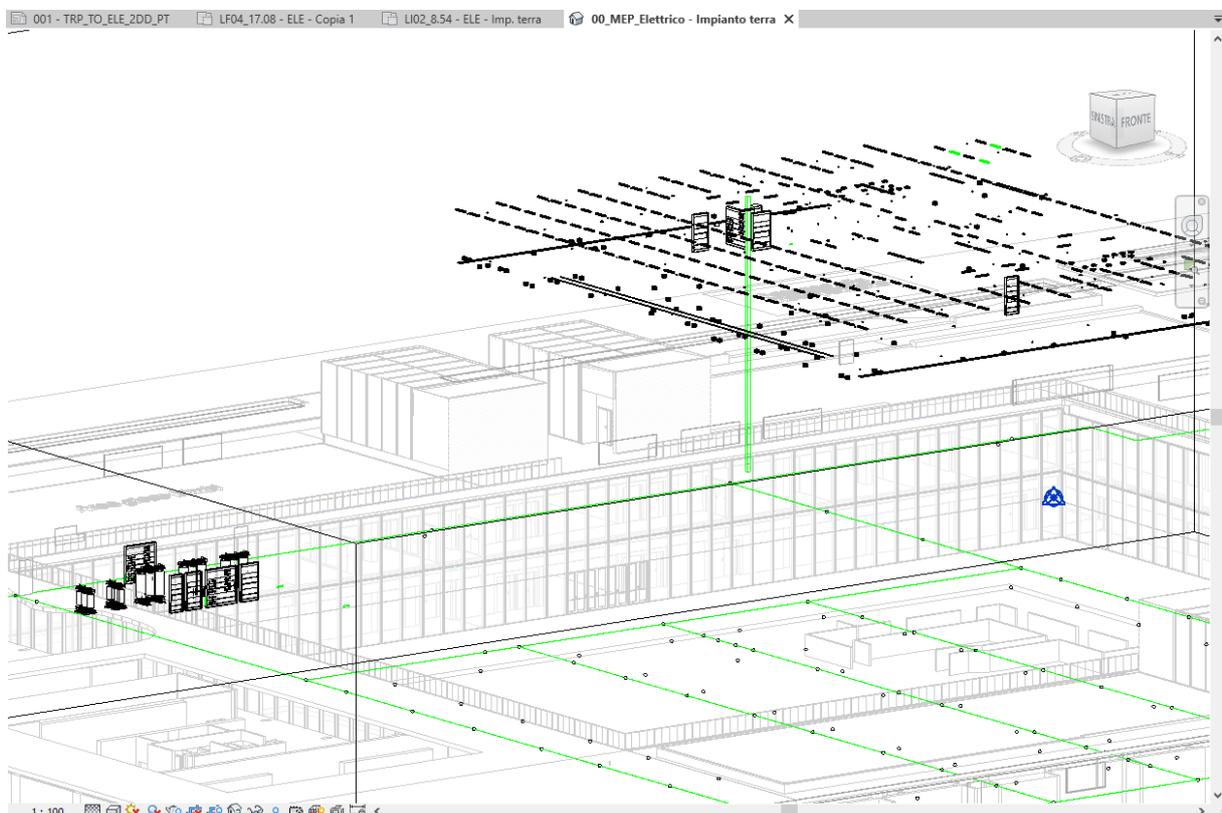


Figura 43: visualizzazione filtro T term O-1

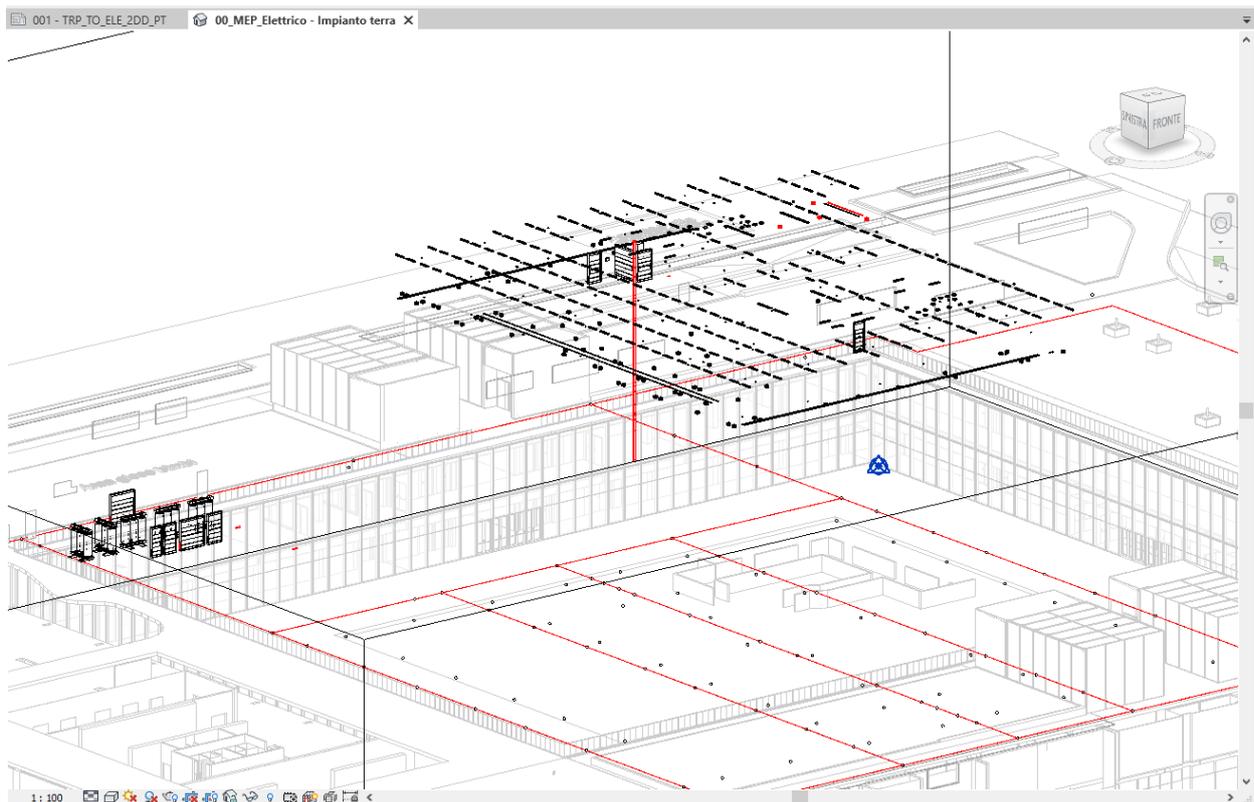


Figura 44: visualizzazione filtro T term O-3

3.3.5 Abachi

Uno strumento molto utile offerto dal software Revit è anche quello relativo agli abachi, ovvero elenchi di raccolta di istanze monocategoria o pluricategoria, filtrate in base ad uno o più parametri e organizzati secondo regole definite dall'utente. Nel caso di questa trattazione, è stato deciso di raccogliere in unico abaco tutte le istanze connesse all'impianto di terra, suddividendole per piano e indicando livello, locale e posizione per una facile individuazione, oltre al nome del circuito di terra di appartenenza (il nome del circuito è visualizzato solo per i due casi tipo considerati per i filtri, gli altri possiedono per forza di cose un campo vuoto). L'abaco in versione integrale è disponibile all'Appendice 1.

4. Gestione e manutenzione

4.1 Generalità

Come anticipato nel capitolo 1, la sesta dimensione (6D) della metodologia BIM è proprio quella dedicata al “Facility Management” (di qui in avanti abbreviato con FM), ovvero alla disciplina che si occupa della gestione operativa degli edifici per tutta la loro vita utile. L'integrazione con il BIM permette di ottimizzare la manutenzione, il funzionamento e il monitoraggio delle strutture attraverso un modello digitale che contiene informazioni dettagliate su ogni aspetto del fabbricato, incluse strutture, impianti, e spazi interni.

Con ciò, il Facility Manager ha accesso a un archivio completo e aggiornato di dati sugli asset della costruzione in questione: questi dati includono specifiche tecniche, documentazione, stato di manutenzione e altre informazioni essenziali che migliorano la gestione operativa.

Si può quindi ottimizzare la manutenzione, monitorando lo stato e il ciclo di vita di ogni componente e pianificando la manutenzione preventiva e predittiva in modo più accurato.

Tenendo traccia delle condizioni e delle esigenze di manutenzione degli asset nel BIM, si è in grado di stimare e controllare meglio i costi operativi; come già accennato, il modello può anche fornire informazioni su materiali e componenti che devono essere sostituiti o aggiornati.

Un altro vantaggio ad esempio può essere quello di analizzare l'uso dello spazio in modo dettagliato, ottimizzando la distribuzione degli ambienti in base alle esigenze operative: il Facility Manager può studiare infatti i flussi di movimento all'interno dell'edificio per migliorare la funzionalità degli spazi.

In ultimo, non per importanza, figura senza dubbio la possibilità di monitorare e gestire i consumi energetici: si simulano interventi migliorativi, come l'installazione di sistemi energetici più efficienti o l'adozione di pratiche più sostenibili, e si verifica l'impatto sull'efficienza complessiva.

Il Facility Manager, per lavorare sempre al pieno delle sue possibilità, necessita di agire su un modello aggiornato affinché rappresenti il cosiddetto “As-built” (tradotto “così come è costruito”) cioè la versione finale del progetto con eventuali modifiche apportate durante la costruzione.

Ad assisterlo in questo compito, vi sono una pletera di software specializzati per il facility management che, sfruttando i formati di file aperti già citati in precedenza, permettono una perfetta interoperabilità tra i programmi di BIM authoring e quelli di FM. Grazie a script e plugin (come Dynamo per Revit) poi, è possibile automatizzare processi di aggiornamento del modello, inserimento di nuovi dati o segnalazione di eventi critici (es. allarmi di manutenzione).

In aggiunta va detto che sensori e dispositivi IoT possono essere collegati al modello BIM per monitorare in tempo reale dati critici (temperatura, umidità, consumi, ecc.): questi dati, qualora combinati con il BIM, permettono una gestione smart dell'edificio, facilitando decisioni rapide e mirate.

Affinchè vi sia una piena compatibilità e completezza dei dati tra il modello BIM e i software di FM, si richiede una buona standardizzazione dei processi e dei formati di dati: questo si raggiunge con un aggiornamento costante dei dati, specialmente quando vengono apportate modifiche strutturali o di manutenzione nel tempo.

Non va inoltre dimenticato che è necessario un cambiamento culturale e tecnico nelle pratiche di gestione dell'edificio e una formazione adeguata del personale per sfruttare al meglio il sistema.

4.2 Verifica degli impianti di terra

La verifica e la manutenzione degli impianti elettrici (e nello specifico quelli di terra) è regolamentata da tempo da testi unici, decreti legislativi, guide e decreti ministeriali: non si può pensare alla progettazione di tali impianti senza un'adeguata conoscenza e rispetto di tali norme.

Nel caso specifico degli impianti di terra vi sono due capisaldi imprescindibili, ovvero il *Decreto Legislativo 9 aprile 2008, n. 81 (D.Lgs. 81/08)*, noto come Testo Unico sulla Sicurezza sul Lavoro [25], e il *Decreto del Presidente della Repubblica 22 ottobre 2001, n. 462 (D.P.R. 462/2001)* [26], che disciplina le procedure di verifica degli impianti elettrici di messa a terra e di protezione contro le scariche atmosferiche.

Il primo decreto è la legge regina che regola tutti gli aspetti di sicurezza che ruotano attorno all'ambiente di lavoro: riguarda la salute e la sicurezza dei lavoratori in tutti i settori di attività, sia pubblici che privati, valutando tutti i tipi di rischio e offrendo una panoramica sulle attrezzature, macchinari, impianti e dispositivi utilizzati.

Il secondo si applica esclusivamente agli impianti elettrici di messa a terra, ai dispositivi di protezione contro le scariche atmosferiche e agli impianti elettrici in luoghi con pericolo di esplosione, disciplinando le procedure di denuncia, verifica e omologazione di questi specifici impianti nei luoghi di lavoro.

In entrambi i decreti si parla di “obblighi del datore di lavoro”, il primo responsabile a capo di una struttura gerarchica che può essere un luogo di lavoro in cui è sufficiente vi sia un dipendente. Tra questi obblighi si annoverano:

- nel D.Lgs. 81/08:

viene richiesta una manutenzione generale, con speciale riferimento ai seguenti articoli

- Art. 15: Il datore di lavoro deve adottare misure per il miglioramento continuo della sicurezza, compresa la manutenzione preventiva;
- Art. 64: Obbligo di mantenere i luoghi di lavoro, le attrezzature, i dispositivi di sicurezza e gli impianti in condizioni efficienti attraverso una manutenzione adeguata;
- Art. 71: Obbligo di garantire che le attrezzature di lavoro siano installate correttamente, utilizzate in modo sicuro e sottoposte a idonea manutenzione per preservarne l'efficienza e la sicurezza.

È necessario redigere un registro di manutenzione, ovvero di un elenco dei controlli e delle manutenzioni effettuate, quando previsto dalla normativa o quando necessario in base alla valutazione dei rischi.

Viene anche introdotta la differenza tra manutenzione ordinaria e straordinaria, elencando le attività preventive e correttive per mantenere o ripristinare l'efficienza e la sicurezza delle attrezzature e degli impianti. I controlli possono essere eseguiti da personale interno qualificato o da tecnici esterni, e non necessariamente da organismi abilitati.

- nel D.P.R. 462/2001:

è compito dell'impresa installatrice rilasciare una dichiarazione di conformità o omologazione alla messa in servizio dell'impianto, che verrà inviata dal datore di lavoro all'ASL/Arpa e Inail entro 30 giorni dalla consegna dei documenti. Vengono invece richieste delle verifiche periodiche, con speciale riferimento ai seguenti articoli:

- Art. 4: Il datore di lavoro è tenuto a far sottoporre gli impianti a verifiche periodiche:
 - Ogni 2 anni: impianti in cantieri, locali ad uso medico, ambienti a maggior rischio in caso di incendio e luoghi con pericolo di esplosione;
 - Ogni 5 anni: altri impianti.Deve richiedere tali verifiche ad ASL/ARPA o ad organismi abilitati dal Ministero e conservare i documenti relativi per 4 anni.

Sarà invece l'Inail a vigilare e ad effettuare i dovuti controlli a campione sugli impianti dell'azienda.

Già nel Decreto Legislativo 12 aprile 2006, n. 163 [27] ("Codice dei contratti pubblici di lavori, servizi e forniture") venivano previste disposizioni riguardanti i piani di manutenzione, documenti fondamentali nel processo di progettazione ed esecuzione di opere pubbliche.

Un primo elemento fondamentale nella redazione di un piano di manutenzione è *l'elenco degli impianti e dei componenti* da sottoporre a manutenzione: questo documento offre una panoramica completa degli elementi su cui intervenire.

La *scheda anagrafica dei componenti* contiene invece informazioni dettagliate su ciascun componente, come l'anno di costruzione, il costruttore e le principali caratteristiche elettriche: tale scheda garantisce una tracciabilità accurata e una conoscenza approfondita dei componenti stessi.

La *scheda di manutenzione* specifica l'insieme delle operazioni da svolgere su ogni componente o impianto, che possono includere prove, misurazioni o ispezioni; per facilitare la gestione, ad ogni tipo di operazione viene associato un codice di intervento e viene indicata una frequenza temporale che stabilisce quando eseguire tali operazioni: questo permette di creare un calendario degli interventi, essenziale per la pianificazione della manutenzione preventiva.

Un ulteriore documento cruciale è il *registro degli interventi*, dove vengono annotati tutti gli interventi eseguiti sull'impianto elettrico o sui singoli componenti: questo registro funge da storico delle attività di manutenzione, offrendo un riferimento costante sullo stato dell'impianto.

Le norme di sicurezza per l'esecuzione della manutenzione sono altrettanto importanti: infatti, basandosi sulle norme tecniche e sulle leggi vigenti, forniscono indicazioni sulle tecniche e le procedure da adottare durante le attività manutentive, oltre a specificare le attrezzature e i dispositivi di protezione individuale (DPI) da utilizzare. Ciò garantisce la sicurezza del personale coinvolto nelle operazioni.

Oltre ai documenti specifici per la manutenzione, è indispensabile disporre poi della documentazione di impianto, che riporta le caratteristiche tecniche e funzionali dell'impianto, generalmente prodotta in fase di progettazione. Questa documentazione comprende:

- *documenti di disposizione funzionale*: come schemi a blocchi e schemi dei circuiti;
- *documenti di disposizione topografica*: come planimetrie e disegni dei quadri elettrici;
- *documenti di connessione*: dettagli sulle connessioni tra i vari componenti;
- *documenti di messa in servizio, funzionamento ed esercizio*: istruzioni operative per l'utilizzo corretto dell'impianto.

Questi documenti sono fondamentali sia per acquisire una conoscenza approfondita dell'impianto, consentendo interventi efficaci e sicuri, sia per poterli aggiornare in caso di modifiche o interventi che ne alterino la configurazione originale.

Nel settore dei lavori pubblici, la redazione di un piano di manutenzione è obbligatoria, come stabilito dal D.Lgs. 163/06, allegato XXI, articolo 24. Il piano di manutenzione deve includere i seguenti documenti:

1. *manuale d'uso*: destinato all'utente finale, fornisce informazioni sul corretto utilizzo dell'impianto e sui rischi associati a usi impropri o guasti: include istruzioni per piccoli interventi di manutenzione eseguibili da personale non specializzato e indicazioni su come

riconoscere anomalie che richiedano l'intervento di tecnici qualificati. Il linguaggio utilizzato è semplice e accessibile;

2. *manuale di manutenzione*: rivolto a tecnici specializzati, contiene informazioni dettagliate per la corretta manutenzione dell'impianto; descrive il funzionamento ordinario, le procedure per verifiche, riparazioni, sostituzioni e dismissioni, utilizzando un linguaggio tecnico adeguato al personale qualificato; include elaborati grafici, schede tecniche e istruzioni dettagliate sui componenti;
3. *programma di manutenzione*: destinato a tecnici specializzati, indica i controlli e le verifiche da eseguire sull'impianto e sui suoi componenti nel tempo, suddiviso a sua volta in sottoprogrammi:
 - *sottoprogramma delle prestazioni*: indica i requisiti e le prestazioni attese dell'impianto e dei suoi componenti;
 - *sottoprogramma dei controlli*: elenca le verifiche e i controlli periodici necessari, il cui esito determina la necessità di interventi per ripristinare il normale funzionamento;
 - *sottoprogramma degli interventi di manutenzione*: definisce la sequenza temporale degli interventi manutentivi da eseguire.

Al di fuori del contesto dei lavori pubblici, la scelta dei documenti da produrre per il piano di manutenzione è a discrezione del progettista.

Generalmente, si utilizzano schede di manutenzione (o scadenziari) che elencano le attività di controllo e manutenzione da eseguire su componenti e impianti, organizzate per tipologia. Per esempio, possono essere previste schede specifiche per:

- quadri elettrici;
- impianti di illuminazione;
- cabine elettriche;
- impianti speciali: come cablaggi strutturati, sistemi antincendio e impianti di videosorveglianza;
- impianto di terra, utile nel caso di questa trattazione;

In ogni scheda, a ciascun intervento viene associato un codice identificativo e viene definita una periodicità per l'esecuzione, che può essere mensile, trimestrale, semestrale o annuale.

La determinazione della periodicità per interventi non vincolati da norme o leggi deve essere effettuata con criterio, tenendo conto di vari fattori quali:

- *condizioni ambientali*: umidità, temperatura, presenza di agenti corrosivi.
- *sollecitazioni esterne*: vibrazioni, urti, esposizione a polveri o sostanze chimiche.

- *frequenza di utilizzo*: componenti utilizzati intensivamente possono richiedere ovviamente interventi più frequenti.

In Appendice 2 è possibile trovare la scheda di manutenzione completa relativa all'impianto di terra redatto dalla Consip, con codice ELT.AP.05 (https://www.mef.gov.it/export/sites/MEF/bandi/bandi-di-gara/2019/documenti/bando_2077/ID-2077-App_4_Elenco-attivita_NEW_NEW.pdf)

Questi aspetti influenzano il degrado dei componenti e, di conseguenza, la necessità di interventi manutentivi; un ulteriore riferimento per stabilire la periodicità degli interventi sono senza dubbio le *indicazioni fornite dai costruttori* nei manuali d'uso e manutenzione.

Quando le verifiche o i controlli periodici devono essere eseguiti su un elevato numero di componenti simili, come apparecchi di illuminazione o prese elettriche presenti nello stesso ambiente o in ambienti analoghi, è possibile adottare una strategia di verifiche a campione: questa metodologia prevede di effettuare un esame visivo su tutti i componenti, eseguendo controlli approfonditi solo su un campione rappresentativo, avendo cura di variare i componenti esaminati in ciascun ciclo di verifica.

È importante sottolineare che le norme tecniche non forniscono una descrizione dettagliata degli interventi di manutenzione da eseguire né della loro frequenza: la definizione del piano di manutenzione è quindi una responsabilità dell'utente dell'impianto, che deve valutare attentamente le esigenze specifiche, considerando:

- *disposizioni legislative e regolamentari*, ovvero gli obblighi normativi applicabili;
- *le norme tecniche*, standard di settore e “best practice”;
- *istruzioni del costruttore*, raccomandazioni specifiche per i prodotti utilizzati.

La scelta delle attività manutentive e della loro periodicità deve risultare da un'analisi approfondita di questi fattori, al fine di garantire la sicurezza, l'efficienza e la durabilità dell'impianto elettrico.

4.3 Esempio applicativo di calcolo dell'impedenza dell'anello di guasto

Per realizzare uno strumento che permettesse sia al progettista che ad un verificatore di ricavare l'impedenza di guasto partendo dagli elementi del modello e confrontarla con il valore misurato dallo strumento attraverso la metodologia descritta nel capitolo 2, ci si è orientati verso l'utilizzo di Dynamo.

Si tratta di un programma interno a Revit che sfrutta la programmazione visuale e che permette agli utenti di creare script personalizzati per automatizzare attività, gestire dati e generare geometrie complesse, senza per forza avere conoscenze avanzate di programmazione.

Con Dynamo, è possibile creare flussi di lavoro avanzati collegando nodi che rappresentano operazioni specifiche, come manipolare parametri degli elementi o interagire con database esterni.

Considerata la struttura dell'anello si è pensato quindi di realizzare dei blocchi che fossero in grado, se collegati in serie, di ripercorrere il circuito da valle a monte, partendo dall'elemento finale oggetto di verifica fino al centro stella del trasformatore che rappresenta, negli impianti di tipo TN-S, l'elemento di chiusura dell'anello.

Anziché percorrere il circuito in andata seguendo il conduttore di alimentazione e al ritorno quello di protezione, si è preferito risalire nello stesso verso il sistema elettrico sui due conduttori, in modo tale da utilizzare un solo tipo di blocco da replicare in ogni situazione.

I macro blocchi principali su cui si basa la struttura sono principalmente due: quello per la lettura dei circuiti ed estrapolazione della lunghezza del singolo circuito e quello per il calcolo della relativa impedenza: questa struttura base è stata poi replicata in successione per tutti i tratti di circuito di livello gerarchico superiore.

Tutto inizia da un blocco di input, che permette all'utente di selezionare l'elemento oggetto di verifica andando a cliccare direttamente l'oggetto nel modello di Revit: la selezione viene tradotta da Dynamo in un codice univoco, ovvero l'ID Revit associato a quello specifico oggetto. Da questo blocco hanno origine tre collegamenti:

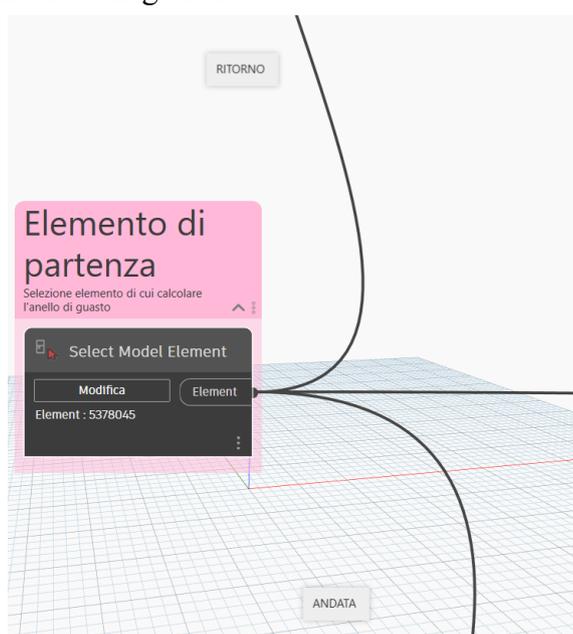


Figura 45: blocco selezione elemento di partenza

- quello chiamato “Ritorno”, che connette la serie di blocchi relativi al conduttore di protezione;
- quello chiamato “Andata”, che connette la serie di blocchi riguardanti il conduttore di alimentazione;
- quello in centro, che andrà a connettersi al blocco di somma e che verrà analizzato successivamente.

Percorrendo il collegamento etichettato come “Ritorno”, si entra nel macro blocco chiamato “Analisi dei circuiti”, consultabile nella sua interezza nell’immagine seguente (ogni blocco è quasi sempre seguito da un blocco watch per avere un pieno controllo dei risultati in uscita da ciascuno di essi):

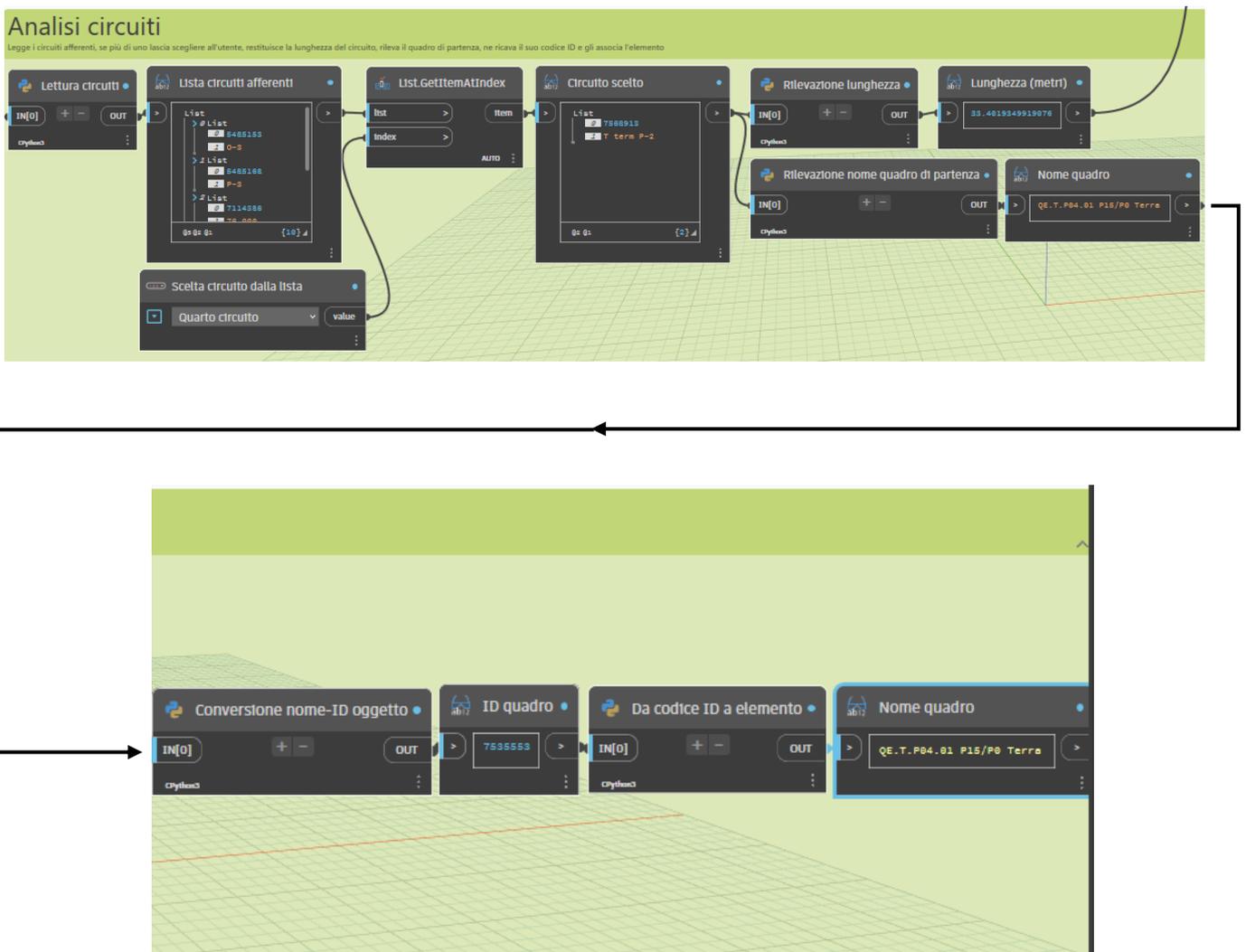


Figura 46: macroblocco "Analisi dei circuiti"

Analizzando i vari blocchi da sinistra a destra si possono trovare:

- Blocco **“Lettura circuiti”**: `get_circuit_elements`, che utilizza un costrutto di raccolta (`FilteredElementCollector`) per estrarre tutti gli elementi della categoria "Circuiti Elettrici" nel modello. Per ogni circuito trovato, lo script verifica se l'elemento selezionato è contenuto nella lista degli elementi del circuito, creando così un filtro mirato per i circuiti che effettivamente lo includono. Infine, lo script crea una lista con le informazioni sui circuiti trovati, tra cui l'ID e un parametro chiamato “Numero di circuito.” Questo output, restituito sotto forma di lista annidata, consente all'utente di ottenere una panoramica chiara e strutturata dei circuiti associati all'elemento selezionato direttamente in Dynamo. Un blocco di tipo `watch` ne visualizza l'output, che consiste in tante liste annidate quante sono i circuiti, ciascuna con un indice progressivo che parte da 0. Per ciascun circuito, in azzurro è presente il codice Id mentre in arancione la stringa di testo contenente il suo nome. Lo script relativo è consultabile all'appendice 3.

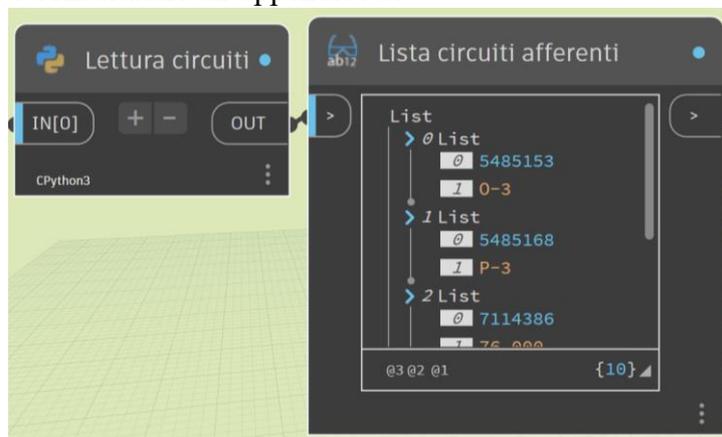


Figura 47: lettura circuiti

- Blocco **List.GetItemAtIndex**, che prende in ingresso le varie liste e un indice selezionabile manualmente dall'utente tramite un menù a tendina: serve ad estrapolare un solo circuito nei casi in cui, come quello di una torretta, ve ne siano collegati più d'uno. Il blocco è presente nella libreria di base di Dynamo; l'output viene poi inviato a due nodi distinti.

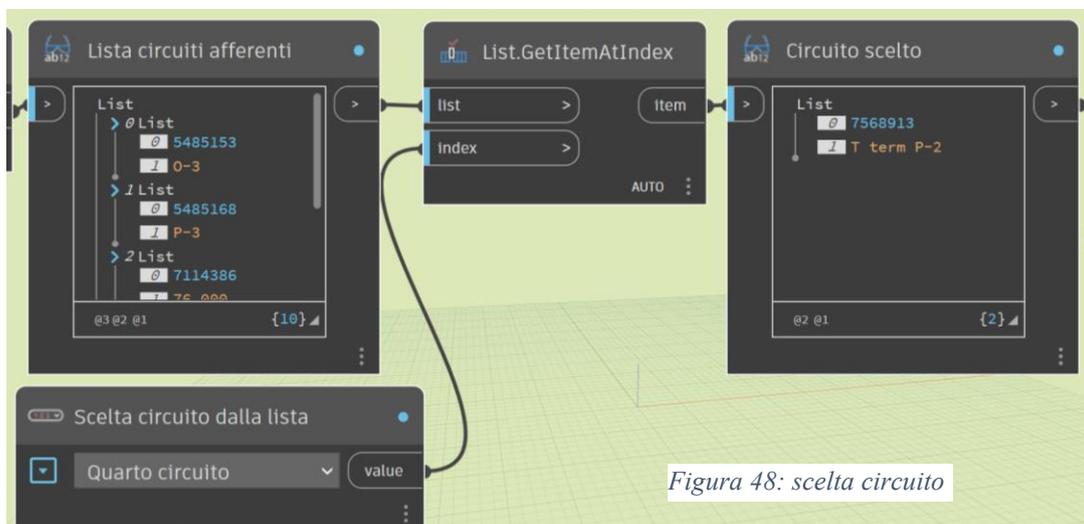


Figura 48: scelta circuito

- Blocco **“Rilevazione Lunghezza”**: è costituito da un blocco contenente uno script Python personalizzato, che legge l’ID Revit del circuito scelto e, consultando le proprietà del circuito prescelto, ne estrapola la lunghezza moltiplicata per un fattore correttivo visto che Dynamo utilizza i piedi e non i metri. Si ricorda che i circuiti fanno parte della Famiglia di sistema “Sistemi Elettrici”, e come tale possiedono parametri come le altre famiglie del progetto. Questo valore viene poi passato al macro blocco per il calcolo dell’impedenza, esaminato successivamente. Lo script relativo si trova in appendice 4.



Figura 49: rilevazione lunghezza

- Blocco **“Rilevazione nome quadro di partenza”**: sfruttando lo stesso script del blocco di sopra estrae dal circuito la stringa di testo relativa al nome del quadro da cui parte il circuito selezionato, ossia un elemento o generalmente un quadro posto al livello gerarchico immediatamente superiore. Lo script relativo si trova in appendice 5.



Figura 50: rilevazione nome quadro di partenza

- In successione si trovano poi altri due blocchi, ovvero **“Conversione nome-ID oggetto”** e **“Da codice ID a elemento”** che, rispettivamente associano alla stringa di testo un codice ID Revit ripercorrendo tutta l’attrezzatura elettrica e dal codice riescano ad associare l’elemento corrispondente.



Figura 51: conversione nome oggetto

Questo macro blocco termina con l’oggetto del quadro, che verrà passato ad un blocco identico per la reiterazione del processo di risalita del circuito completo: nel caso di questo progetto, la conclusione avviene raggiungendo l’elemento “Collettore CabMT”, ovvero l’elemento dell’impianto di terra più in alto nella scala gerarchica a cui il centro stella del trasformatore “TR1” è direttamente collegato. Di seguito viene proposta una raffigurazione e la descrizione del secondo macro blocco, ovvero quello relativo al calcolo dell’impedenza di guasto, data la lunghezza calcolata in precedenza.

Calcolo impedenza

Scelta tipologia e sezione, dizionario con valori specifici, calcolo valori totali, impedenza totale come output

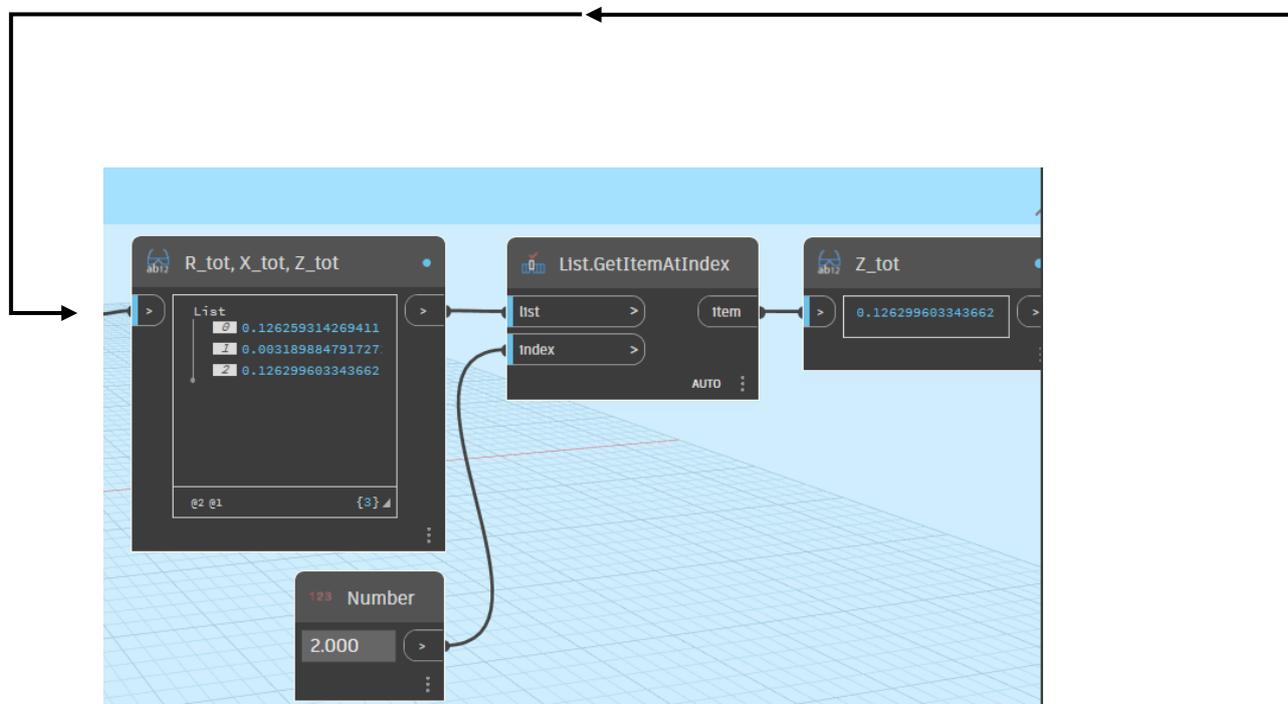
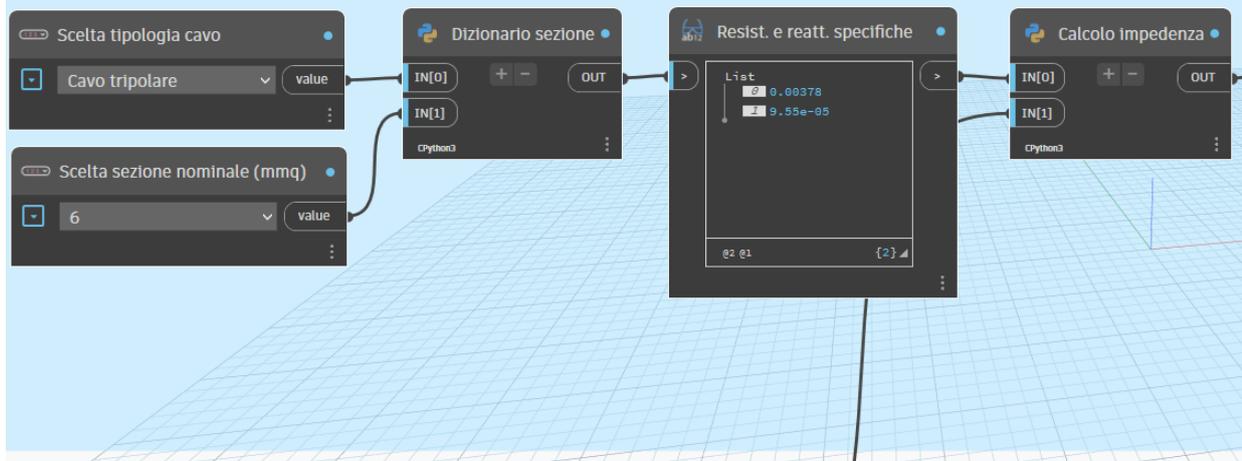


Figura 52: macroblocco "Calcolo impedenza"

Da sinistra verso destra:

- Blocco **“Dizionario Sezione”**: tramite uno script Python personalizzato, si è riusciti ad integrare una libreria che contenesse tutti i valori di resistenza e reattanza per unità di lunghezza (Ω/m), rilevati dalla norma CEI-UNEL 35023 [28]. La coppia di valori viene individuata lasciando all’utente la scelta della tipologia di cavo e la sua sezione, fermo restando che in uno scenario futuro questi parametri possano essere letti direttamente dal modello in Revit dopo la verifica di un corretto dimensionamento. Lo script è disponibile all’appendice 6.

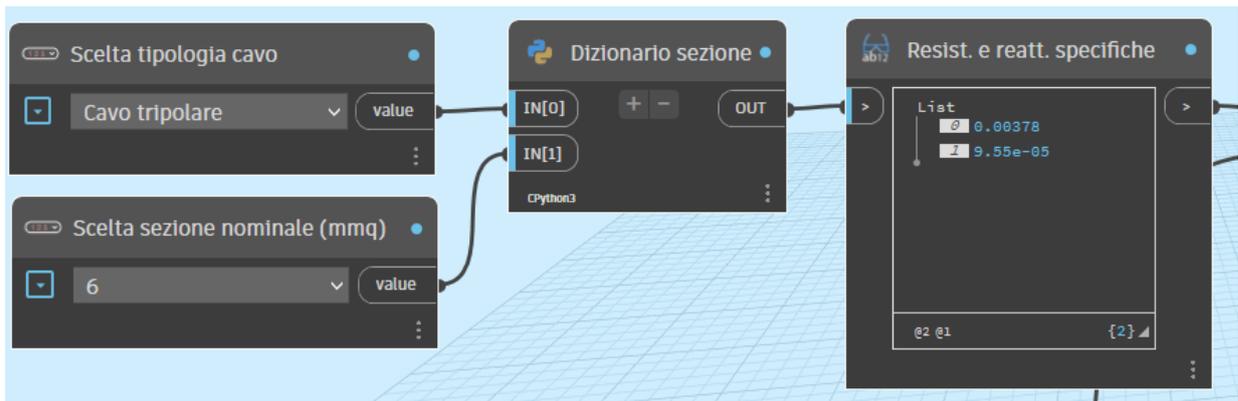


Figura 53: scelta parametri per ricavare i valori specifici

- Blocco **“Calcolo impedenza”**: sfruttando un breve script Matlab, prendendo in ingresso la coppia di valori precedenti e la lunghezza, si trova una tripletta di valori che nell’ordine risultano essere la resistenza totale, la reattanza totale e l’impedenza totale. Per l’impedenza si è utilizzato il teorema di pitagora per il calcolo del modulo:

$$|Z_{tot}| = \sqrt{R_{tot}^2 + X_{tot}^2} \quad (11)$$

Lo script è disponibile all’appendice 7.

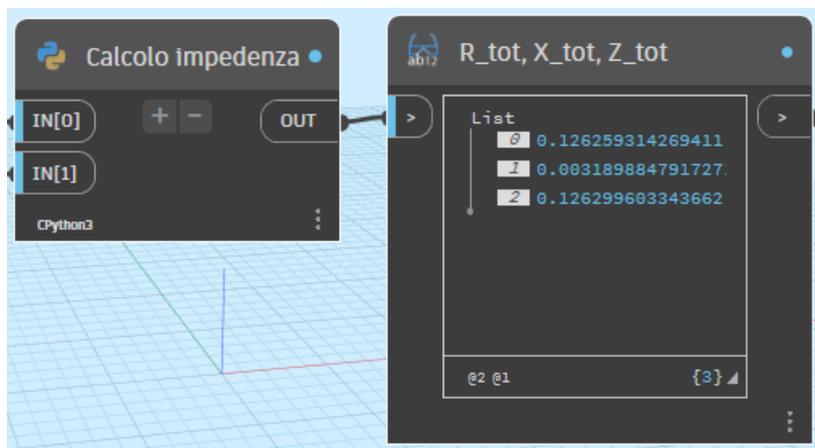


Figura 54: calcolo impedenza

- Blocco **List.GetItemAtIndex**, che prende in ingresso i tre valori e un indice selezionabile manualmente dall'utente: serve ad estrapolare il terzo valore (con indice 2) che è proprio l'impedenza totale. L'approssimazione del valore avverrà soltanto in fase finale, dopo la somma di tutte le impedenze.

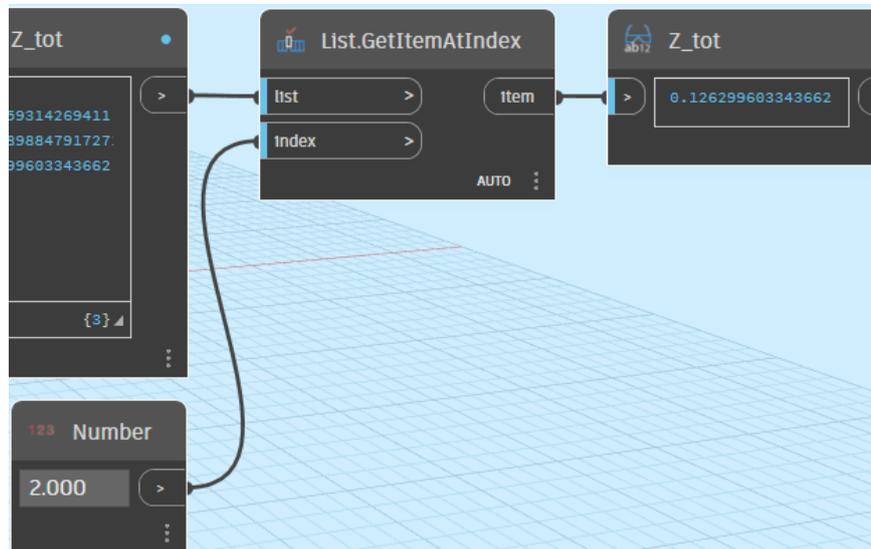


Figura 55: estrapolazione valore impedenza totale

Lo stesso procedimento ricorsivo viene adottato per il percorso di andata, seguendo il conduttore di alimentazione: il circuito termina con il trasformatore “TR1”. Il tratto che mette in comunicazione il centro stella del trasformatore “TR1” e il “Collettore CabMT” e che serve per chiudere effettivamente l’anello di calcolo, è stato trattato separatamente in quanto non viene considerato dallo script di risalita circuiti come una via da percorrere per salire di un livello gerarchico. Questo a pensarci bene non fa sorprendere, in quanto il circuito di alimentazione e quello di messa a terra, per come sono stati gestiti nel browser dei sistemi, appartengono a due circuiti elettrici non dialoganti tra di loro.

Entra perciò in gioco un macro blocco particolare (**Tratto trasformatore-collettore MT**) che richiede all’utente di scrivere nel campo di testo in ingresso il nome del centro stella del trasformatore interessato dal calcolo dell’anello di guasto: esso filtra tutti gli elementi della categoria attrezzatura elettrica e, tramite una maschera di tipo booleano, ricava l’oggetto corrispondente. In uscita si ha l’elemento corrispondente che viene passato in ingresso ai macro blocchi base descritti prima, ovvero quello dedicato alla lunghezza del circuito e quello per il calcolo dell’impedenza.

Di seguito è proposta la panoramica della serie di nodi utilizzata:

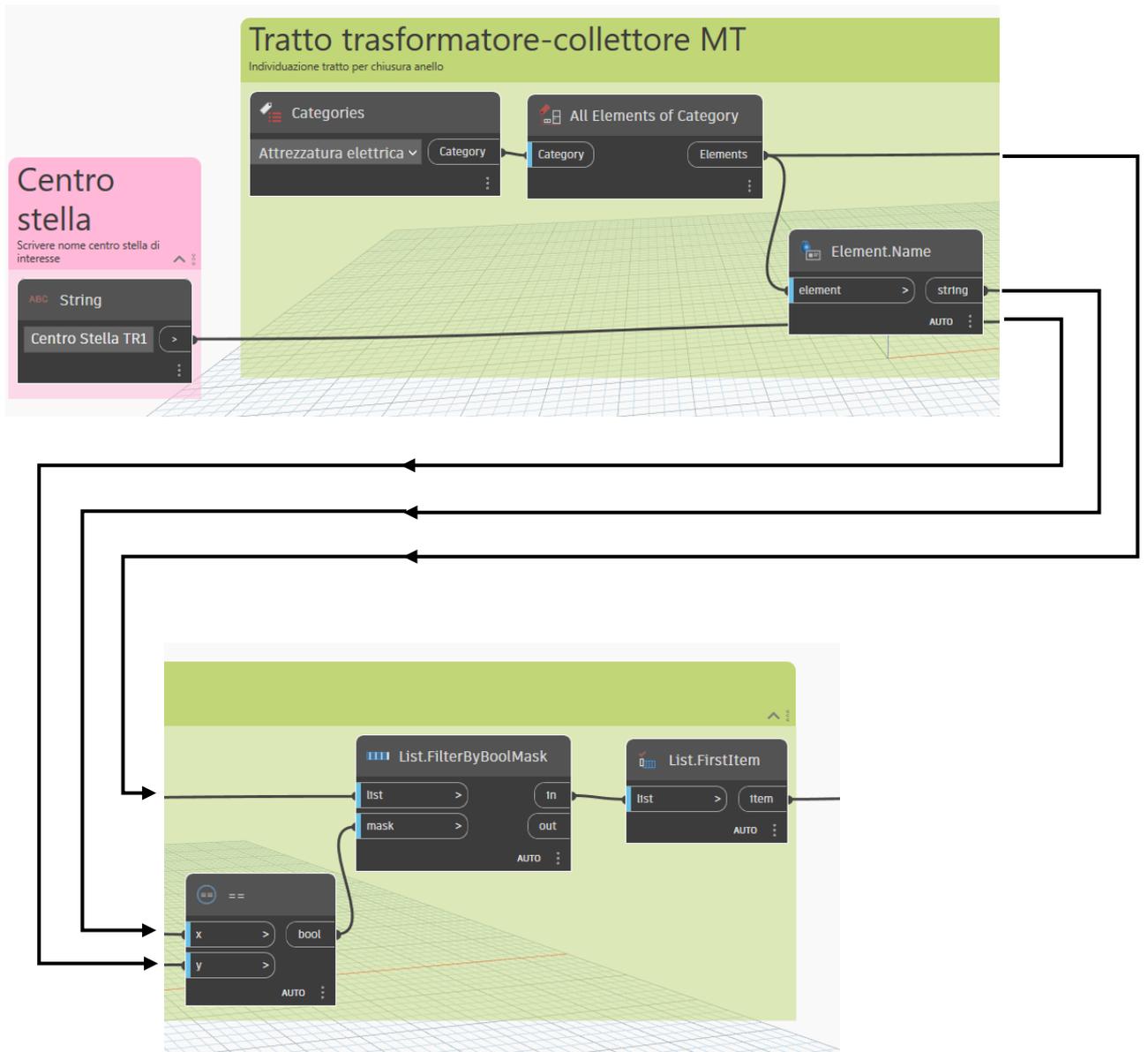


Figura 56: macroblocco "Calcolo tratto trasformatore-collettore MT"

Per chiudere il cerchio, è stato necessario anche il calcolo dell'impedenza interna al trasformatore: il calcolo è stato effettuato lasciando come sempre all'utente la scelta del trasformatore tramite il blocco che permette la selezione diretta dal modello Revit.

Si è utilizzata la seguente formula:

$$|Z_{tr}| = \frac{V_{2nom}^2}{S_{nom}} \cdot \frac{v_{cc\%}}{100} \quad (12)$$

con V_{2nom} la tensione nominale a secondario del trasformatore, con S_{nom} la sua potenza nominale e con $v_{cc\%}$ la tensione di cortocircuito percentuale.

I valori di questi tre parametri devono essere inseriti come stringa di testo manualmente dall'utilizzatore poiché ogni produttore gestisce i modelli secondo una propria metodologia, con nomi e sigle differenti, utilizzando talvolta, come nel caso mostrato la propria lingua madre anziché l'Inglese standard. Consapevoli del fatto che ogni elemento modellato importato andrebbe rivisto e adattato al proprio progetto, ma considerando che il fine ultimo di questa trattazione non sia la modellazione nel senso stretto del termine, ci si è limitati a mantenere i parametri originali; come è evidente, i tre valori di targa utilizzati sono in lingua francese, essendo il trasformatore prodotto da Schneider Electric.

Per mostrare un metodo alternativo all'integrazione del calcolo in un singolo codice Python, si è deciso di utilizzare una combinazione di nodi elementari per ricomporre i calcoli della formula di sopra, fatto salvo per il blocco di codice per eliminare il simbolo % dal valore della $v_{cc\%}$.

Qui di seguito la struttura completa del macro blocco:

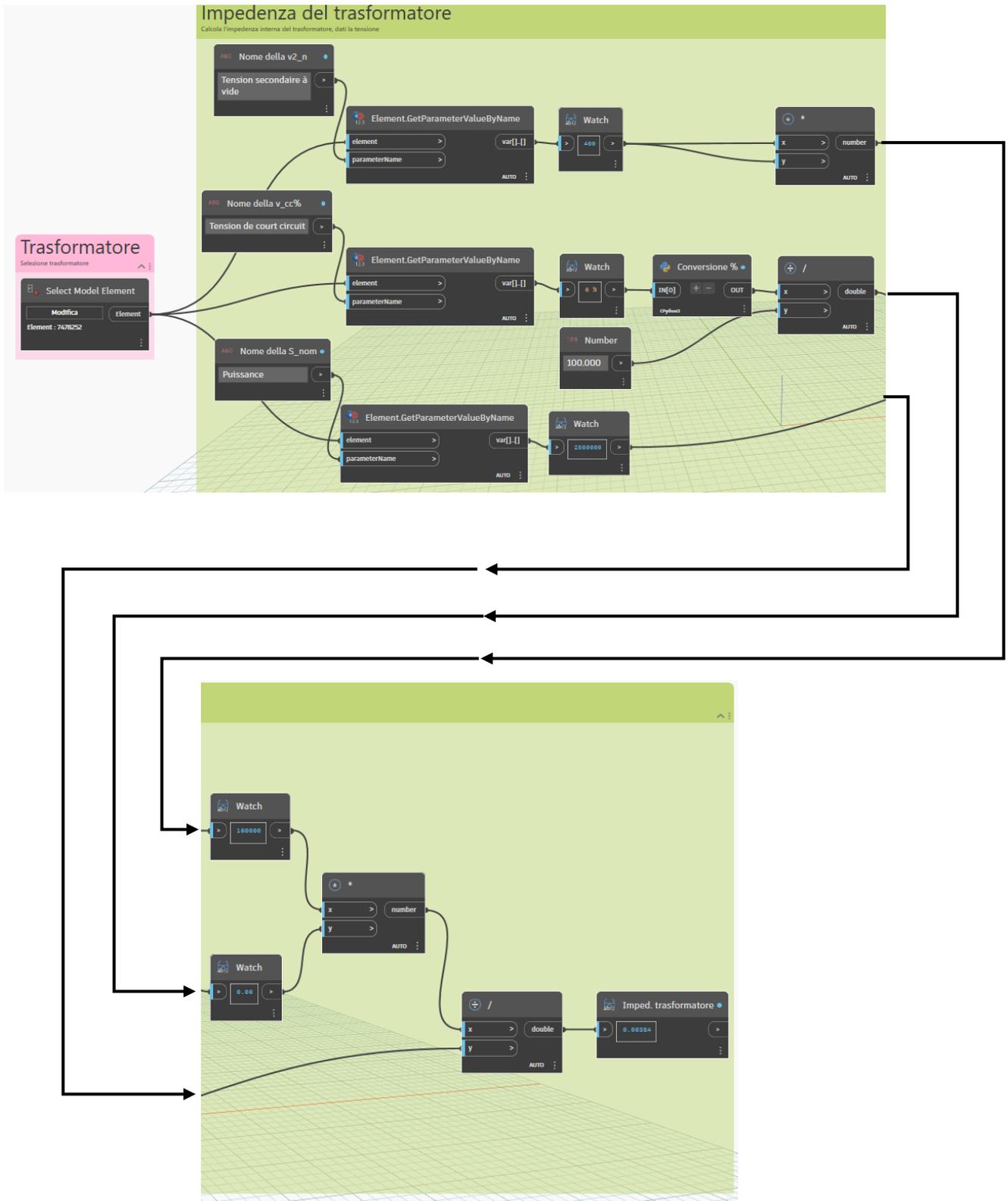


Figura 57: macroblocco "Impedenza del trasformatore"

Per giungere al valore dell'impedenza totale dell'anello, si è pensato di raggruppare tutte le singole impedenze in una lista chiamata "Raccolta impedenze": in questo modo è stato possibile implementare un controllo dei valori della lista.

Potrebbe succedere infatti che, scegliendo un elemento di partenza per il calcolo dell'anello posto più in alto nella gerarchia dei sistemi, si raggiunga prima l'origine dell'impianto, lasciando i blocchi successivi vuoti (con la comparsa del valore "null"): in questo caso il blocco **Math.Sum** che opera la sommatoria dei valori andrebbe in errore senza un opportuno meccanismo di controllo della consistenza dei valori.

Tale meccanismo è stato realizzato aggiungendo una maschera booleana che ha come criterio quello di individuare i valori nulli con il blocco **Object.IsNull** e negarne il valore con il blocco **Not**, così da ottenere la condizione "false" solo in corrispondenza di un valore nullo.

Superato il controllo, si entra nel nodo sommatore e il risultato viene ancora approssimato con tre cifre decimali grazie al blocco **Math.Round**, con visualizzazione in un nodo **watch** dedicato.

Per concludere la procedura e visualizzare su Revit il valore di impedenza calcolato, si passa il risultato ad un blocco nativo di Dynamo ovvero **Element.SetParameterByName**, che ha il compito di associare il valore ad un parametro il cui nome è inseribile nel blocco **String**: nel caso specifico del modello si tratta di un parametro condiviso di progetto chiamato "Impedenza anello di guasto".

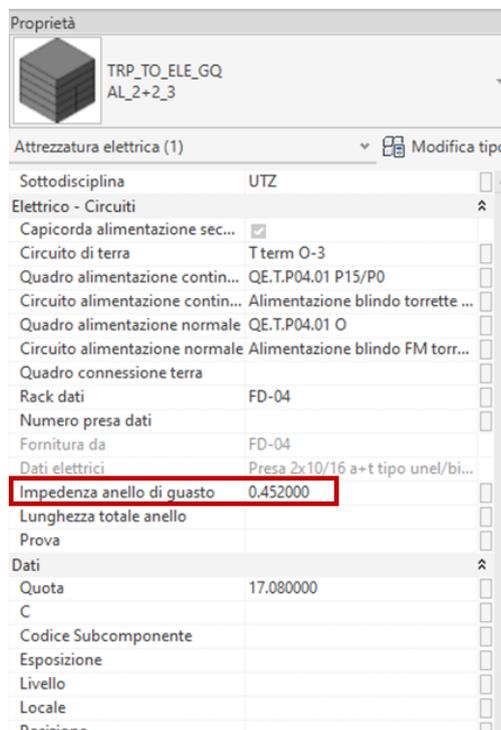


Figura 58: pannello delle proprietà torretta con valore dell'impedenza abbinato

Il blocco di abbinamento sa a quale elemento appartiene il parametro poiché vi è la connessione con il blocco di selezione iniziale su cui si è scelto di effettuare la verifica, come è possibile notare nella versione integrale dei blocchi qui sotto.

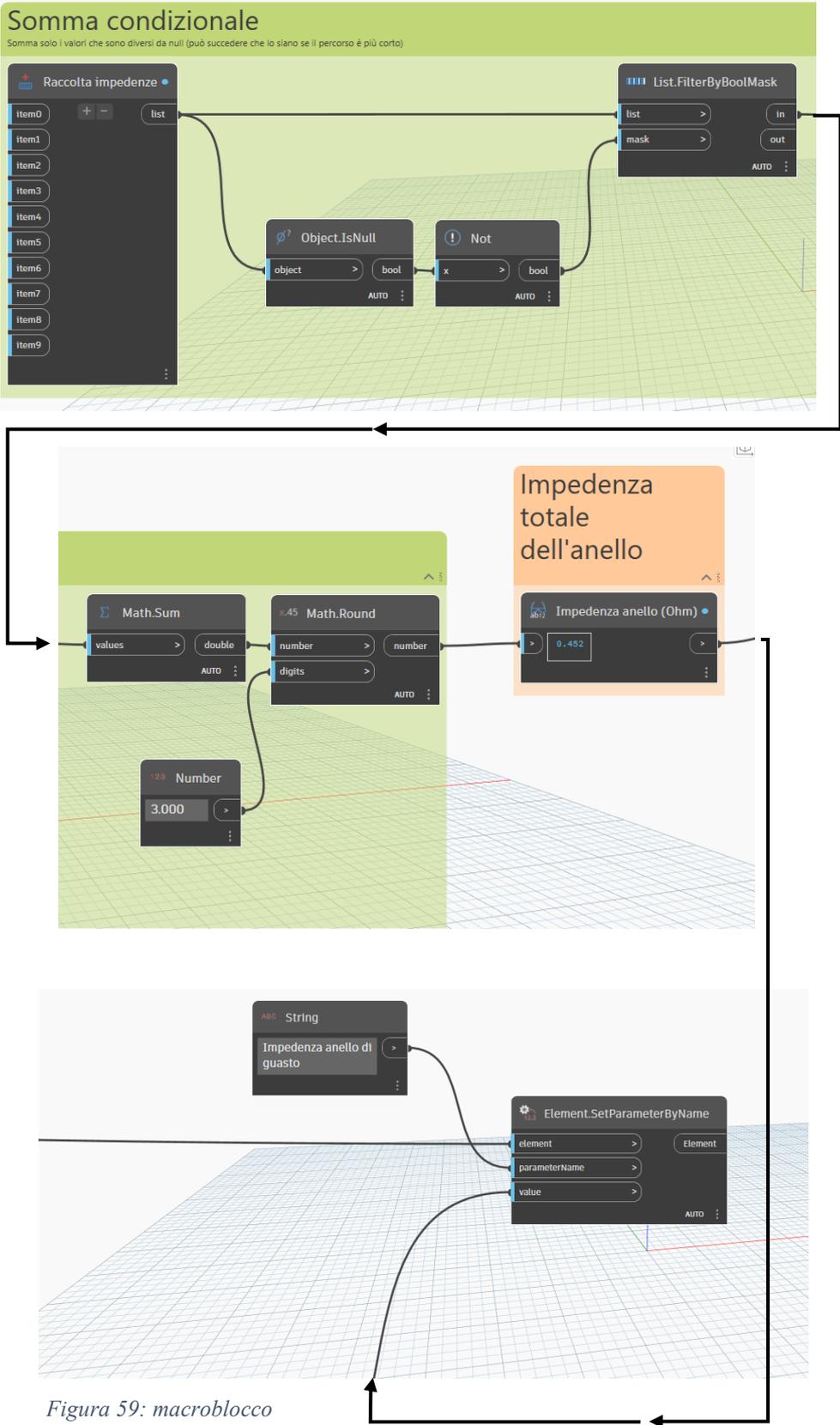


Figura 59: macroblocco "Somma condizionale"

Per semplificare l'interfaccia tra il programma e l'utilizzatore si è infine deciso di sfruttare il Lettore di Dynamo, ovvero un pannello che racchiude al suo interno tutti i blocchi del progetto identificato dall'utente come input o output: in questo modo si rende ancora più fruibile l'applicazione creata, dando la possibilità ad un eventuale verificatore, estraneo alle logiche di programmazione, di interagire soltanto con menù a tendina, campi di testo o di selezione elementi del progetto. Questo sicuramente sopperisce all'impossibilità di utilizzare il pacchetto Data.Shapes con la sua interfaccia utente, riscontrata a causa dell'incompatibilità tra le versioni di Python, garantendo l'interazione uomo-macchina in una finestra separata dal resto del codice, pulita e chiara.

Nell'immagine sottostante si può vedere come siano stati scelti, a titolo di esempio, i blocchi di partenza dei vari macroblocchi come ingresso e il valore di impedenza totale dell'anello di guasto come uscita.

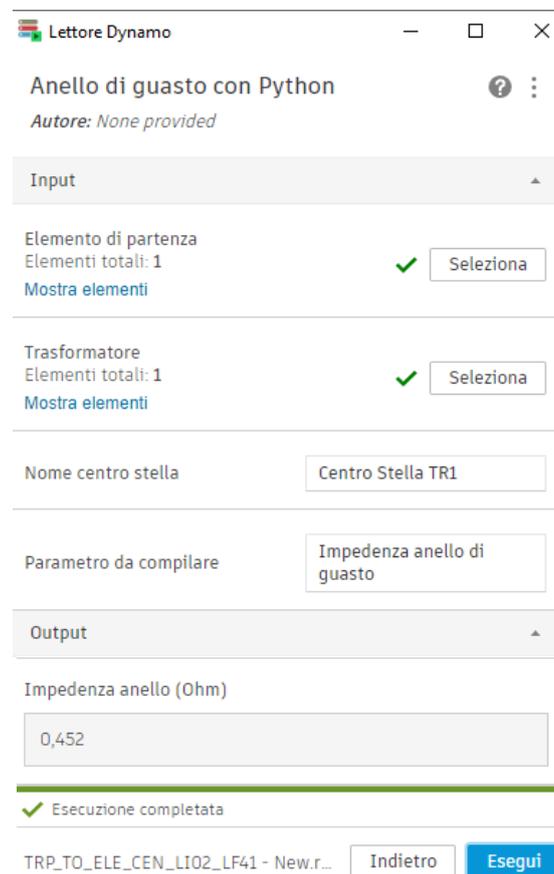


Figura 60: finestra Lettore Dynamo

4.4 Risultati numerici

Qui di seguito vengono riportati i risultati di ciascun macroblocco, sia in termini di lunghezza che di impedenza: la scelta dei parametri in ingresso non rappresenta la realtà delle cose ma è stata fatta secondo criteri di buon senso per dimostrare la funzionalità del sistema elaborato.

Elemento di partenza: torretta con **ID 5378045**

Circuito di alimentazione “Andata”:

- Nome circuito percorso: O-3
 - Lunghezza: 33.8 m
 - Tipologia cavo: tripolare
 - Sezione cavo: 6 mm²
 - Elemento di partenza circuito: QE.T.P04.01 O
 - Impedenza calcolata: 0.128 Ω
- Nome circuito percorso: 1
 - Lunghezza: 74.2 m
 - Tipologia cavo: unipolare
 - Sezione cavo: 35 mm²
 - Elemento di partenza circuito: QGBT.1.O
 - Impedenza calcolata: 0.048 Ω
- Nome circuito percorso: 1
 - Lunghezza: 12.9 m
 - Tipologia cavo: unipolare
 - Sezione cavo: 70 mm²
 - Elemento di partenza circuito: TR1
 - Impedenza calcolata: 0.004 Ω

Circuito di alimentazione “Ritorno”:

- Nome circuito percorso: T term O-3
 - Lunghezza: 33.1 m
 - Tipologia cavo: tripolare
 - Sezione PE: 6 mm²
 - Elemento di partenza circuito: QE.T.P04.01 O Terra
 - Impedenza calcolata: 0.125 Ω

- Nome circuito percorso: T-2
 - Lunghezza: 7.9 m
 - Tipologia cavo: unipolare
 - Sezione PE: 6 mm²
 - Elemento di partenza circuito: Collettore LT nord P04
 - Impedenza calcolata: 0.029 Ω
- Nome circuito percorso: T-1
 - Lunghezza: 8.2 m
 - Tipologia cavo: unipolare
 - Sezione PE: 6 mm²
 - Elemento di partenza circuito: Blindosbarra Verticale Terra
 - Impedenza calcolata: 0.030 Ω
- Nome circuito percorso: T-5
 - Lunghezza: 50.7 m
 - Tipologia cavo: unipolare
 - Sezione PE: 16 mm²
 - Elemento di partenza circuito: Collettore CabBT
 - Impedenza calcolata: 0.072 Ω
- Nome circuito percorso: T-13
 - Lunghezza: 11.7 m
 - Tipologia cavo: unipolare
 - Sezione PE: 35 mm²
 - Elemento di partenza circuito: Collettore CabMT
 - Impedenza calcolata: 0.007 Ω

Impedenza interna trasformatore:

- Trasformatore: TR1 (**ID 7478252**)
 - Tensione secondaria a vuoto: 400 V
 - Tensione di cortocircuito percentuale: 6%
 - Potenza: 2.5 MVA
 - Impedenza calcolata: 0.004 Ω

Tratto conduttore di terra tra TRI e Collettore CabMT:

- Nome circuito percorso: T-1
 - Lunghezza: 6.8 m
 - Tipologia cavo: unipolare
 - Sezione cavo: 70 mm²
 - Elemento di partenza circuito: Collettore CabMT
 - Impedenza calcolata: 0.002 Ω

Risultato somma condizionale:

Impedenza totale dell'anello di guasto: 0.451 Ω

Questo valore risulta essere abbastanza ragionevole, dato che per impianti con un'estensione simile a quella del grattacielo della Regione si tende a non superare quasi mai i 0.5 Ω.

5. Visualizzazione

Uno degli obiettivi principali di questa tesi è sicuramente quella della visualizzazione degli elementi modellati: nel primo capitolo vi è già stato un abbozzo di rappresentazione all'interno di Revit, tramite i filtri di vista colorati.

Si è deciso successivamente di rappresentare, inizialmente in ambiente virtuale, gli spazi della torre e tutte le attrezzature elettriche installate, con interesse specifico a quelle connesse all'impianto di terra.

La richiesta era quella di poter visualizzare fisicamente gli oggetti nello spazio tridimensionale e di ricavare, su richiesta dell'utente, i parametri relativi ad ogni specifico elemento, rendendoli modificabili tramite interazione e scrittura nei campi di testo.

Per tale scopo, ci si è orientati verso l'esportazione del modello nel software Unity, una piattaforma di sviluppo versatile per la creazione di contenuti interattivi, come giochi, simulazioni e applicazioni in 2D, 3D, realtà aumentata e virtuale: grazie alla sua flessibilità e agli strumenti integrati, Unity consente di sviluppare progetti per una vasta gamma di piattaforme, offrendo un ambiente intuitivo e ricco di funzionalità per sviluppatori di tutti i livelli.

5.1 Esportazione da Revit

Il primo passo è stato quello di trovare un plug-in in grado di generare un file leggibile da Unity, che non fosse a pagamento o che richiedesse complicati processi di gestione, dovendo un giorno potenzialmente far compiere questo passaggio a persone diverse da chi ha scritto questo trattato.

L'attenzione è ricaduta perciò sul plug-in VR-READY [29], estensione di Revit scaricabile gratuitamente dallo Store di Autodesk.



VR-READY
★★★★★ 1 recensione

Sistema operativo: Win64
Lingua: English

Gratuito
Scarica
Add to Wishlist ▾

Dimensioni download: 5,5 MB
Data di rilascio: 18/08/2023
Ultimo aggiornamento: 25/04/2024
Informazioni sulla versione: 1.1.1
Sito Web: <https://www.linkedin.com/in/ehsan-rahimian-b:z/>
Supporto clienti: xr.erz.apps@gmail.com

Figura 61: Plugin VR-READY

Una volta installato il pacchetto, esso si associa automaticamente a Revit, aggiungendo un nuovo tab nella barra degli strumenti; all'interno del programma presenta questa videata:

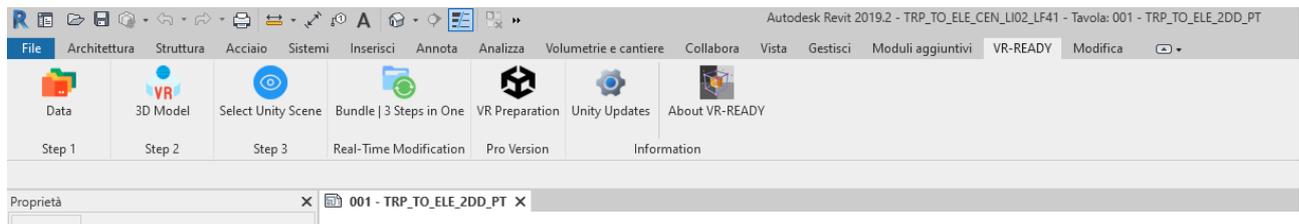


Figura 62: Barra degli strumenti di VR-READY

Cosa importante da ricordare è che l'estensione lavora solo su una vista 3D per volta, quindi si suggerisce di scegliere attentamente la vista 3D che contenga gli elementi da rappresentare e che siano visibili.

La procedura da seguire è guidata, cliccando nell'ordine i primi tre "step", da sinistra verso destra:

1. *Data*: questo pulsante genera un file .csv in una cartella nello stesso percorso del file Revit, con tutti i valori degli elementi presenti nella cista prescelta incluse le tavole e le viste stesse;
2. *3D Model*: questo pulsante genera invece il modello tridimensionale di ciò che è presente nella vista, depositandolo nella stessa cartella di prima;
3. *Select Unity Scene*: con quest'ultimo passo, si deve scegliere il percorso file che riconduca alla scena di Unity all'interno di un progetto esistente. La scena si trova all'interno della cartella di installazione di Unity, nel percorso tipo *.../Nome_Progetto/Assets/Scenes/Nome_della_scena*;

Questa sequenza va ripetuta soltanto la prima volta, per creare le cartelle, poi dalla volta seguente sarà sufficiente premere sul quarto pulsante da sinistra, ovvero il *Bundle | 3 Steps in One*, che come dice il nome esegue i primi 3 punti in un colpo solo: questo risulta estremamente utile per aggiornare il file di esportazione dopo la modifica del modello su Revit, potendo visualizzare così la variazione in tempo reale su Unity.

Si segnala anche la presenza di una funzionalità a pagamento, ovvero quella associata al tasto *VR Preparations* che consiste in un elenco dettagliato di azioni da compiere direttamente all'interno di Unity, comprendente le azioni descritte sopra, più gli script per la configurazione di un ambiente di gioco funzionante e l'associazione automatica dei materiali agli oggetti di gioco; per ultimo viene inserito nella scena anche uno scanner, impugnabile durante la visualizzazione interattiva ed in grado di restituire i parametri degli oggetti selezionati. Si è preferito realizzare autonomamente quest'ultima parte, al fine di una configurazione incentrata su determinati aspetti.

Una volta completata l'esportazione, la finestra in Unity si presenta così:

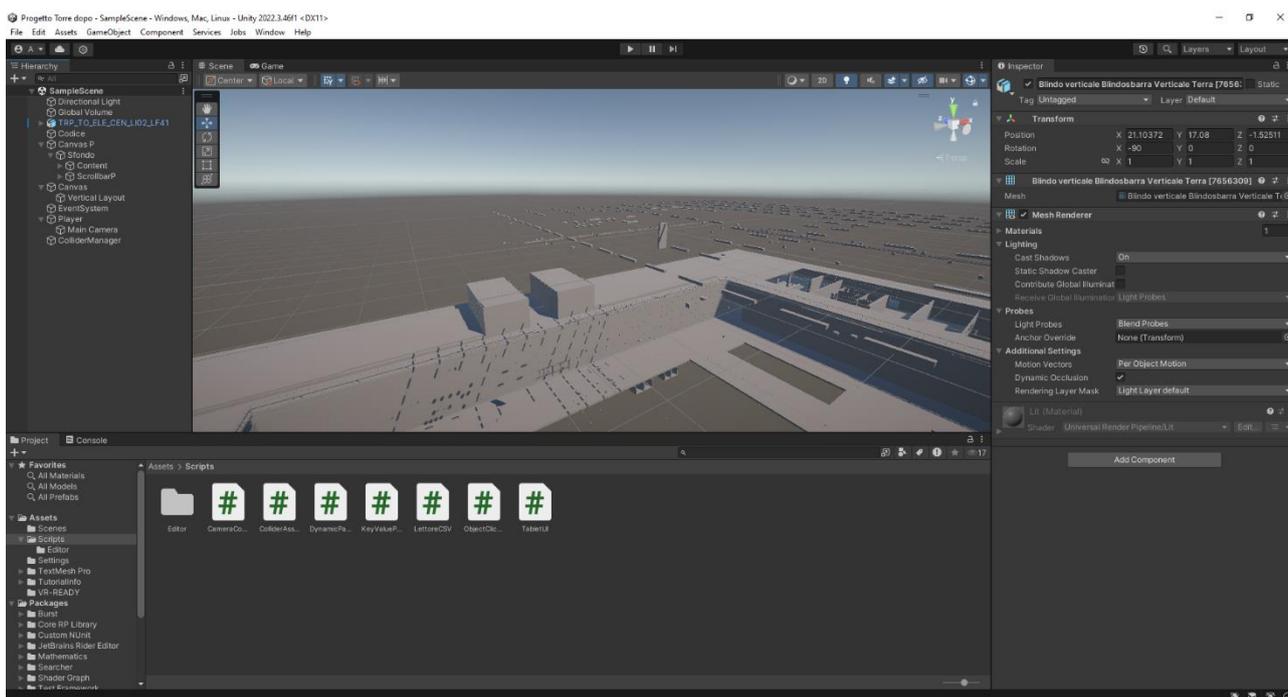


Figura 63: Finestra principale di Unity con modello importato

Nella parte sinistra trova posto la finestra della gerarchia, ovvero la struttura ad albero discendente di tutto ciò che è inserito nella scena, dagli oggetti del modello, al tablet in sovrimpressione fino alla camera e al personaggio.

In centro è collocata la videata relativa alla *scena*, ovvero l'ambiente in cui il creatore del "gioco" inserisce gli oggetti: si differenzia dalla videata "game" poiché quest'ultima è solo ciò che viene visto dal giocatore, in modalità in prima persona genericamente.

Sulla destra si ha invece l'*Inspector*, ovvero il pannello delle proprietà relativa agli oggetti di gioco: offre la possibilità di aggiungere blocchi e script per plasmare gli oggetti secondo le proprie esigenze.

Si fa notare che si è scelto volutamente di non importare il modello architettonico collegato della torre poiché avrebbe reso difficile visionare nella sua interezza l'impianto elettrico.

In basso si trova invece la sezione che raccoglie le cartelle di progetto, utile per accedere, tra le altre cose, agli script in C# che verranno descritti in seguito.

5.2 Importazione parametri da Revit

Volendo non solo visualizzare gli elementi nello spazio ma anche interagire con loro ricavando i parametri ad esso associati esattamente come se stessimo utilizzando la finestra delle proprietà in Revit, si è cercato di trovare un modo di leggere il file .csv generato da VR-READY: per far ciò è stata implementata una serie di script in C#, linguaggio informatico compatibile con Unity.

5.2.1 Codice LettoreCSV.cs

Lo script *LettoreCSV.cs* rappresenta il primo passo nel processo di integrazione dei dati BIM provenienti da *Revit* nella scena di *Unity*: esso gestisce la lettura e l'interpretazione dei dati da un file CSV, assegnando i parametri estratti ai *GameObject* presenti nella scena, che vengono così arricchiti con informazioni specifiche provenienti dal modello BIM. Questo script funge da base per il funzionamento di altri script di visualizzazione e interazione, che saranno descritti successivamente.

Il funzionamento principale di *LettoreCSV.cs* avviene nel metodo *Start*, eseguito all'avvio della scena: questo metodo chiama *ReadCsv* per leggere i dati dal file CSV e memorizzarli, riga per riga, in una lista. Subito dopo, viene eseguito *AssignDataToGameObjects*, che associa i dati importati ai *GameObject* appropriati all'interno della scena; per abilitare l'interazione fisica con gli oggetti, *Start* verifica anche la presenza dello script *ColliderAssigner*, che assegna automaticamente i collider agli oggetti trovati, permettendo la selezione tramite click. In assenza del *ColliderAssigner*, viene segnalato un errore, indicando che il componente è necessario per l'interattività completa.

La lettura effettiva del file CSV avviene nel metodo *ReadCsv*, che utilizza il riferimento al file *csvFile* per scorrere ogni riga del file e memorizzare i dati in una lista chiamata *csvData*: se il file CSV non è assegnato, viene generato un messaggio di errore per indicare la mancanza del file.

Il processo di assegnazione dei dati è gestito invece dal metodo *AssignDataToGameObjects*: per ogni riga letta, *ExtractParameters* estrae le coppie chiave-valore (ad esempio, Revit ID e Categoria) e le salva in un dizionario, rendendo accessibili i parametri per ogni *GameObject*. Se tra i parametri estratti è presente il Revit ID, *AssignDataToGameObjects* cerca il *GameObject* corrispondente nella scena tramite *FindGameObjectByRevitID*; quando il *GameObject* viene trovato, lo script gli assegna i parametri usando un componente *DynamicParameters*, che viene aggiunto automaticamente se non è già presente. *DynamicParameters* si occupa poi di memorizzare e aggiornare i dati, rendendoli visibili anche

nell'*Inspector* di Unity. Se invece il *GameObject* non viene trovato, l'ID mancante viene registrato nella lista *missingRevitIDs*, che alla fine del processo è riportata per segnalare quali oggetti non sono stati associati ai dati.

La ricerca del *GameObject* tramite Revit ID avviene nel metodo *FindGameObjectByRevitID*, che scorre tutti gli oggetti della scena alla ricerca di un nome che contenga l'ID all'interno di parentesi quadre: per estrarre l'ID dal nome dell'oggetto, *ExtractIDFromName* utilizza un'espressione regolare per verificare se il nome contenga un numero valido tra parentesi quadre: in caso affermativo, restituisce l'ID per il confronto, altrimenti restituisce il valore -1.

Lo script *LettoreCSV.cs* costituisce quindi la base del sistema di importazione e assegnazione dei parametri: altri script, che verranno illustrati successivamente, sfrutteranno questi dati per consentire la visualizzazione e interazione avanzata con i *GameObject*, rendendo il modello BIM completamente accessibile e interattivo all'interno di Unity.

È possibile consultare lo script completo all'appendice 8.

5.2.2 Codice *KeyValuePair.cs*

Lo script *KeyValuePair.cs* definisce una semplice classe per rappresentare coppie di chiavi e valori: viene utilizzato come struttura dati all'interno del sistema per gestire e visualizzare i parametri importati dal file CSV.

La classe *KeyValuePair* permette di memorizzare ogni parametro come una coppia key-value, agevolando l'organizzazione e il recupero delle informazioni in modo strutturato: questa classe è dichiarata come *[Serializable]*, rendendo ogni istanza serializzabile e quindi visibile e gestibile sia nell'*Inspector* di Unity che in altre strutture dati. È composta da due variabili pubbliche:

- *key*: rappresenta il nome del parametro (es., Revit ID, Tensione nominale);
- *value*: rappresenta il valore associato alla chiave (es., 212664, 400 V).

Il costruttore della classe *KeyValuePair* consente di inizializzare rapidamente entrambe le variabili, facilitando così la creazione di coppie chiave-valore.

All'interno del sistema complessivo, la classe *KeyValuePair* è utilizzata in combinazione con *LettoreCSV.cs* e altri script che gestiscono i dati del modello BIM: ad esempio, *LettoreCSV* popola dinamicamente i parametri importati nel formato *KeyValuePair* e li assegna ai *GameObject* tramite il componente *DynamicParameters*.

Inoltre, *KeyValuePair* agevola il passaggio e la visualizzazione dei dati nella UI, in quanto il formato key-value è intuitivo da leggere e visualizzare; l'utilizzo di questa struttura rende così il codice più leggibile e gestibile, soprattutto nei casi in cui i parametri devono essere manipolati o visualizzati in altre parti dell'applicazione, come nella UI del tablet.

Questo script fornisce quindi un tassello essenziale per l'organizzazione e la manipolazione dei dati, che verranno poi sfruttati dagli altri script per la visualizzazione e interazione in Unity.

È possibile consultare lo script completo all'appendice 9.

5.2.3 Codice *DynamicParameters.cs*

Lo script *DynamicParameters* gestisce in modo dinamico i parametri per ciascun *GameObject* importato nella scena di Unity, offrendo funzioni per il salvataggio e il caricamento persistente dei dati: in stretta relazione con gli script *LettoreCSV.cs*, che si occupa di popolare i *GameObject* con parametri letti da un file CSV e *KeyValuePair.cs*, che organizza i parametri come coppie chiave-valore, *DynamicParameters* consente di manipolare e salvare queste informazioni in modo efficace.

I parametri vengono poi memorizzati in un dizionario “parameters”, in cui ogni coppia chiave-valore rappresenta un attributo dell'oggetto, come ad esempio un codice identificativo o una categoria: per rendere questi parametri visibili e modificabili direttamente nell'Inspector di Unity, lo script include anche una lista *keyValuePair* composta da istanze della classe *KeyValuePair*.

Il file di salvataggio dei parametri viene generato con un percorso univoco *saveFilePath*, basato sul nome del *GameObject*, così da evitare sovrapposizioni tra i vari file di dati.

All'avvio della scena, il metodo *Start* prepara il file di salvataggio creando un nome sicuro (*safeFileName*) per il *GameObject* corrente, eliminando caratteri non validi; successivamente, richiama *LoadParameters* per caricare i dati salvati in precedenza, popolando così sia il dizionario *parameters* che la lista *keyValuePair*. In questo modo, eventuali modifiche apportate ai parametri vengono automaticamente caricate e pronte per l'uso all'inizio della sessione.

Il metodo *UpdateInspectorData* svolge una funzione chiave per mantenere aggiornati i dati visualizzati nell'Inspector: tale metodo trasforma ogni coppia di *parameters* in un oggetto *KeyValuePair* nella lista *keyValuePair*, permettendo di visualizzare e modificare facilmente le informazioni direttamente nell'interfaccia *Unity*; questo aggiornamento rende immediatamente visibili le modifiche, migliorando la gestione interattiva dei parametri durante la fase di sviluppo.

DynamicParameters garantisce anche la persistenza dei dati tramite i metodi *SaveParameters* e *LoadParameters*, che salvano e ricaricano i parametri modificati, in modo da conservare le modifiche una volta usciti dal “play mode”.

Al termine della sessione, *OnApplicationQuit* richiama *SaveParameters* per salvare i dati in un file JSON: questo processo utilizza la classe *SerializableDictionary*, che converte il dizionario *parameters* in due liste (keys e values), facilitando la serializzazione e il salvataggio in JSON. *LoadParameters* opera in modo simile, leggendo il file JSON e ricostruendo il dizionario dei parametri tramite *SerializableDictionary*, garantendo che le modifiche siano sempre disponibili al riavvio.

Infine, la classe di supporto *SerializableDictionary* facilita la conversione tra liste e dizionario, offrendo una soluzione pratica per la gestione della serializzazione in *Unity*.

Con *DynamicParameters*, i parametri diventano facilmente gestibili e persistenti, completando l’integrazione con gli altri script per offrire una visualizzazione interattiva e ricca di informazioni all'interno della scena di *Unity*.

La coesistenza di questi tre script permette al metodo di importazione di essere abbastanza flessibile per quanto riguarda la quantità di parametri associati ad ogni oggetto, in modo tale che non vada in errore se tra un play e l’altro in Revit si modifichi qualcosa al contenuto informativo delle istanze.

È possibile consultare lo script completo all’appendice 10.

5.2.4 Codice *DynamicParametersEditor.cs*

Lo script è un editor personalizzato per il componente *DynamicParameters* che consente di visualizzare e modificare i parametri dinamici direttamente nell’*Inspector* di *Unity*: esso estende la classe *Editor* di *Unity*, offrendo un’interfaccia più intuitiva per interagire con i dati contenuti nel componente *DynamicParameters*. Utilizzato insieme a *DynamicParameters.cs*, *DynamicParametersEditor* migliora l’usabilità nell’editor, rendendo più semplice la gestione e la modifica dei dati.

Nel metodo *OnInspectorGUI*, lo script crea una tabella nell’*Inspector* per visualizzare la lista *keyValuePair*, che contiene i parametri chiave-valore di ciascun *GameObject*. All’inizio, viene visualizzato un titolo “Parametri Dinamici” e, se *keyValuePair* è vuota o nulla, viene mostrato un messaggio che indica l’assenza di parametri disponibili: questo aiuta a identificare rapidamente la presenza di dati per ciascun oggetto.

Se *keyValuePair*s contiene parametri, lo script genera una tabella con due colonne: Key, che rappresenta la chiave (ad esempio, "Categoria" o "Revit ID") e Value, che mostra il valore associato alla chiave. Ogni chiave è visualizzata come sola lettura, mentre il valore può essere modificato dall'utente tramite un campo di testo (*TextField*): questo permette di aggiornare rapidamente i dati associati a ogni *GameObject* direttamente dall'*Inspector*.

Per garantire che le modifiche ai valori siano salvate, lo script aggiorna *keyValuePair*s ogni volta che viene effettuata una modifica, impostando *EditorUtility.SetDirty(target)* se ci sono cambiamenti, così che Unity riconosca l'aggiornamento dei dati e lo mantenga anche durante il Play Mode.

È possibile consultare lo script completo all'appendice 11.

5.2.5 Codice *ColliderAssigner.cs*

Lo script *ColliderAssigner* si occupa di assegnare dinamicamente collider ai *GameObject* presenti nella scena di Unity, consentendo un'interazione fisica e il rilevamento dei click sugli oggetti importati dal modello BIM: questo script è utilizzato insieme a *LettoreCSV.cs*, che popola la scena con i dati importati dal file CSV e permette di escludere specifici oggetti dall'assegnazione dei collider per ottimizzare l'interattività.

Lo script include una variabile *excludedNamePart*, che contiene una stringa utilizzata per identificare i *GameObject* da escludere dall'assegnazione dei collider: infatti ci si è accorti che un modello architettonico denominato "Modello di Revit collegato" con la relativa vista statica andava a posizionarsi proprio davanti alla camera di gioco. Questa esclusione è particolarmente utile per mantenere le prestazioni ottimali e ridurre il numero di oggetti con cui è possibile interagire.

Il metodo principale dello script è *AssignColliders*, che trova tutti i *GameObject* presenti nella scena e li scorre uno per uno; per ciascun oggetto, verifica se il nome contiene la stringa di esclusione (*excludedNamePart*): se è così, l'oggetto viene ignorato e il ciclo prosegue con il successivo. Altrimenti, lo script controlla la scala locale dell'oggetto: se uno dei valori della scala è negativo, viene aggiunto un *MeshCollider* convesso, poiché i *MeshCollider* sono più adatti per oggetti di forme irregolari, soprattutto con scale non positive, e garantiscono una migliore risposta al *raycast*. Se la scala è positiva, lo script aggiunge invece un *BoxCollider* all'oggetto, a meno che non sia già presente un collider.

Questo script risulta indispensabile per abilitare l'interazione con i *GameObject*, rendendo possibile la selezione degli oggetti tramite *raycast*, un elemento chiave per la visualizzazione

interattiva dei dati importati. In particolare, *ColliderAssigner* opera in sinergia con *LettoreCSV.cs* per consentire all'utente di cliccare e visualizzare le informazioni di ogni oggetto attraverso la UI e lo script *DynamicParameters*, che gestisce e memorizza i parametri associati.

È possibile consultare lo script completo all'appendice 12.

5.2.6 Codice CameraController.cs

Lo script *CameraController* gestisce il movimento della camera in *Unity*, permettendo all'utente di navigare liberamente nella scena attraverso input da mouse e tastiera: questa funzionalità di navigazione si è resa necessaria per esplorare e analizzare da vicino i *GameObject* importati, arricchiti con parametri provenienti dal modello BIM. Lo script si integra con altri componenti della scena, come *ColliderAssigner.cs*, che abilita l'interazione con i *GameObject*, e *LettoreCSV.cs*, che importa e visualizza i dati sui *GameObject*.

L'interazione con la camera è regolata da due parametri: *mouseSensitivity*, che determina la sensibilità del movimento della camera in risposta agli input del mouse, e *moveSpeed*, che imposta la velocità di spostamento della camera: questi parametri possono essere modificati nell'*Inspector* per adattarsi alle preferenze dell'utente o alle esigenze della scena.

Nel metodo *Start*, lo script rende visibile il cursore e sblocca il suo movimento, in modo che l'utente possa interagire liberamente con la scena: ciò è particolarmente utile quando si desidera esaminare oggetti specifici o navigare verso un'area particolare del modello importato.

Nel metodo *Update* invece, la camera risponde agli input di rotazione e movimento; la rotazione viene gestita tramite gli assi del mouse: *mouseX* controlla la rotazione orizzontale, mentre *mouseY* gestisce la rotazione verticale, la quale è limitata tra -90° e 90° tramite *Mathf.Clamp* per evitare una rotazione completa a testa in giù. Il movimento avanti, indietro e laterale è invece controllato dagli input *W*, *A*, *S*, e *D*, che modificano la posizione della camera lungo gli assi laterale e frontale.

Grazie a questa navigazione, è possibile selezionare e analizzare agevolmente i *GameObject* interattivi, rendendo l'ambiente di *Unity* un vero e proprio strumento di esplorazione e analisi del modello BIM.

È possibile consultare lo script completo all'appendice 13.

5.2.7 Codice TabletUI.cs

Lo script *TabletUI* gestisce la visualizzazione e l'interazione con i parametri dei *GameObject* all'interno di una UI simile a un tablet, permettendo all'utente di leggere e modificare

i valori direttamente sullo schermo. Questo script si integra strettamente con *DynamicParameters.cs*, che memorizza i parametri del *GameObject*, e con *LettoreCSV.cs*, che importa i dati originali da un file CSV. Grazie a questo script, è possibile visualizzare e interagire in tempo reale con i dati, rendendo l'esperienza utente interattiva e accessibile.

Lo script si basa su tre riferimenti principali assegnati tramite l'*Inspector*:

- **content**: un contenitore in cui vengono istanziati i campi per ogni parametro;
- **parameterFieldPrefab**: un *prefab* utilizzato per visualizzare ogni coppia chiave-valore come un elemento della UI;
- **scrollRect**: componente che consente di scorrere i parametri visualizzati.

Il metodo principale, *ShowParameters*, viene utilizzato per popolare la UI con i parametri di un *GameObject* selezionato: prima di aggiungere nuovi elementi, *ShowParameters* elimina tutti i figli presenti nel contenitore *content* per evitare duplicazioni, garantendo una visualizzazione aggiornata a ogni nuova selezione. Ogni parametro viene quindi visualizzato utilizzando *parameterFieldPrefab*, che contiene due campi: *ParamName* per la chiave (non modificabile) e *ParamValue*, un *TMP_InputField* per il valore.

Ogni volta che il valore di un parametro viene modificato nell'*InputField*, un *listener* aggiorna automaticamente sia il *keyValuePair* sia il dizionario *parameters* di *DynamicParameters*, consentendo di memorizzare i nuovi valori anche dopo il Play Mode. Questo processo bidirezionale è essenziale per assicurare che tutte le modifiche siano riflesse in tempo reale sia nella UI che nella struttura dati sottostante, favorendo la coerenza e l'integrità delle informazioni.

Per ottimizzare l'interfaccia, *TabletUI* include il metodo *ForceScrollRectUpdate*, che forza un aggiornamento del layout: questo metodo assicura che la *ScrollRect* sia impostata correttamente e che i nuovi parametri siano visualizzati dall'inizio della lista, migliorando l'usabilità della UI.

È possibile consultare lo script completo all'appendice 14.

5.2.8 Codice *ObjectClickHandler.cs*

Questo script permette di selezionare e visualizzare i parametri dei *GameObject* nella scena di Unity tramite il click del mouse: rileva l'interazione dell'utente con gli oggetti interattivi della scena, rendendo possibile l'apertura del pannello UI (il "tablet") che mostra i dati specifici di ciascun oggetto. *ObjectClickHandler* opera in stretto contatto con gli altri script, come *TabletUI.cs*, che gestisce la visualizzazione dei dati sul tablet, e *DynamicParameters.cs*, che memorizza e struttura i parametri.

Il funzionamento dello script ruota attorno al metodo `Update`, che rileva i clic del mouse e li traduce in comandi per attivare la visualizzazione dei parametri. Il primo controllo, tramite `EventSystem.current.IsPointerOverGameObject()`, verifica se il clic è stato effettuato su un elemento UI: se l'utente clicca su un'interfaccia, l'interazione viene ignorata per evitare conflitti. Successivamente, `Update` rileva se è stato premuto il tasto sinistro del mouse e genera un Ray dalla posizione del clic nello spazio della scena, utilizzando `mainCamera` come punto di origine.

Un Raycast verifica quindi se il Ray colpisce un `GameObject` all'interno del `layer clickableLayer`: se l'oggetto colpito contiene il componente `DynamicParameters`, lo script procede ad attivare il pannello del tablet (`tabletPanel`) e a passare i dati di `DynamicParameters` allo script `TabletUI`, il quale popola la UI con i parametri del `GameObject` selezionato. Questo processo è reso possibile dalla funzione `ShowParameters` di `TabletUI`, che riceve i dati da `ObjectClickHandler` e aggiorna la visualizzazione del tablet. Se `tabletPanel` o `tabletUI` non sono assegnati, viene generato un errore per ricordare di collegare correttamente i riferimenti.

In caso di clic su un oggetto che non contiene `DynamicParameters`, viene mostrato un messaggio di avviso per indicare che l'oggetto non è interattivo; inoltre, se il Raycast non colpisce alcun `GameObject`, viene visualizzato un messaggio di avvertimento per indicare l'assenza di oggetti selezionabili sotto il cursore.

È possibile consultare lo script completo all'appendice 15.

5.3 Visualizzazione in realtà virtuale

La visualizzazione in realtà virtuale è la trasposizione di quanto realizzato in Unity, con la possibilità di essere proiettati nel modello e poter interagire attraverso i controller anziché con il mouse e la tastiera.

I risvolti positivi sono molteplici, dalla formazione in aula del personale addetto alla manutenzione a studi di fattibilità su modifiche e sostituzioni sull'impianto; rispetto alla realtà mista, non è richiesto il lavoro di posizionamento delle ancore né tantomeno l'essere fisicamente sul posto per visionare la porzione di impianto soggetta a verifica.

Allo scopo è stato utilizzato il Meta Quest 3, un visore di realtà virtuale (VR) e realtà mista (MR) sviluppato da Meta (precedentemente Facebook): si distingue per essere un dispositivo standalone, ovvero funzionante autonomamente senza bisogno di un computer, grazie al processore integrato e ad una risoluzione migliorata rispetto ai suoi predecessori. Il Quest 3 offre un'esperienza di realtà mista a colori, che permette di vedere e interagire con il mondo reale mentre si è immersi in ambienti virtuali, oppure visualizzare modelli interamente virtuali; possiede inoltre due controller avanzati e supporto al tracciamento delle mani.



Figura 64: visore Meta Quest 3 e controller

Da Unity, con le dovute modifiche alle impostazioni, è possibile generare delle build in formato .apk da salvare ed eseguire sul visore, senza bisogno di un collegamento ad un computer o richiedendo una connessione alla rete.

Gli script sono stati mantenuti quasi totalmente uguali a quelli generati per la visualizzazione su Unity, riorganizzando la gerarchia degli oggetti e aggiungendo ad esempio un raggio di colore rosso per puntare gli oggetti ed interagire con essi.

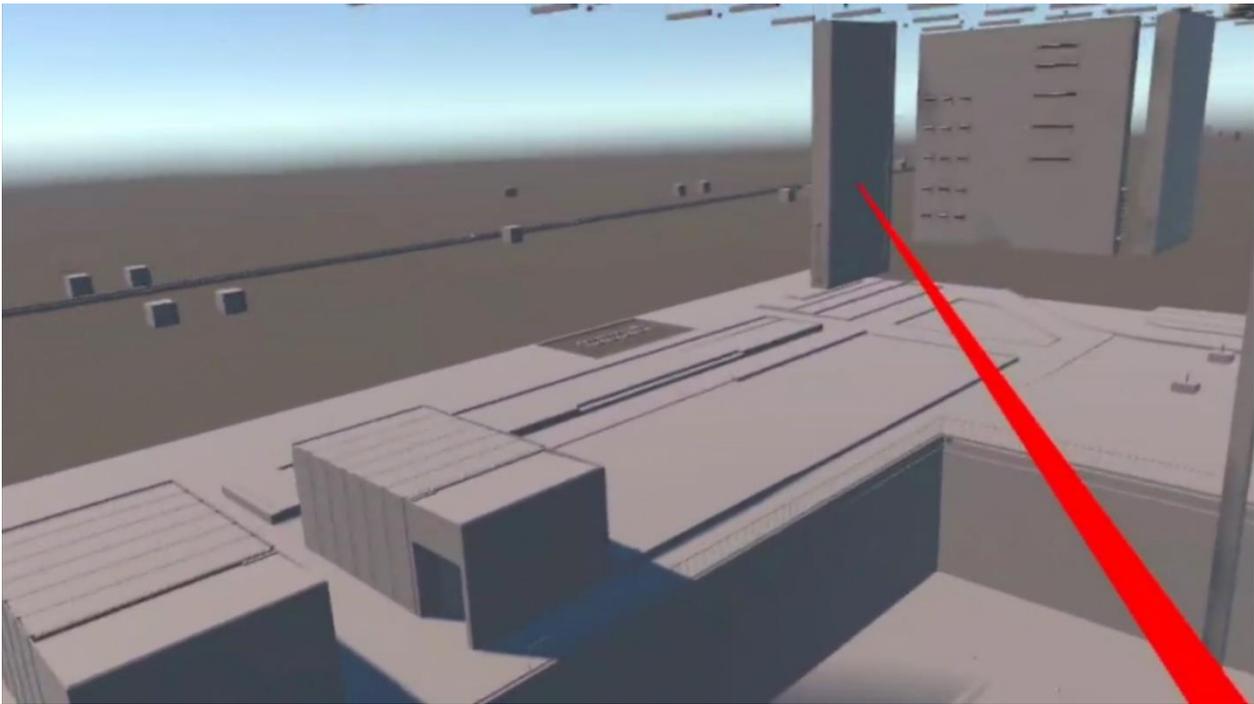


Figura 65: navigazione nel modello virtuale e interazione con puntatore rosso

Cliccando su ogni oggetto compare a sinistra dello schermo un pannello che raccoglie tutti i parametri relativi ad esso: le coppie chiave-valore, create e associate dagli script descritti in precedenza, vengono stampate in quest'area, proponendo nel caso dei valori un campo modificabile manualmente con la tastiera dall'utilizzatore.

Alla chiusura della scena i valori vengono salvati nel file .json come descritto nel capitolo relativo agli script e da quest'ultimo, volendo, è possibile invertire il processo per l'aggiornamento dei dati in Revit.

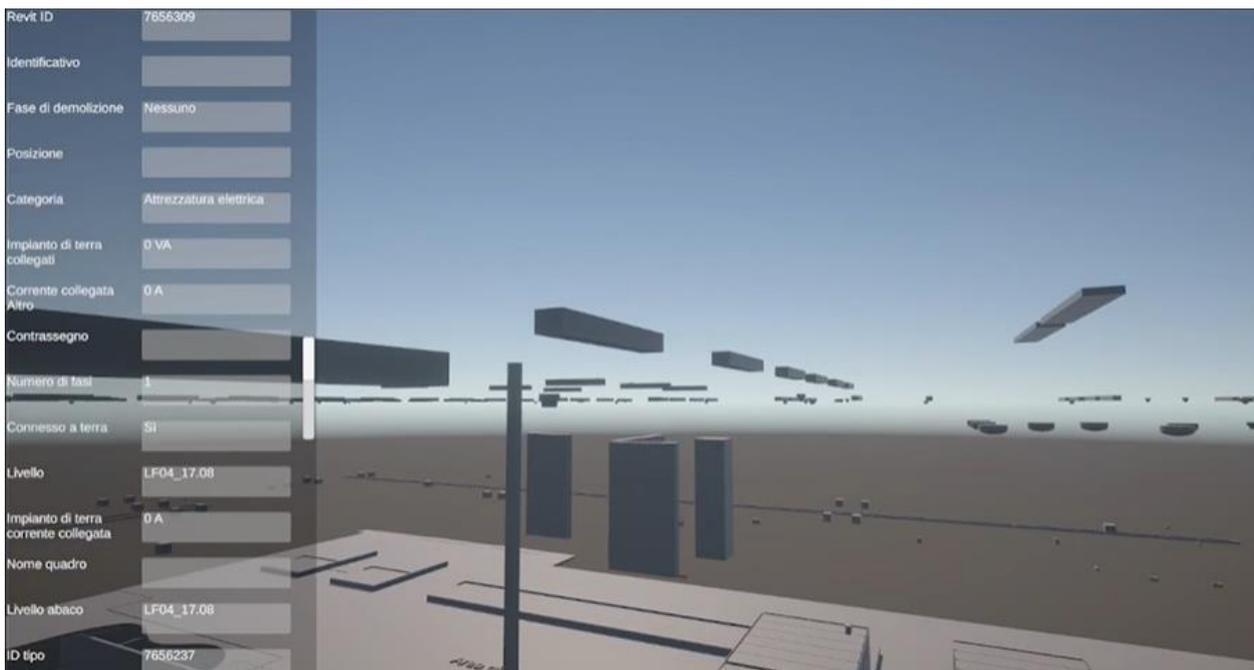


Figura 66: comparsa del pannello delle proprietà modificabili

6. Conclusione e sviluppi futuri

Questa tesi ha esplorato l'applicazione del Building Information Modeling (BIM) nella gestione e manutenzione dell'impianto di messa a terra e del conduttore di protezione del grattacielo della Regione Piemonte, fornendo una metodologia integrata che unisce la precisione della modellazione BIM con l'interattività della realtà virtuale. Il lavoro ha evidenziato come l'utilizzo di strumenti avanzati di visualizzazione e gestione dei dati possa migliorare l'efficienza operativa e la sicurezza, favorendo una gestione informata e reattiva degli impianti elettrici.

L'implementazione di script dedicati per la gestione dei parametri, il salvataggio dei dati e la visualizzazione interattiva in Unity ha permesso di creare un sistema completo e adattabile per visualizzare e monitorare i parametri critici degli impianti; questo sistema offre una visione approfondita delle condizioni e delle prestazioni dei componenti dell'impianto di messa a terra, con un'interfaccia intuitiva che permette un facile accesso alle informazioni per scopi di manutenzione e analisi, senza per forza essere esperti di progettazione elettrica.

Sicuramente quanto ottenuto ha delle criticità, a partire dal fatto di essere stato realizzato in una versione ormai obsoleta di Revit e quindi costellata di “escamotage” per aggirare le mancanze del software: in ambito elettrico le nuove versioni si sono evolute, offrendo strumenti e opzioni che in qualche circostanza sarebbero stati essenziali per il raggiungimento di alcuni degli obiettivi di questa trattazione. Una soluzione è sicuramente quella di aggiornare il modello, tenendo presente il tempo necessario a controllare le eventuali perdite di dati nella migrazione.

Il primo sviluppo futuro naturale è quella di proseguire il lavoro fin qui eseguito utilizzando la realtà aumentata, rendendo il sistema accessibile anche a tecnici in campo: attraverso dispositivi di AR/VR, sarebbe possibile visualizzare in tempo reale la posizione dei componenti nascosti, ricevere indicazioni di percorso per gli interventi e interagire con i parametri direttamente nell'ambiente operativo, migliorando la sicurezza e l'efficienza.

Un passo successivo potrebbe essere l'integrazione del sistema BIM con sensori IoT per raccogliere dati in tempo reale sui parametri dell'impianto, come correnti di dispersione, temperatura e umidità: questi dati, visualizzati nel modello BIM, permetterebbero una manutenzione predittiva e un controllo costante delle condizioni dell'impianto, riducendo i tempi di intervento a favore della sicurezza.

Considerato poi il periodo storico, risulta tutt'altro che fuori luogo l'idea di introdurre l'intelligenza artificiale in questo ambito, che sfrutterebbe algoritmi di “machine learning” in grado di imparare a identificare schemi di comportamento dell'impianto e a riconoscere le condizioni che precedono un guasto, come variazioni anomale della corrente o temperature fuori

norma: il sistema, monitorando costantemente i dati, potrebbe rilevare se e per quanto i conduttori di protezione sono stati percorsi da corrente, prevedendo la probabilità di guasti, problemi in specifici componenti e suggerendo interventi preventivi. Partendo da uno scadenziario come quello analizzato potrebbe addirittura riorganizzare la frequenza degli interventi di controllo, aumentando quelli riguardanti elementi sensibili e riducendo quelli relativi ad elementi più robusti, sempre nei limiti imposti dalla legge.

L'intelligenza artificiale potrebbe, in un altro scenario, fungere da sentinella monitorando il valore dell'impedenza dell'anello di guasto calcolato in fase di progetto, confrontandolo con il valore reale e segnalando al personale se e di quanto si sta uscendo dalla zona limite consentita dalla norma: non è una situazione così distante dalla realtà, basti pensare a quanti interventi vengono eseguiti sulle linee di alimentazione per le espansioni senza verificare se la protezione a monte continui ad essere adeguata.

Un altro impiego potrebbe essere quello di integrare il sistema BIM con un Building Management System (BMS), permettendo di monitorare e controllare non solo i parametri dell'impianto di messa a terra, bensì di tutte le apparecchiature elettriche. Si pensi come, in un sistema IT, nel caso di un primo guasto a terra si abbia la possibilità di ricevere gli allarmi in un sistema integrato anziché su un singolo dispositivo isolato e non sempre sorvegliato dal personale: anche qui, tutte le informazioni relative ai malfunzionamenti entrerebbero a fare parte del patrimonio informativo gestito dal BIM, arricchendo il modello e rendendo ancor più mirati gli interventi di manutenzione, ordinaria e straordinaria.

Tutto quanto appena detto, assieme ai risultati di questa tesi, rappresenta un passo verso una gestione sempre più digitale e interconnessa, che non solo migliora l'efficienza e la sicurezza, ma apre anche la strada verso un futuro di manutenzione predittiva e controllo automatizzato degli edifici.

Appendice 1

Identificativo	ABACO DISPOSITIVI CONNESSI A TERRA				Circuito di terra
	Connesso a terra	Livello	Locale	Posizione	
LI02_8.54					
Collettore CabBT	Si	LI02_8.54	Cabina BT	Muro	
QGBT.1.O Terra	Si	LI02_8.54	Cabina BT	QGBT.1.O	
QGBT.1.P15 Terra	Si	LI02_8.54	Cabina BT	QGBT.1.P15	
QGBT.1.SA Terra	Si	LI02_8.54	Cabina BT	QGBT.1.SA	
QGBT.1.SB Terra	Si	LI02_8.54	Cabina BT	QGBT.1.SB	
Centro Stella TR1	Si	LI02_8.54	Cabina MT	TR1	
Centro Stella TR2	Si	LI02_8.54	Cabina MT	TR2	
Centro Stella TR3	Si	LI02_8.54	Cabina MT	TR3	
Centro Stella TR4	Si	LI02_8.54	Cabina MT	TR4	
Centro Stella TR5	Si	LI02_8.54	Cabina MT	TR3	
Collettore CabMT	Si	LI02_8.54	Cabina MT	Muro	
MAGLIA DI TERRA	Si	LI02_8.54	Cabina MT	Interrato	
Massa TR1	Si	LI02_8.54	Cabina MT	TR1	
Massa TR2	Si	LI02_8.54	Cabina MT	TR2	
Massa TR3	Si	LI02_8.54	Cabina MT	TR3	
Massa TR4	Si	LI02_8.54	Cabina MT	TR4	
Massa TR5	Si	LI02_8.54	Cabina MT	TR5	
Picchetto	Si	LI02_8.54	Cabina MT	Interrato	
QGMT Terra	Si	LI02_8.54	Cabina MT	QGMT	
Rete Prot. Trafo	Si	LI02_8.54	Cabina MT		
LF04_17.08					
	Si	LF04_17.08			
	Si	LF04_17.08			
	Si	LF04_17.08			
	Si	LF04_17.08			
	Si	LF04_17.08			
Blindosbarra Verticale Terra	Si	LF04_17.08			
RI_24_LF04	Si	LF04_17.08			
TRP_TO_ELE_BD_AL_01_LF04_00093	Si	LF04_17.08			
TRP_TO_ELE_BD_AL_01_LF04_00094	Si	LF04_17.08			
TRP_TO_ELE_BD_AL_01_LF04_00095	Si	LF04_17.08			
TRP_TO_ELE_BD_AL_01_LF04_00096	Si	LF04_17.08			
TRP_TO_ELE_BD_AL_01_LF04_00097	Si	LF04_17.08			
TRP_TO_ELE_BD_AL_01_LF04_00098	Si	LF04_17.08			
TRP_TO_ELE_BD_AL_01_LF04_00099	Si	LF04_17.08			
TRP_TO_ELE_BD_AL_01_LF04_00100	Si	LF04_17.08			
TRP_TO_ELE_BD_AL_01_LF04_00101	Si	LF04_17.08			
TRP_TO_ELE_BD_AL_01_LF04_00102	Si	LF04_17.08			
TRP_TO_ELE_GP_AL_1+1_1_LF04_00007	Si	LF04_17.08			
TRP_TO_ELE_GP_AL_1+1_1_LF04_00008	Si	LF04_17.08			
TRP_TO_ELE_GP_AL_1+1_1_LF04_00009	Si	LF04_17.08			
TRP_TO_ELE_GP_AL_1+1_1_LF04_00010	Si	LF04_17.08			
TRP_TO_ELE_GP_AL_1+1_1_LF04_00011	Si	LF04_17.08			
TRP_TO_ELE_GP_AL_1+1_1_LF04_00012	Si	LF04_17.08			
TRP_TO_ELE_GP_AL_1+1_1_LF04_00013	Si	LF04_17.08			
TRP_TO_ELE_GP_AL_1+1_1_LF04_00014	Si	LF04_17.08			
TRP_TO_ELE_GP_AL_1+1_1_LF04_00015	Si	LF04_17.08			
TRP_TO_ELE_GP_AL_1+1_1_LF04_00016	Si	LF04_17.08			
TRP_TO_ELE_GP_AL_1+1_1_LF04_00017	Si	LF04_17.08			
TRP_TO_ELE_GP_AL_1+1_1_LF04_00018	Si	LF04_17.08			
TRP_TO_ELE_GP_AL_1+1_1_LF04_00019	Si	LF04_17.08			
TRP_TO_ELE_GP_AL_1+1_1_LF04_00020	Si	LF04_17.08			
TRP_TO_ELE_GP_AL_1+1_1_LF04_00021	Si	LF04_17.08			

T term O-3

TRP_TO_ELE_IM_SL_49W_LF04_00362	Si	LF04_17.08		
TRP_TO_ELE_IM_SL_49W_LF04_00363	Si	LF04_17.08		
TRP_TO_ELE_IM_SL_49W_LF04_00364	Si	LF04_17.08		
TRP_TO_ELE_IM_SL_49W_LF04_00365	Si	LF04_17.08		
TRP_TO_ELE_IM_SL_49W_LF04_00366	Si	LF04_17.08		
TRP_TO_ELE_IM_SL_49W_LF04_00367	Si	LF04_17.08		
TRP_TO_ELE_IM_SL_49W_LF04_00368	Si	LF04_17.08		
TRP_TO_ELE_IM_SL_49W_LF04_00369	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00333	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00334	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00335	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00336	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00337	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00338	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00339	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00340	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00341	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00342	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00343	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00344	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00345	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00346	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00347	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00348	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00349	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00350	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00351	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00352	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00353	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00354	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00355	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00356	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00357	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00358	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00359	Si	LF04_17.08		
TRP_TO_ELE_IR_SL_26W_LF04_00360	Si	LF04_17.08		
TRP_TO_ELE_PC_UN_IP_LF04_00001	Si	LF04_17.08		T term O-17
TRP_TO_ELE_PC_UN_IP_LF04_00002	Si	LF04_17.08		
TRP_TO_ELE_PC_UN_OR_LF04_00001	Si	LF04_17.08		
TRP_TO_ELE_PC_UN_OR_LF04_00002	Si	LF04_17.08		
TRP_TO_ELE_PC_UN_OR_LF04_00003	Si	LF04_17.08		
TRP_TO_ELE_PC_UN_OR_LF04_00004	Si	LF04_17.08		
TRP_TO_ELE_SP_AC_230V_LF04_00085	Si	LF04_17.08		
TRP_TO_ELE_SP_AC_230V_LF04_00086	Si	LF04_17.08		
TRP_TO_ELE_SP_AC_230V_LF04_00087	Si	LF04_17.08		
TRP_TO_ELE_SP_AC_230V_LF04_00088	Si	LF04_17.08		
TRP_TO_ELE_SP_AC_230V_LF04_00089	Si	LF04_17.08		
TRP_TO_ELE_SP_AC_230V_LF04_00091	Si	LF04_17.08		
TRP_TO_ELE_SP_AC_230V_LF04_00092	Si	LF04_17.08		
Collettore LT nord P04	Si	LF04_17.08	LT nord	Muro
QE.T.P04.01 O Terra	Si	LF04_17.08	LT nord	QE.T.P04.01 O
QE.T.P04.01 P15/P0 Terra	Si	LF04_17.08	LT nord	QE.T.P04.01 P15/P0
QE.T.P04.01 SA Terra	Si	LF04_17.08	LT nord	QE.T.P04.01 SA
TRP_TO_ELE_AF_AL_600x600_LF04_00084	Si	LF04_17.08	LT nord	
Collettore LT sud P04	Si	LF04_17.08	LT sud	Muro
QE.T.P04.01 SB Terra	Si	LF04_17.08	LT sud	QE.T.P04.01 SB

Appendice 2



ELT.AP.04-04	Raddrizzatore	Per Raddrizzatore si intende il dispositivo che durante il funzionamento in rete trasforma la tensione alternata che esce dal trasformatore di ingresso in tensione continua, alimentando, quindi, il caricabatteria e l'inverter
Verifica della corrente assorbita		3M
Verifica della tensione e corrente di mantenimento		3M
Verifica efficienza dei dispositivi di controllo e regolazione		3M
Controllo morsettiere e serraggio connessioni varie		3M
ELT.AP.04-05	Inverter	Per Inverter o invertitore si intende il dispositivo che trasforma la tensione continua del raddrizzatore o delle batterie in tensione alternata sinusoidale di ampiezza e frequenza costanti
Verificare lo stato di funzionamento del quadro di parallelo invertitori misurando alcuni parametri quali le tensioni, le correnti e le frequenze di uscita dall'inverter		2M
Effettuare le misurazioni della potenza in uscita su inverter-rete		2M

ELT.AP.05 Impianto di terra

L'impianto di messa a terra ha la funzione di collegare determinati punti, elettricamente definiti, con un conduttore a potenziale nullo ed è generalmente composto da collettore di terra, i conduttori equipotenziali, conduttore di protezione principale e quelli che raccordano i singoli impianti, da un sistema di dispersione e da pozzetti di ispezione

Codice attività	Elementi Manutenibili / Interventi	Frequenza
ELT.AP.05-01	Conduttori di protezione	Per Conduttori di protezione si intendono i conduttori di protezione principale o montanti che raccolgono i conduttori di terra dai piani dell'edificio
Verificare che i componenti (conduttori interrati, collettori di terra e masse metalliche, ecc.) siano in buone condizioni. Verificare inoltre che siano in buone condizioni i serraggi dei bulloni.		6M
Misura resistenza di isolamento di terra con strumento certificato. La prova intende verificare se l'isolamento dei cavi e delle relative connessioni sia rimasto adeguato nel tempo.		2A
Misura impedenza anello di guasto in fondo al circuito, cioè nel punto più lontano dal relativo dispositivo di protezione		2A
Verificare con controlli a campione che i conduttori di protezione arrivino fino al nodo equipotenziale		6M
Verificare l'intensità della corrente scaricata a terra dall'impianto		6M
Verificare la presenza ed il corretto posizionamento della cartellonistica di sicurezza		6M
Eseguire il controllo visivo per verificare l'integrità dell'impianto		A
Verificare il serraggio delle connessioni nei punti accessibili		A
Sostituire i componenti danneggiati o deteriorati o che presentano evidenti segni di ossidazione		A
Ripristinare quelle parti che non dovessero risultare in condizioni ottimali per il buon funzionamento dell'impianto		A
Verifica, pulizia e serraggio delle giunzioni e capicorda, ricoprire con pasta neutralizzante tutte le connessioni		A
Verificare con controlli a campione che i conduttori di protezione arrivino fino al nodo equipotenziale.		A

Classificazione del documento: Consip Public

Gara a procedura aperta, per l'affidamento di un Accordo Quadro avente ad oggetto i servizi di Facility Management

Grandi Immobili – Edizione 1 – ID 2077

Appendice 4 al Capitolato Tecnico – Elenco attività – NEW - NEW

Pag. 21 di 131

Verificare che i componenti (quali conduttori, ecc.) siano in buone condizioni. Verificare inoltre che siano in buone condizioni i serraggi dei bulloni		A
ELT.AP.05-02	Pozzetti	Per Pozzetti si intendono, appositi manufatti realizzati in calcestruzzo o in muratura, quasi sempre totalmente interrati, dotati di chiusini metallici per l'accesso dall'esterno che devono essere forniti di opportuni sistemi di chiusura
Verificare lo stato dei chiusini di accesso ai pozzetti controllando che siano facilmente removibili		6M
Eeguire una disincretazione dei chiusini di accesso ai pozzetti con prodotti sgrassanti		6M
Controllare l'integrità delle strutture individuando la presenza di eventuali anomalie come fessurazioni, disgregazioni, distacchi, riduzione del copriferro e relativa esposizione a processi di corrosione dei ferri d'armatura. Verifica dello stato del calcestruzzo e controllo del degrado e/o eventuali processi di carbonatazione		A
ELT.AP.05-03	Sistema di dispersione	Per Sistema di dispersione si intende l'insieme di un corpo conduttore o gruppi di corpi conduttori in contatto elettrico con il terreno e che realizzano un collegamento elettrico con la terra
Apertura dei pozzetti di terra; controllo dello stato dei collegamenti della rete di terra con i dispersori. In presenza di ossidazioni provvedere allo smontaggio dei collegamenti, alla rimozione dell'ossido, all'ingrassaggio ed al nuovo serraggio dei morsetti		A
Verificare che i componenti (quali connessioni, pozzetti, capicorda, ecc.) del sistema di dispersione siano in buone condizioni e non ci sia presenza di corrosione di detti elementi. Verificare inoltre la presenza dei cartelli indicatori degli schemi elettrici		A
Verificare che il tipo di giunzione e il tipo di contatto con il conduttore di terra sia conforme alle norme vigenti		A

ELT.AP.06 Impianto di protezione contro le scariche atmosferiche

L'impianto di protezione contro le scariche atmosferiche ha la funzione di proteggere gli utenti ed il sistema edilizio da scariche atmosferiche ed è generalmente costituito da un sistema di captazione, da calate, da un sistema di dispersione e da pozzetti di ispezione

Codice attività	Elementi Manutenibili / Interventi	Frequenza
ELT.AP.06-01	Sistema di captazione	Per sistema di captazione si intende l'insieme dei dispositivi di captazione che possono essere composti da diversi elementi e che sono combinabili tra di loro secondo necessità (aste, fili e funi tese, conduttori a maglie)
Prove, misure e ispezioni a vista secondo la Norme CEI 81-1, CEI 81-10; CEI 64-2, CEI EN 62305 e secondo Livello di protezione dell'LPS		A
Verifica che il sistema di captazione sia conforme al progetto (o alla norma) che tutti i componenti siano in buone condizioni ed atti a compiere le funzioni ad essi assegnate e che non vi sia corrosione o segni di deterioramento		A
Verificare non vi siano connessioni allentate o rotture accidentali nei conduttori e nelle giunzioni		A
Verificare che le connessioni visibili siano intatte e funzionalmente operanti e che i componenti visibili siano ancorati alla superficie di fissaggio		A
Verifica della stabilità degli ancoraggi e delle giunzioni ed eventuale serraggio dei bulloni		A
ELT.AP.06-02	Calate	Per Calate si intende il collegamento elettrico tra il dispositivo di captazione e l'impianto di messa a terra
Verificare che i componenti del sistema delle calate siano ben agganciati tra di loro, che i bulloni siano serrati e che vi siano gli anelli di collegamento		2A
Verificare che i componenti del sistema delle calate siano in buone condizioni e che siano stati disposti ad interasse medio di 25 m		2A
Verifica della stabilità degli ancoraggi e delle giunzioni ed eventuale serraggio dei bulloni		A

Classificazione del documento: Consip Public

Gara a procedura aperta, per l'affidamento di un Accordo Quadro avente ad oggetto i servizi di Facility Management Grandi Immobili – Edizione 1 – ID 2077

Appendice 4 al Capitolato Tecnico – Elenco attività – NEW - NEW

Pag. 22 di 131

Appendice 3 (script lettura circuiti)

C:\Users\Alessandro\Desktop\Codice Dynamo.cs

1

```
1 import clr
2 clr.AddReference('RevitAPI')
3 clr.AddReference('RevitServices')
4 from Autodesk.Revit.DB import *
5 from RevitServices.Persistence import DocumentManager
6
7 # Ottenere il documento attivo di Revit
8 doc = DocumentManager.Instance.CurrentDBDocument
9
10 # Input
11 element = UnwrapElement(IN[0]) # Elemento finale selezionato
12
13 def get_circuit_elements(element):
14     """
15     Ottiene tutti i circuiti associati all'elemento selezionato.
16     """
17     # Collezione tutti gli elementi della categoria Circuiti Elettrici
18     circuits = FilteredElementCollector(doc).OfCategory
19         (BuiltInCategory.OST_ElectricalCircuit).WhereElementIsNotElementType().ToElements()
20
21     # Filtra i circuiti associati all'elemento selezionato
22     associated_circuits = [circuit for circuit in circuits if element.Id in [e.Id for e in
23         circuit.Elements]]
24     return associated_circuits
25
26 # Recupera i circuiti associati all'elemento e restituisce i loro nomi
27 circuits = get_circuit_elements(element)
28 circuit_names = [(c.Id.IntegerValue, c.LookupParameter("Numero di circuito").AsString()) for
29     c in circuits]
30
31 # Output: lista degli ID e dei numeri di circuito associati
32 OUT = circuit_names
33
34 import clr
35 clr.AddReference('RevitAPI')
36 clr.AddReference('RevitServices')
37 from Autodesk.Revit.DB import *
38 from RevitServices.Persistence import DocumentManager
39
40 # Ottenere il documento attivo di Revit
41 doc = DocumentManager.Instance.CurrentDBDocument
42
43 # Input
44 element = UnwrapElement(IN[0]) # Elemento finale selezionato
45
46 def get_circuit_elements(element):
47     """
48     Ottiene tutti i circuiti associati all'elemento selezionato.
49     """
50     # Collezione tutti gli elementi della categoria Circuiti Elettrici
51     circuits = FilteredElementCollector(doc).OfCategory
52         (BuiltInCategory.OST_ElectricalCircuit).WhereElementIsNotElementType().ToElements()
53
54     # Filtra i circuiti associati all'elemento selezionato
55     associated_circuits = [circuit for circuit in circuits if element.Id in [e.Id for e in
56         circuit.Elements]]
57     return associated_circuits
58
59 # Recupera i circuiti associati all'elemento e restituisce i loro nomi
60 circuits = get_circuit_elements(element)
61 circuit_names = [(c.Id.IntegerValue, c.LookupParameter("Numero di circuito").AsString()) for
62     c in circuits]
```

```
57 # Output: lista degli ID e dei numeri di circuito associati
58 OUT = circuit_names
59 using System;
60
61 [Serializable]
62 public class KeyValuePair
63 {
64     public string key;
65     public string value;
66
67     // Costruttore per inizializzare la coppia chiave-valore
68     public KeyValuePair(string key, string value)
69     {
70         this.key = key;
71         this.value = value;
72     }
73 }
74
```

Appendice 4 (script rilevazione lunghezza)

C:\Users\Alessandro\Desktop\Codice Dynamo.cs

1

```
1 import clr
2 clr.AddReference('RevitAPI')
3 clr.AddReference('RevitServices')
4 from Autodesk.Revit.DB import *
5 from RevitServices.Persistence import DocumentManager
6
7 # Ottenere il documento attivo di Revit
8 doc = DocumentManager.Instance.CurrentDBDocument
9
10 # Input: circuito selezionato (ID e nome del circuito)
11 selected_circuit = IN[0] # Formato: [ID, nome circuito]
12
13 # Trova l'elemento circuito usando l'ID
14 circuit_id = ElementId(selected_circuit[0])
15 circuit_element = doc.GetElement(circuit_id)
16
17 # Recupera la lunghezza del circuito (in piedi) e la converte in metri
18 length_param = circuit_element.LookupParameter("Lunghezza")
19 if length_param:
20     length_in_meters = length_param.AsDouble() * 0.3048 # Conversione da piedi a metri
21 else:
22     length_in_meters = "Lunghezza non disponibile"
23
24 # Output: lunghezza del circuito in metri
25 OUT = length_in_meters
26 import clr
27 clr.AddReference('RevitAPI')
28 clr.AddReference('RevitServices')
29 from Autodesk.Revit.DB import *
30 from RevitServices.Persistence import DocumentManager
31
32 # Ottenere il documento attivo di Revit
33 doc = DocumentManager.Instance.CurrentDBDocument
34
35 # Input: circuito selezionato (ID e nome del circuito)
36 selected_circuit = IN[0] # Formato: [ID, nome circuito]
37
38 # Trova l'elemento circuito usando l'ID
39 circuit_id = ElementId(selected_circuit[0])
40 circuit_element = doc.GetElement(circuit_id)
41
42 # Recupera la lunghezza del circuito (in piedi) e la converte in metri
43 length_param = circuit_element.LookupParameter("Lunghezza")
44 if length_param:
45     length_in_meters = length_param.AsDouble() * 0.3048 # Conversione da piedi a metri
46 else:
47     length_in_meters = "Lunghezza non disponibile"
48
49 # Output: lunghezza del circuito in metri
50 OUT = length_in_meters
51
```

Appendice 5 (script rilevazione nome quadro)

C:\Users\Alessandro\Desktop\Codice Dynamo.cs

1

```
1 import clr
2 clr.AddReference('RevitAPI')
3 clr.AddReference('RevitServices')
4 from Autodesk.Revit.DB import *
5 from RevitServices.Persistence import DocumentManager
6
7 # Ottenere il documento attivo di Revit
8 doc = DocumentManager.Instance.CurrentDBDocument
9
10 # Input: circuito selezionato (ID e nome del circuito)
11 selected_circuit = IN[0] # Formato: [ID, nome circuito]
12
13 # Trova l'elemento circuito usando l'ID
14 circuit_id = ElementId(selected_circuit[0])
15 circuit_element = doc.GetElement(circuit_id)
16
17 # Recupera il parametro "Quadro" del circuito
18 board_param = circuit_element.LookupParameter("Quadro")
19 if board_param:
20     board_name = board_param.AsString()
21 else:
22     board_name = "Parametro 'Quadro' non disponibile"
23
24 # Output: nome del quadro associato
25 OUT = board_name
26 import clr
27 clr.AddReference('RevitAPI')
28 clr.AddReference('RevitServices')
29 from Autodesk.Revit.DB import *
30 from RevitServices.Persistence import DocumentManager
31
32 # Ottenere il documento attivo di Revit
33 doc = DocumentManager.Instance.CurrentDBDocument
34
35 # Input: circuito selezionato (ID e nome del circuito)
36 selected_circuit = IN[0] # Formato: [ID, nome circuito]
37
38 # Trova l'elemento circuito usando l'ID
39 circuit_id = ElementId(selected_circuit[0])
40 circuit_element = doc.GetElement(circuit_id)
41
42 # Recupera il parametro "Quadro" del circuito
43 board_param = circuit_element.LookupParameter("Quadro")
44 if board_param:
45     board_name = board_param.AsString()
46 else:
47     board_name = "Parametro 'Quadro' non disponibile"
48
49 # Output: nome del quadro associato
50 OUT = board_name
51
```

Appendice 6 (script dizionario sezione)

```
C:\Users\Alessandro\Desktop\Codice Dynamo.cs 1
1 # Dizionario di resistenza e reattanza per unità di lunghezza (in Ohm per metro) per ciascuna ▶
  sezione per cavi tripolari
2 valori_tripolari = {
3   "1": { "resistenza": 22.5 / 1000, "reattanza": 0.125 / 1000},
4   "1.5": { "resistenza": 15.1 / 1000, "reattanza": 0.118 / 1000},
5   "2.5": { "resistenza": 9.08 / 1000, "reattanza": 0.109 / 1000},
6   "4": { "resistenza": 5.68 / 1000, "reattanza": 0.101 / 1000},
7   "6": { "resistenza": 3.78 / 1000, "reattanza": 0.0955 / 1000},
8   "10": { "resistenza": 2.27 / 1000, "reattanza": 0.0861 / 1000},
9   "16": { "resistenza": 1.43 / 1000, "reattanza": 0.0817 / 1000},
10  "25": { "resistenza": 0.907 / 1000, "reattanza": 0.0813 / 1000},
11  "35": { "resistenza": 0.654 / 1000, "reattanza": 0.0783 / 1000},
12  "50": { "resistenza": 0.483 / 1000, "reattanza": 0.0779 / 1000},
13  "70": { "resistenza": 0.334 / 1000, "reattanza": 0.0751 / 1000},
14  "95": { "resistenza": 0.241 / 1000, "reattanza": 0.0762 / 1000},
15  "120": { "resistenza": 0.191 / 1000, "reattanza": 0.0740 / 1000},
16  "150": { "resistenza": 0.157 / 1000, "reattanza": 0.0745 / 1000},
17  "185": { "resistenza": 0.125 / 1000, "reattanza": 0.0742 / 1000},
18  "240": { "resistenza": 0.0966 / 1000, "reattanza": 0.0752 / 1000},
19  "300": { "resistenza": 0.0780 / 1000, "reattanza": 0.0750 / 1000},
20  "400": { "resistenza": 0.0625 / 1000, "reattanza": 0.0742 / 1000},
21  "500": { "resistenza": 0.0512 / 1000, "reattanza": 0.0744 / 1000},
22  "630": { "resistenza": 0.0417 / 1000, "reattanza": 0.0749 / 1000}
23 }
24 }
25
26 # Dizionario di resistenza e reattanza per unità di lunghezza (in Ohm per metro) per ciascuna ▶
  sezione per cavi unipolari
27 valori_unipolari = {
28  "1": { "resistenza": 22.1 / 1000, "reattanza": 0.176 / 1000},
29  "1.5": { "resistenza": 14.8 / 1000, "reattanza": 0.168 / 1000},
30  "2.5": { "resistenza": 8.91 / 1000, "reattanza": 0.155 / 1000},
31  "4": { "resistenza": 5.57 / 1000, "reattanza": 0.143 / 1000},
32  "6": { "resistenza": 3.71 / 1000, "reattanza": 0.135 / 1000},
33  "10": { "resistenza": 2.24 / 1000, "reattanza": 0.119 / 1000},
34  "16": { "resistenza": 1.41 / 1000, "reattanza": 0.112 / 1000},
35  "25": { "resistenza": 0.889 / 1000, "reattanza": 0.106 / 1000},
36  "35": { "resistenza": 0.641 / 1000, "reattanza": 0.101 / 1000},
37  "50": { "resistenza": 0.473 / 1000, "reattanza": 0.101 / 1000},
38  "70": { "resistenza": 0.328 / 1000, "reattanza": 0.0965 / 1000},
39  "95": { "resistenza": 0.236 / 1000, "reattanza": 0.0975 / 1000},
40  "120": { "resistenza": 0.188 / 1000, "reattanza": 0.0939 / 1000},
41  "150": { "resistenza": 0.153 / 1000, "reattanza": 0.0928 / 1000},
42  "185": { "resistenza": 0.123 / 1000, "reattanza": 0.0908 / 1000},
43  "240": { "resistenza": 0.0943 / 1000, "reattanza": 0.0902 / 1000},
44  "300": { "resistenza": 0.0761 / 1000, "reattanza": 0.0895 / 1000},
45  "400": { "resistenza": 0.0607 / 1000, "reattanza": 0.0876 / 1000},
46  "500": { "resistenza": 0.0496 / 1000, "reattanza": 0.0867 / 1000},
47  "630": { "resistenza": 0.0402 / 1000, "reattanza": 0.0865 / 1000}
48 }
49
50 # Input: tipo di cavo e sezione del cavo selezionata
51 tipo_cavo = IN[0] # "tripolare" o "unipolare"
52 sezione = IN[1] # Sezione scelta dal dropdown per il cavo
53
54 # Seleziona il dizionario corretto in base al tipo di cavo
55 if tipo_cavo == "tripolare":
56     dizionario_cavi = valori_tripolari
57 elif tipo_cavo == "unipolare":
58     dizionario_cavi = valori_unipolari
59 else:
60     OUT = "Errore: tipo di cavo non valido"
```

```
61     quit()
62
63 # Controlla e converte l'input della sezione in una stringa senza decimali inutili
64 if isinstance(sezione, float):
65     sezione = str(int(sezione)) if sezione.is_integer() else str(sezione)
66 elif isinstance(sezione, int):
67     sezione = str(sezione)
68 else:
69     sezione = str(sezione)
70
71 # Debug: stampa l'input ricevuto e verifica se è una stringa
72 print("Sezione ricevuta (formattata):", sezione)
73 print("Tipo di cavo:", tipo_cavo)
74 print("Chiavi del dizionario selezionato:", list(dizionario_cavi.keys()))
75
76 # Recupera i valori dal dizionario selezionato
77 if sezione in dizionario_cavi:
78     resistenza = dizionario_cavi[sezione]["resistenza"]
79     reattanza = dizionario_cavi[sezione]["reattanza"]
80 else:
81     resistenza = "Valore non disponibile"
82     reattanza = "Valore non disponibile"
83
84 # Debug: stampa i valori trovati
85 print("Resistenza:", resistenza)
86 print("Reattanza:", reattanza)
87
88 # Output separato: resistenza e reattanza per unità di lunghezza (in Ohm/metro)
89 OUT = resistenza, reattanza
90
```

Appendice 7 (script calcolo impedenza)

C:\Users\Alessandro\Desktop\Codice Dynamo.cs

1

```
1 import math
2
3 # Input: lista con resistenza e reattanza specifica (ohm/metro) e lunghezza (in metri)
4 valori_specifici = IN[0] # lista [resistenza_specifica, reattanza_specifica]
5 lunghezza = IN[1]      # lunghezza totale del cavo
6
7 # Verifica che valori_specifici sia una lista con due elementi
8 if isinstance(valori_specifici, list) and len(valori_specifici) == 2:
9     resistenza_specifica = valori_specifici[0]
10    reattanza_specifica = valori_specifici[1]
11 else:
12    # Se l'input non è corretto, restituisce un errore come output
13    OUT = "Errore: l'input 'valori_specifici' deve essere una lista con due valori."
14    quit()
15
16 # Calcola la resistenza totale e la reattanza totale
17 resistenza_totale = resistenza_specifica * lunghezza
18 reattanza_totale = reattanza_specifica * lunghezza
19
20 # Calcola il modulo dell'impedenza
21 impedenza_totale = math.sqrt(resistenza_totale**2 + reattanza_totale**2)
22
23 # Output separato per debug e verifica
24 OUT = resistenza_totale, reattanza_totale, impedenza_totale
25 import math
26
27 # Input: lista con resistenza e reattanza specifica (ohm/metro) e lunghezza (in metri)
28 valori_specifici = IN[0] # lista [resistenza_specifica, reattanza_specifica]
29 lunghezza = IN[1]      # lunghezza totale del cavo
30
31 # Verifica che valori_specifici sia una lista con due elementi
32 if isinstance(valori_specifici, list) and len(valori_specifici) == 2:
33     resistenza_specifica = valori_specifici[0]
34     reattanza_specifica = valori_specifici[1]
35 else:
36     # Se l'input non è corretto, restituisce un errore come output
37     OUT = "Errore: l'input 'valori_specifici' deve essere una lista con due valori."
38     quit()
39
40 # Calcola la resistenza totale e la reattanza totale
41 resistenza_totale = resistenza_specifica * lunghezza
42 reattanza_totale = reattanza_specifica * lunghezza
43
44 # Calcola il modulo dell'impedenza
45 impedenza_totale = math.sqrt(resistenza_totale**2 + reattanza_totale**2)
46
47 # Output separato per debug e verifica
48 OUT = resistenza_totale, reattanza_totale, impedenza_totale
49
```

Appendice 8

```
...o\Desktop\CODICI LETTORE CSV\V12 versione definitiva\LettoreCSV.cs 1
1 using System.Collections.Generic;
2 using System.IO;
3 using System.Text.RegularExpressions;
4 using UnityEngine;
5 using TMPro;
6
7 public class CsvReader : MonoBehaviour
8 {
9     public TextAsset csvFile; // File CSV da leggere
10    public TextMeshProUGUI uiText; // Testo UI per mostrare i dati
11    public ColliderAssigner colliderAssigner; // Riferimento allo script ColliderAssigner
12
13    private List<string> csvData = new List<string>(); // Lista delle righe CSV
14
15    // Lista per memorizzare i Revit ID che non hanno un GameObject associato
16    private List<int> missingRevitIDs = new List<int>();
17
18    void Start()
19    {
20        ReadCsv();
21        AssignDataToGameObjects();
22
23        // Leggo e poi assegno collider
24        if (colliderAssigner != null)
25        {
26            colliderAssigner.AssignColliders();
27        }
28        else
29        {
30            Debug.LogError("ColliderAssigner non assegnato. Assicurati di collegare il
31                componente.");
32        }
33    }
34
35    void ReadCsv()
36    {
37        if (csvFile != null)
38        {
39            using (StringReader reader = new StringReader(csvFile.text))
40            {
41                string line;
42                while ((line = reader.ReadLine()) != null)
43                {
44                    csvData.Add(line);
45                }
46            }
47        }
48        else
49        {
50            Debug.LogError("Il file CSV non è assegnato!");
51        }
52    }
53
54    void AssignDataToGameObjects()
55    {
56        foreach (string row in csvData)
57        {
58            // Estrae tutte le coppie chiave-valore dalla riga
59            Dictionary<string, string> parameters = ExtractParameters(row);
60
61            if (parameters.ContainsKey("Revit ID"))
62            {
```

```
62     int revitID = int.Parse(parameters["Revit ID"]);
63
64     // Filtra le viste 3D ma mantieni i dati per l'export
65     if (parameters.ContainsKey("Categoria") && parameters["Categoria"].Contains ▶
        ("Vista 3D"))
66     {
67         Debug.Log($"Vista 3D {revitID} ignorata nella scena, ma mantenuta per ▶
            l'export.");
68         continue; // Salta la creazione del GameObject, mantiene i dati
69     }
70
71     // Trova il GameObject con il Revit ID corrispondente
72     GameObject obj = FindGameObjectByRevitID(revitID);
73
74     if (obj != null)
75     {
76         DynamicParameters dynParams = obj.GetComponent<DynamicParameters>();
77         if (dynParams == null)
78         {
79             dynParams = obj.AddComponent<DynamicParameters>();
80         }
81
82         dynParams.parameters = parameters;
83
84         // Chiamare UpdateInspectorData per aggiornare le liste visibili ▶
            nell'Inspector
85         dynParams.UpdateInspectorData();
86     }
87     else
88     {
89         // Aggiunge l'ID mancante alla lista
90         missingRevitIDs.Add(revitID);
91     }
92 }
93
94
95 // Mostra un messaggio riepilogativo alla fine del processo
96 if (missingRevitIDs.Count > 0)
97 {
98     Debug.LogWarning($" {missingRevitIDs.Count} GameObjects non trovati per i ▶
        seguenti Revit ID: {string.Join(", ", missingRevitIDs)}");
99 }
100 }
101
102 Dictionary<string, string> ExtractParameters(string row)
103 {
104     // Crea un dizionario per memorizzare le coppie chiave-valore
105     Dictionary<string, string> parameters = new Dictionary<string, string>();
106
107     // Usa un'espressione regolare per trovare tutte le coppie "Chiave: Valore"
108     Regex regex = new Regex(@"(\w[\w\s]*):s*([^\s,]+)");
109     MatchCollection matches = regex.Matches(row);
110
111     foreach (Match match in matches)
112     {
113         string key = match.Groups[1].Value.Trim(); // La chiave (es: "Revit ID")
114         string value = match.Groups[2].Value.Trim(); // Il valore (es: "212664")
115         parameters[key] = value;
116     }
117
118     return parameters;
119 }
```

```
120
121     GameObject FindGameObjectByRevitID(int revitID)
122     {
123         // Trova tutti i GameObject nella scena
124         GameObject[] allObjects = FindObjectsOfType<GameObject>();
125
126         foreach (GameObject obj in allObjects)
127         {
128             // Verifica se il nome del GameObject contiene il codice tra parentesi quadre
129             string objName = obj.name;
130             int extractedID = ExtractIDFromName(objName);
131
132             if (extractedID == revitID)
133             {
134                 return obj;
135             }
136         }
137         return null;
138     }
139
140     int ExtractIDFromName(string name)
141     {
142         // Cerca il codice tra parentesi quadre nel nome del GameObject
143         Regex regex = new Regex(@"\[([d+)]");
144         Match match = regex.Match(name);
145         if (match.Success)
146         {
147             return int.Parse(match.Groups[1].Value);
148         }
149
150         return -1; // Ritorna -1 se non viene trovato un codice valido
151     }
152 }
153
```

Appendice 9

C:\Users\Alessandro\Desktop\KeyValuePair.cs

1

```
1 using System;
2
3 [Serializable]
4 public class KeyValuePair
5 {
6     public string key;
7     public string value;
8
9     // Costruttore per inizializzare la coppia chiave-valore
10    public KeyValuePair(string key, string value)
11    {
12        this.key = key;
13        this.value = value;
14    }
15 }
16
```

Appendice 10

C:\Users\Alessandro\Desktop\DynamicParameters.cs

1

```
1 using System.Collections.Generic;
2 using UnityEngine;
3 using System.IO;
4
5 [System.Serializable]
6 public class DynamicParameters : MonoBehaviour
7 {
8     public Dictionary<string, string> parameters = new Dictionary<string, string>();
9     [SerializeField] public List<KeyValuePair> keyValuePairs = new List<KeyValuePair>();
10
11     private string saveFilePath;
12
13     private void Start()
14     {
15         // Sostituisci i caratteri non validi solo per il nome del file
16         string safeFileName = gameObject.name.Replace("\\", "").Replace("/", "").Replace
17             (":", "").Replace(" ", "")
18             .Replace("?", "").Replace("\", "").Replace("<",
19             "")
20             .Replace(">", "").Replace("|", "");
21         // Usa il safeFileName solo per il percorso di salvataggio
22         saveFilePath = Path.Combine(Application.persistentDataPath, $"{safeFileName}
23             _parameters.json");
24
25         if (parameters == null)
26         {
27             parameters = new Dictionary<string, string>();
28         }
29
30         if (keyValuePairs == null)
31         {
32             keyValuePairs = new List<KeyValuePair>();
33         }
34
35         LoadParameters();
36     }
37
38     public void UpdateInspectorData()
39     {
40         if (keyValuePairs == null)
41         {
42             keyValuePairs = new List<KeyValuePair>();
43         }
44
45         keyValuePairs.Clear();
46         if (parameters != null)
47         {
48             foreach (var param in parameters)
49             {
50                 keyValuePairs.Add(new KeyValuePair(param.Key, param.Value));
51             }
52         }
53
54     private void OnApplicationQuit()
55     {
56         SaveParameters();
57     }
58
59     private void SaveParameters()
60     {
61     }
```

```
60     if (keyValuePairs == null || keyValuePairs.Count == 0)
61     {
62         Debug.LogWarning("keyValuePairs è vuoto, nessun dato da salvare.");
63         return;
64     }
65
66     if (parameters == null)
67     {
68         parameters = new Dictionary<string, string>();
69     }
70     else
71     {
72         parameters.Clear();
73     }
74
75     foreach (var kvp in keyValuePairs)
76     {
77         if (!string.IsNullOrEmpty(kvp.key))
78         {
79             parameters[kvp.key] = kvp.value;
80         }
81         else
82         {
83             Debug.LogWarning("Trovata una chiave vuota o nulla, ignorata.");
84         }
85     }
86
87     if (parameters.Count > 0)
88     {
89         // Serializza usando la classe modificata SerializableDictionary
90         SerializableDictionary serializableDict = new SerializableDictionary      ↗
91             (parameters);
92         string json = JsonUtility.ToJson(serializableDict, true);
93         File.WriteAllText(saveFilePath, json);
94     }
95     else
96     {
97         Debug.LogWarning("Il dizionario parameters è vuoto dopo il trasferimento, nessun ↗
98             dato salvato.");
99     }
100
101     private void LoadParameters()
102     {
103         if (File.Exists(saveFilePath))
104         {
105             string json = File.ReadAllText(saveFilePath);
106
107             SerializableDictionary savedData = JsonUtility.FromJson<SerializableDictionary> ↗
108                 (json);
109
110             if (savedData != null && savedData.keys.Count > 0)
111             {
112                 parameters = savedData.ToDictionary();
113                 UpdateInspectorData();
114             }
115             else
116             {
117                 Debug.LogWarning("Nessun parametro salvato trovato nel file.");
118             }
119         }
120     }
```

```
119         else
120         {
121             Debug.LogWarning($"Nessun file di parametri trovato per {gameObject.name}");
122         }
123     }
124 }
125
126 [System.Serializable]
127 public class SerializableDictionary
128 {
129     public List<string> keys = new List<string>();
130     public List<string> values = new List<string>();
131
132     public SerializableDictionary(Dictionary<string, string> dictionary)
133     {
134         foreach (var kvp in dictionary)
135         {
136             keys.Add(kvp.Key);
137             values.Add(kvp.Value);
138         }
139     }
140
141     // Metodo per riconvertire le liste in un dizionario
142     public Dictionary<string, string> ToDictionary()
143     {
144         Dictionary<string, string> dict = new Dictionary<string, string>();
145
146         for (int i = 0; i < keys.Count; i++)
147         {
148             dict[keys[i]] = values[i];
149         }
150
151         return dict;
152     }
153 }
154
```

Appendice 11

C:\Users\Alessandro\Desktop\DynamicParametersEditor.cs

1

```
1 using UnityEditor;
2 using UnityEngine;
3
4 [CustomEditor(typeof(DynamicParameters))]
5 public class DynamicParametersEditor : Editor
6 {
7     public override void OnInspectorGUI()
8     {
9         // Ottiene riferimento al componente
10        DynamicParameters dynamicParameters = (DynamicParameters)target;
11
12        // Titolo
13        EditorGUILayout.LabelField("Parametri Dinamici", EditorStyles.boldLabel);
14
15        // Se la lista è vuota, mostra un messaggio
16        if (dynamicParameters.keyValuePairs == null || dynamicParameters.keyValuePairs.Count == 0)
17        {
18            EditorGUILayout.LabelField("Nessun parametro disponibile.");
19            return;
20        }
21
22        // Crea una tabella con due colonne
23        EditorGUILayout.BeginHorizontal();
24        EditorGUILayout.LabelField("Key", EditorStyles.boldLabel, GUILayout.MaxWidth(200));
25        EditorGUILayout.LabelField("Value", EditorStyles.boldLabel);
26        EditorGUILayout.EndHorizontal();
27
28        // Cicla attraverso tutti i key-value pair
29        for (int i = 0; i < dynamicParameters.keyValuePairs.Count; i++)
30        {
31            var kvp = dynamicParameters.keyValuePairs[i];
32
33            EditorGUILayout.BeginHorizontal();
34
35            // Colonna "Key" (solo in lettura)
36            EditorGUILayout.LabelField(kvp.key, GUILayout.MaxWidth(200));
37
38            // Colonna "Value" (modificabile)
39            kvp.value = EditorGUILayout.TextField(kvp.value);
40
41            EditorGUILayout.EndHorizontal();
42
43            // Aggiorna il valore nella lista dopo la modifica
44            dynamicParameters.keyValuePairs[i] = kvp;
45        }
46
47        // Applica eventuali modifiche
48        if (GUI.changed)
49        {
50            EditorUtility.SetDirty(target);
51        }
52    }
53 }
```

Appendice 12

C:\Users\Alessandro\Desktop\ColliderAssigner.cs

1

```
1 using UnityEngine;
2
3 public class ColliderAssigner : MonoBehaviour
4 {
5     public string excludedNamePart = "Modello di Revit collegato"; // Parte del nome dei
        GameObject da escludere
6
7     public void AssignColliders()
8     {
9         // Trova tutti i GameObject nella scena
10        GameObject[] allObjects = FindObjectsOfType<GameObject>();
11
12        foreach (GameObject obj in allObjects)
13        {
14            // Verifica se il nome del GameObject contiene la stringa da escludere
15            if (obj.name.Contains(excludedNamePart))
16            {
17                continue; // Salta l'oggetto che contiene "Modello di Revit collegato" nel
                nome
18            }
19
20            // Verifica la scala dell'oggetto
21            Vector3 scale = obj.transform.localScale;
22
23            // Usa un MeshCollider se la scala è negativa
24            if (scale.x < 0 || scale.y < 0 || scale.z < 0)
25            {
26                MeshCollider meshCollider = obj.AddComponent<MeshCollider>();
27                meshCollider.convex = true; // Assicurati che il MeshCollider sia convesso
                per il raycast
28            }
29            else
30            {
31                // Aggiungi un BoxCollider se la scala è positiva
32                if (obj.GetComponent<Collider>() == null)
33                {
34                    obj.AddComponent<BoxCollider>();
35                }
36            }
37        }
38    }
39 }
40
```

Appendice 13

C:\Users\Alessandro\Desktop\CameraController.cs

1

```
1 using UnityEngine;
2
3 public class CameraController : MonoBehaviour
4 {
5     public float mouseSensitivity = 100f; // Sensibilità del mouse
6     public float moveSpeed = 10f;        // Velocità di movimento della camera
7
8     private float xRotation = 0f;        // Rotazione sull'asse X per il movimento
9     private float yRotation = 0f;        // Rotazione sull'asse Y per il movimento
10    private float zRotation = 0f;        // Rotazione sull'asse Z per il movimento
11
12    void Start()
13    {
14        // Mostra il cursore e consente di usarlo nel Play Mode
15        Cursor.visible = true;
16        Cursor.lockState = CursorLockMode.None;
17    }
18
19    void Update()
20    {
21        /*// Debug.Log per verificare se Unity riceve gli input da tastiera e mouse
22        Debug.Log("Mouse X: " + Input.GetAxis("Mouse X") + ", Mouse Y: " + Input.GetAxis
23            ("Mouse Y"));
24        Debug.Log("Horizontal (A/D): " + Input.GetAxis("Horizontal") + ", Vertical (W/S): " +
25            Input.GetAxis("Vertical"));*/
26
27        // Controllo della rotazione della camera con il mouse
28        float mouseX = Input.GetAxis("Mouse X") * mouseSensitivity * Time.deltaTime;
29        float mouseY = Input.GetAxis("Mouse Y") * mouseSensitivity * Time.deltaTime;
30
31        xRotation -= mouseY;
32        xRotation = Mathf.Clamp(xRotation, -90f, 90f); // Limita la rotazione verticale
33
34        // Rotazione della camera
35        transform.localRotation = Quaternion.Euler(xRotation, 0f, 0f);
36        transform.parent.Rotate(Vector3.up * mouseX);
37
38        // Controllo del movimento della camera con WASD
39        float moveX = Input.GetAxis("Horizontal") * moveSpeed * Time.deltaTime; // Movimento
40        float moveZ = Input.GetAxis("Vertical") * moveSpeed * Time.deltaTime; // Movimento
41        float moveY = Input.GetAxis("Mouse Y") * moveSpeed * Time.deltaTime; // Movimento
42        Vector3 move = transform.right * moveX + transform.forward * moveZ + transform.up * moveY;
43        transform.parent.position += move;
44    }
45 }
```

Appendice 14

C:\Users\Alessandro\Desktop\TabletUI.cs

1

```
1 using UnityEngine;
2 using TMPro; // Per TextMeshPro
3 using UnityEngine.UI; // Per ScrollRect
4 using System.Collections.Generic;
5
6 public class TabletUI : MonoBehaviour
7 {
8     public Transform content; // Il contenitore dei campi nel tablet
9     public GameObject parameterFieldPrefab; // Prefab per visualizzare i parametri
10    public ScrollRect scrollRect; // Riferimento allo Scroll Rect
11
12    public void ShowParameters(DynamicParameters dynParams)
13    {
14        if (content == null || parameterFieldPrefab == null)
15        {
16            Debug.LogError("Contenitore o Prefab non assegnato nell'Inspector.");
17            Debug.LogError("Content status: " + (content == null ? "Null" : "Assigned"));
18            Debug.LogError("Prefab status: " + (parameterFieldPrefab == null ? "Null" :
19                "Assigned"));
20        }
21        return;
22    }
23
24    Debug.Log("Prefab attuale: " + parameterFieldPrefab.name);
25
26    // Rimuove tutti i figli esistenti prima di aggiungere i nuovi parametri
27    foreach (Transform child in content)
28    {
29        Destroy(child.gameObject);
30    }
31
32    // Cicla attraverso i parametri e li visualizza nel tablet
33    foreach (var kvp in dynParams.keyValuePair)
34    {
35        GameObject field = Instantiate(parameterFieldPrefab, content);
36
37        // Recupera il campo per ParamName e lo lascia invariato
38        TextMeshProUGUI paramName = field.transform.Find
39            ("ParamName").GetComponent<TextMeshProUGUI>();
40        paramName.text = kvp.key;
41
42        // Recupera il campo per ParamValue come Input Field, anziché TextMeshProUGUI
43        TMP_InputField paramValueInputField = field.transform.Find
44            ("ParamValue").GetComponent<TMP_InputField>();
45
46        if (paramValueInputField != null)
47        {
48            // Log del valore che si sta cercando di assegnare
49            Debug.Log($"Impostazione del valore per {kvp.key}: {kvp.value}");
50
51            // Imposta il valore iniziale nell'Input Field
52            paramValueInputField.text = kvp.value;
53
54            // Aggiunge un listener per aggiornare il valore quando viene modificato
55            paramValueInputField.onValueChanged.AddListener((newValue) =>
56            {
57                kvp.value = newValue; // Aggiorna il valore nella lista keyValuePairs
58                dynParams.parameters[kvp.key] = newValue; // Aggiorna anche nel
59                    dizionario parameters
60            });
61
62            Debug.Log($"Valore aggiornato per {kvp.key}: {newValue}");
63        }
64    }
65 }
```

```
59         else
60         {
61             Debug.LogError("paramValueInputField è null!");
62         }
63     }
64
65     // Chiama il metodo per forzare l'aggiornamento dello Scroll Rect
66     ForceScrollRectUpdate();
67 }
68
69 // Definisci il metodo ForceScrollRectUpdate qui fuori da ShowParameters
70 void ForceScrollRectUpdate()
71 {
72     // Forza l'aggiornamento del Canvas per aggiornare il layout
73     Canvas.ForceUpdateCanvases();
74
75     // Posizione dello scroll sia in alto
76     scrollRect.verticalNormalizedPosition = 1f;
77
78     // Forza di nuovo l'aggiornamento del Canvas
79     Canvas.ForceUpdateCanvases();
80 }
81 }
82
```

Appendice 15

C:\Users\Alessandro\Desktop\ObjectClickHandler.cs

1

```
1 using UnityEngine;
2 using UnityEngine.EventSystems; // Per il controllo UI
3
4 public class ObjectClickHandler : MonoBehaviour
5 {
6     public Camera mainCamera; // Riferimento alla telecamera principale
7     public LayerMask clickableLayer; // Layer dei GameObject cliccabili (opzionale)
8     public GameObject tabletPanel; // Il pannello del tablet da attivare quando si clicca
9     public TabletUI tabletUI; // Riferimento allo script che gestisce il tablet
10
11 void Update()
12 { // Verifica se il clic è su un oggetto UI
13     if (EventSystem.current.IsPointerOverGameObject())
14     {
15         // Se clic su un oggetto UI, non fare nulla
16         Debug.Log("Clic su UI ignorato");
17         return;
18     }
19     // Verifica se l'utente ha cliccato con il tasto sinistro del mouse
20     if (Input.GetMouseButton(0))
21     {
22         // Crea un ray dal punto di clic del mouse sulla scena
23         Ray ray = mainCamera.ScreenPointToRay(Input.mousePosition);
24         RaycastHit hit;
25
26         // Esegue un raycast per verificare se colpisce un GameObject
27         if (Physics.Raycast(ray, out hit, Mathf.Infinity, clickableLayer.value == 0 ? ~0 : clickableLayer))
28         {
29             // Debug.Log per verificare che il raycast stia colpendo qualcosa
30             Debug.Log($"Hai cliccato su: {hit.collider.gameObject.name}");
31
32             // Verifica se il GameObject colpito ha un componente DynamicParameters
33             DynamicParameters dynParams = hit.collider.gameObject.GetComponent<DynamicParameters>();
34
35             if (dynParams != null)
36             {
37                 Debug.Log($"Oggetto selezionato: {hit.collider.gameObject.name}");
38
39                 // Controlla che tabletPanel e tabletUI siano assegnati correttamente
40                 if (tabletUI != null && tabletPanel != null)
41                 {
42                     // Attiva il pannello del tablet e aggiorna con i dati del GameObject selezionato
43                     tabletPanel.SetActive(true);
44                     tabletUI.ShowParameters(dynParams);
45                 }
46                 else
47                 {
48                     Debug.LogError("Il tabletUI o il tabletPanel non sono assegnati correttamente nell'Inspector.");
49                 }
50             }
51             else
52             {
53                 Debug.LogWarning($"L'oggetto {hit.collider.gameObject.name} non ha il componente DynamicParameters.");
54             }
55         }
56     }
57 }
```

```
58         Debug.LogWarning("Nessun GameObject colpito dal raycast.");
59     }
60 }
61 }
62 }
63
```


Bibliografia e sitografia

- [1] Infobuild, «La rivoluzione digitale dell'edilizia: il BIM o Building Information Modeling,» 5 Marzo 2021. [Online]. Available: <https://www.infobuild.it/approfondimenti/bim-rivoluzione-digitale-edilizia-building-information-modeling/>. [Consultato il giorno 9 Ottobre 2024].
- [2] BibLus Acca, «Vantaggi dei BIM: come il Building Information Modeling cambia il nostro modo di lavorare,» 1 Giugno 2016. [Online]. Available: <https://biblus.acca.it/come-il-bim-cambia-il-nostro-modo-di-lavorare/>. [Consultato il giorno 9 Ottobre 2024].
- [3] buildingSmart International, «What is openBIM?,» [Online]. Available: <https://www.buildingsmart.org/about/openbim/openbim-definition/>. [Consultato il giorno 9 Ottobre 2024].
- [4] Adhox srl, «Il formato IFC: cos'è? A cosa serve?,» 10 Novembre 2022. [Online]. Available: <https://adhox.it/formato-ifc/>. [Consultato il giorno 9 Ottobre 2024].
- [5] National Bim Standard-United States, «National BIM Standard-United States® Version 4,» [Online]. Available: <https://www.nibs.org/nbims/v4>. [Consultato il giorno 10 Ottobre 2024].
- [6] Cabinet Office, «Government Construction Strategy,» 31 Maggio 2011. [Online]. Available: https://assets.publishing.service.gov.uk/media/5a78ce8eed915d07d35b2933/Government-Construction-Strategy_0.pdf. [Consultato il giorno 10 Ottobre 2024].
- [7] International Standard, «ISO 19650-1:2018,» 2018. [Online]. Available: <https://www.iso.org/standard/68078.html>. [Consultato il giorno 10 Ottobre 2024].
- [8] Statsbygg, «BIM manual,» [Online]. Available: https://dok.statsbygg.no/wp-content/uploads/2020/06/statsbyggs-bim-manual-1-2-1_en_20131217.pdf. [Consultato il giorno 10 Ottobre 2024].
- [9] Senate Properties, «Senate Properties and BIM,» 24 Novembre 2008. [Online]. Available: https://media.rilem.se/2015/12/Sem-2008_5_Karjalainen_Open-BIM-Senate-Properties-and-BIM.pdf. [Consultato il giorno 10 Ottobre 2024].
- [10] Parlamento Europeo e Consiglio, «DIRETTIVA 2014/24/UE,» 26 Febbraio 2014. [Online]. Available: <https://www.mit.gov.it/normativa/direttiva-201424ue>. [Consultato il giorno 10 Ottobre 2024].
- [11] Gazzetta Ufficiale della Repubblica Italiana, «DECRETO LEGISLATIVO 18 aprile 2016, n. 50 Attuazione delle direttive 2014/23/UE, 2014/24/UE e 2014/25/UE sull'aggiudicazione dei contratti di concessione, sugli appalti pubblici e sulle procedure d'appalto degli enti erogatori nei settori dell'acqua, dell,» 19 Aprile 2016. [Online]. Available: <https://www.gazzettaufficiale.it/eli/id/2016/04/19/16G00062/sg>. [Consultato il giorno 10 Ottobre 2024].
- [12] Ministero delle Infrastrutture e dei Trasporti, «Decreto Ministeriale numero 560 del 01/12/2017,» 1 Dicembre 2017. [Online]. Available: <https://www.mit.gov.it/nfsmitgov/files/media/normativa/2018-01/Decreto%20Ministro%20MIT%20n.%20560%20del%201.12.2017.pdf>. [Consultato il giorno 10 Ottobre 2024].
- [13] Ministero delle Infrastrutture e dei trasporti, «Decreto Ministeriale numero 312 del 02/08/2021,» 2 Agosto 2021. [Online]. Available: https://www.mit.gov.it/nfsmitgov/files/media/normativa/2021-08/DM_2021-08-02_BIM.pdf. [Consultato il giorno 10 Ottobre 2024].
- [14] Gazzetta ufficiale della Repubblica Italiana, «DECRETO LEGISLATIVO 31 marzo 2023, n. 36 Ripubblicazione del testo del decreto legislativo 31 marzo 2023, n. 36, recante: «Codice dei contratti pubblici in attuazione dell'articolo 1 della legge 21 giugno 2022, n. 78, recante delega al Governo in materia,» 31 Marzo 2023. [Online]. Available:

- <https://www.gazzettaufficiale.it/eli/id/2023/04/13/23A02179/sg>. [Consultato il giorno 10 Ottobre 2024].
- [15] UNI, «UNI 11337-1:2017 e parti seguenti,» 26 Gennaio 2017. [Online]. Available: <https://store.uni.com/uni-11337-1-2017>. [Consultato il giorno 10 Ottobre 2024].
- [16] BibLus Acca, «Che cos'è la ISO 19650,» 14 Luglio 2023. [Online]. Available: <https://biblus.acca.it/iso-19650/>. [Consultato il giorno 10 Ottobre 2024].
- [17] F. R. Cecconi, S. Maltese e M. Garramone, «3.1.2 Plain Language Questions (PLQs),» in *Capitolati informativi e attributi dei modelli BIM*, Maggioli Editore, 2024, p. 160.
- [18] 01Building, «Viaggio fra i livelli di maturità dell'approccio Bim,» 24 Dicembre 2020. [Online]. Available: <https://www.01building.it/bim/livelli-maturita-approccio-bim/>. [Consultato il giorno 10 Ottobre 2024].
- [19] Nti, «Le 10 dimensioni del BIM: quali sono e cosa significano,» Novembre 2023. [Online]. Available: <https://www.nti-group.com/it/blog/it/2023/10-dimensioni-bim-quali-sono/>. [Consultato il giorno 11 Ottobre 2024].
- [20] Politecnico di Torino, «Linee Guida, BIM per il Facility Management, Caso studio: NUOVO PALAZZO PER UFFICI REGIONE PIEMONTE,» 2022.
- [21] V. Cataliotti e A. Campoccia, Impianti di terra, TuttoNormel, 2003.
- [22] Comitato Elettrotecnico Italiano, «CEI EN 50522-Impianti di terra per impianti elettrici con tensione superiore a 1 kV in c.a.,» Novembre 2022. [Online]. Available: <https://mycatalogo.ceinorme.it/cei/item/0000019000?sso=y>. [Consultato il giorno 10 Novembre 2024].
- [23] Comitato Elettrotecnico Italiano, «CEI 0-16;V3/EC2-Regola tecnica di riferimento per la connessione di Utenti attivi e passivi alle reti AT ed MT delle imprese distributrici di energia elettrica,» Marzo 2024. [Online]. Available: <https://mycatalogo.ceinorme.it/cei/item/0010020082>. [Consultato il giorno 10 Novembre 2024].
- [24] Comitato Elettrotecnico Italiano, «CEI 64-8-Impianti elettrici a tensione nominale non superiore a 1 000 V in corrente alternata e a 1 500 V in corrente continua,» Luglio 2024. [Online]. [Consultato il giorno 10 Novembre 2024].
- [25] BibLus Acca, «Testo unico sulla Sicurezza [PDF] – D.Lgs. 81/08 (versione 2024),» 2024. [Online]. Available: <https://biblus.acca.it/download/testo-unico-sicurezza-dlgs-81-2008/#toggle-id-2-closed>. [Consultato il giorno 11 Novembre 2024].
- [26] Gazzetta ufficiale della Repubblica Italiana, «DECRETO DEL PRESIDENTE DELLA REPUBBLICA 22 ottobre 2001, n. 462 Regolamento di semplificazione del procedimento per la denuncia di installazioni e dispositivi di protezione contro le scariche atmosferiche, di dispositivi di messa a terra di impianti elett,» 22 Ottobre 2001. [Online]. Available: <https://www.gazzettaufficiale.it/eli/id/2002/01/08/001G0521/sg>. [Consultato il giorno 11 Novembre 2024].
- [27] Gazzetta ufficiale della Repubblica Italiana, «DECRETO LEGISLATIVO 12 aprile 2006, n. 163 Codice dei contratti pubblici relativi a lavori, servizi e forniture in attuazione delle direttive 2004/17/CE e 2004/18/CE,» 12 Aprile 2006. [Online]. Available: <https://www.gazzettaufficiale.it/eli/id/2006/05/02/006G0184/sg>. [Consultato il giorno 11 Novembre 2024].
- [28] Comitato Elettrotecnico Italiano, «CEI-UNEL 35023-Cavi di energia per tensione nominale U uguale a 1 kV - Cadute di tensione,» [Online]. Available: <https://mycatalogo.ceinorme.it/cei/item/0000017459?sso=y>. [Consultato il giorno 11 Novembre 2024].

- [29] xr_erz, «VR-READY (Autodesk App Store),» [Online]. Available:
<https://apps.autodesk.com/RVT/it/Detail/Index?id=293813844256407121&appLang=en&os=Win64>.
- [30] TuttoNormel, Guida Blu N.13 • CABINE MT/BT, Torino: Edizioni TNE, 2013.
- [31] TuttoNormel, N.9 • VERIFICHE, Torino: Edizioni TNE, 2012.