

# Ricerca di Sistema elettrico



**Predisposizione degli ambienti digitale e  
sperimentale propedeutici alle attività di sviluppo e  
testing (LA1.14)**

A. Ricca, V. Sorrentino, A. Merola,  
V. Capaldo, M. Valenti

RdS\_PTR 22-24\_PR 2.3\_LA1.1\_064

LA1.14 PREDISPOSIZIONE DEGLI AMBIENTI DIGITALE E SPERIMENTALE PROPEDEUTICI ALLE ATTIVITÀ DI SVILUPPO E TESTING

A. Ricca (ENEA), V. Sorrentino (ENEA), A. Merola (ENEA), V. Capaldo (ENEA), M. Valenti (ENEA)

Giugno 2023

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dell'Ambiente e della Sicurezza Energetica - ENEA  
Piano Triennale di Realizzazione 2022-2024

Obiettivo: *Decarbonizzazione/Digitalizzazione ed evoluzione delle reti*

Progetto: *Evoluzione, pianificazione, gestione ed esercizio delle reti elettriche*

Linea di attività: 1.14

Responsabile del Progetto: *Maria Valenti, ENEA*

Responsabile Linea di Attività: *Antonio Ricca, ENEA*

Mese inizio previsto: 1

Mese inizio effettivo: 1

Mese fine previsto: 18

Mese fine effettivo: 18

## Indice

1	RISULTATI ATTESI .....	3
2	RISULTATI OTTENUTI.....	4
3	PRODOTTI ATTESI.....	5
4	PRODOTTI SVILUPPATI .....	5
5	ANALISI DEGLI SCOSTAMENTI SU ATTIVITÀ E RISULTATI .....	6
6	SINTESI DELLE ATTIVITÀ SVOLTE .....	6
7	DETTAGLIO DELLE ATTIVITÀ SVOLTE.....	7
8	CONTRIBUTO DELLE EVENTUALI CONSULENZE ALLE ATTIVITÀ SOPRA DESCRITTE.....	11
9	PUBBLICAZIONI SCIENTIFICHE.....	11
10	EVENTI DI DISSEMINAZIONE .....	11
A.	APPENDICE .....	12

# 1 Risultati attesi

## Lista dei risultati attesi come da capitolato vigente

Di seguito si riporta lista dei risultati attesi come da capitolato vigente.

- Test funzionali alla verifica del corretto setup delle apparecchiature
- Caratteristiche della connessione NanoGrid ENEA – LabZERO POLIBA
- Architettura del tool
- Analisi e identificazione dei moduli Open Source per l'esecuzione dei Power Flow
- Analisi e identificazione dei linguaggi di programmazione Open Source per lo sviluppo del tool.

## 2 Risultati ottenuti

*Lista dei risultati ottenuti (Evidenziare in che misura il risultato è stato ottenuto ed il beneficio per il sistema elettrico nazionale e i suoi utenti. Aggiungere eventuali risultati ottenuti non previsti nel capitolato)*

La LA1.14 è stata focalizzata sulla predisposizione degli ambienti digitale e sperimentale propedeutici alle attività di sviluppo e testing. Le attività svolte nella LA1.14 sono raggruppabili in tre sotto attività: definizione della struttura del software per la fruizione dei modelli sviluppati all'interno del progetto, predisposizione dell'ambiente sperimentale per le caratterizzazioni elettriche e termiche dei componenti di rete e definizione delle caratteristiche della connessione tra la Nanogrid sperimentale ENEA e il laboratorio LabZERO di POLIBA. Per quanto concerne la definizione della **struttura del software**, il risultato atteso è stato ampiamente raggiunto, in quanto è stata definita l'architettura di base del software (Figura A.2), di cui sono stati individuati gli elementi principali ed i flussi logici tra di essi. È stato quindi definito il relativo albero di navigazione (Figura A.3) che evidenzia le interazioni tra i moduli e definisce la base dell'interfaccia grafica del tool stesso, nonché gli input e output ad esso afferenti. L'architettura sviluppata, basata su una ponderata scelta di soluzioni open-source in termini di linguaggio di sviluppo e di tool per il calcolo dei Load flow, è basata su un approccio modulare che rende il software scalabile, facilitando l'estensione del tool con funzionalità custom da parte degli utenti avanzati. Essa, inoltre, adotterà un'interfaccia di tipo user-friendly per facilitare la fruizione da parte dell'utente finale.

La **predisposizione dell'ambiente sperimentale** per le caratterizzazioni elettriche e termiche dei componenti di rete è stata finalizzata attraverso due ambiti principali. In relazione al primo ambito, si è proceduto, dapprima, all'acquisto dell'estensione del sistema Hardware In the Loop (HIL) integrativo e alle attrezzature e materiali integrativi necessari e, successivamente, alla configurazione e connessione degli stessi ai dispositivi preesistenti nei laboratori della NanoGrid SGRE di ENEA. A valle di tale installazione, si è proceduto a valutare l'efficacia delle configurazioni e dei relativi sistemi di comunicazione attraverso test dedicati, volti a verificare: l'effettivo invio dei comandi imposti dal sistema HIL, la corretta lettura delle grandezze di interesse, i tempi transitori della comunicazione. La procedura definita e adottata per i test è di seguito dettagliata (paragrafo 7.1). In parallelo, è stata avviata la progettazione di una vasca sperimentale per la caratterizzazione di cavi e giunti elettrici con l'obiettivo specifico di studiarne il comportamento al variare della temperatura e dell'umidità. Questo tipo di studio consentirà, in ottica prospettica, di ottenere dati sperimentali sul comportamento dei componenti delle reti elettriche in presenza di eventi riconducibili al cambiamento climatico (es. le ondate di calore). La LA1.1, infatti, ha evidenziato l'indisponibilità di questo tipo di dati e, più in generale, di dati per l'esecuzione delle analisi previste dal progetto con specifico riferimento ai componenti delle reti elettriche lato distribuzione in bassa tensione. Le attività condotte offriranno informazioni rilevanti ai fini della conduzione di studi simili sia in regime simulativo che emulativo da parte di utenti avanzati della filiera energetica (enti di ricerca, produttori di tecnologie connesse al sistema di rete, etc.).

In relazione alla terza sotto attività - la **connessione tra la nanogrid sperimentale ENEA ed il laboratorio LabZERO di POLIBA** - è stata definita la modalità da applicare e, in particolare, è stato identificato sia il protocollo di comunicazione, sia la necessità di un Tunnel VPN IPsec. Le soluzioni individuate, che saranno sviluppate nella LA1.16, consentiranno di interconnettere ambienti emulativi remoti e disomogenei (e.g. Typhoon HIL e Opal-RT), problematica rilevante ai fini della realizzazione delle reti elettriche negli scenari energetici futuri.

Le attività condotte hanno portato ai seguenti risultati.

### **PREDISPOSIZIONE DELL'AMBIENTE SPERIMENTALE**

- Acquisto, installazione e connessione delle apparecchiature per la predisposizione dell'ambiente sperimentale e progettazione preliminare di una vasca sperimentale per la caratterizzazione di cavi e giunti elettrici

- Test funzionali alla verifica del corretto setup delle apparecchiature di emulazione costituenti l'ambiente di testing (carichi elettronici, simulatore di rete, generatori fotovoltaici, batterie, fuel-cell, carico rigenerativo) e della relativa comunicazione con il sistema HIL (corretto invio dei comandi imposti dal sistema HIL, corretta lettura delle grandezze di interesse, valutazione dei tempi transitori della comunicazione).

## **SVILUPPO DEL SOFTWARE**

- **Definizione dell'architettura del tool**
  - Definizione delle macro-aree del software: Interfaccia grafica; Acquisizione delle reti da analizzare, Dizionario degli elementi; Funzionalità del tool; Operatore DSS;
  - Definizione dei flussi logici principali: Il core dell'architettura è il dizionario degli elementi, a cui afferiscono tutti i moduli del tool per attingere le informazioni sulla rete e per scriverne e condividerne i risultati;
  - Progettazione preliminare del dizionario delle reti: il dizionario ha una struttura ad albero, in cui in un primo livello si trovano gli elementi di rete; per ognuno di questi sono presenti sotto-dizionari relativi alla loro caratterizzazione, ed alle funzionalità (studi) disponibili.
- **Analisi e identificazione dei moduli Open Source per l'esecuzione dei Power Flow**
  - Analisi dei moduli PyDSS, PandaPower e OpenDSS ai fini della selezione del modulo più idoneo. La scelta è ricaduta su OpenDSS in quanto capace di analisi avanzate di reti sbilanciate, dotato di una propria interfaccia grafica per l'implementazione delle reti, e già in uso dai partner e nella comunità scientifica.
- **Analisi e identificazione dei linguaggi di programmazione Open Source per lo sviluppo del tool**
  - Analisi di diversi linguaggi di tipo open-source, tra cui Python, Julia, Java, C e Fortran ai fini della selezione del linguaggio di idoneo alle specifiche di progetti. Per lo sviluppo del software si è scelto il Python per la sua ottima integrazione con OpenDSS, per poter sfruttare alcuni moduli già implementati nel precedente Progetto 2.7 (Prg 10 – PTR2019-2021), e per la sua diffusione nel mondo scientifico.

## **CONNESSIONE TRA LA NANOGRID SPERIMENTALE ENEA ED IL LABORATORIO LABZERO DI POLIBA**

- Identificazione della tipologia di connessione tra i laboratori NanoGrid ENEA e LabZERO POLIBA, a valle dell'analisi delle caratteristiche degli strumenti dedicati presenti nei due laboratori ed individuazione di eventuali strumenti hardware e software per rendere efficace la connessione.

## **3 Prodotti attesi**

Lista dei prodotti hardware/software eventualmente attesi per la LA

Per la presente LA non sono attesi prodotti hardware/software.

## **4 Prodotti sviluppati**

Lista dei prodotti hardware/software eventualmente sviluppati nella LA, illustrando, per il software, le modalità di accesso per gli utenti (*Aggiungere eventuali prodotti sviluppati non previsti nel capitolato*)

La LA1.14 non prevede lo sviluppo di prodotti hardware/software.

## 5 Analisi degli scostamenti su attività e risultati

**(8000 caratteri max)**

Descrivere le motivazioni di eventuali scostamenti tecnici/economici rispetto al preventivo e criticità riscontrate (*Evidenziare il contenuto in riferimento al piano di rischi presentato*)

Non si sono registrati scostamenti tecnico e/o economici nell'ambito della LA1.14.

## 6 Sintesi delle attività svolte

**(1000 caratteri max)**

Sintesi delle attività svolte e dei risultati ottenuti in relazione ai risultati attesi.

Nella LA1.14 si è proceduto a: i) predisporre le infrastrutture di laboratorio per l'esecuzione di test di emulazione e per le caratterizzazioni dei componenti di rete; ii) condurre test funzionali sulle apparecchiature costituenti l'ambiente di testing, al fine di verificare le connessioni tra i dispositivi di laboratorio e le possibili relative configurazioni funzionali; iii) definire la connessione tra la NanoGrid ENEA e il LabZERO di POLIBA per le attività di emulazione congiunta; iv) progettare l'architettura di un software per lo studio di adeguatezza, affidabilità, e resilienza di sistemi di potenza, incentrata su un dizionario della rete contenente le caratteristiche della rete e degli studi ad essa associati; v) selezionare il modulo Open Source OpenDSS, largamente diffuso nel mondo scientifico per l'esecuzione dei PowerFlow; vi) scegliere quale linguaggio di programmazione il Python per facilità di integrazione con OpenDSS e con i moduli già sviluppati all'interno di OraTool.

## 7 Dettaglio delle attività svolte

(15000 caratteri max)

Descrivere in dettaglio le attività svolte nella LA (*Evidenziare come si sono ottenuti i risultati. Descrivere brevemente anche le attività, per le quali si sono spese delle risorse, che tuttavia non hanno portato all'ottenimento dei risultati previsti al fine di permettere la corretta valutazione di congruità e pertinenza dei costi rendicontati.*)

### 7.1 Test funzionali alla verifica del corretto setup e della comunicazione delle apparecchiature

Uno degli obiettivi della LA1.14 riguarda la predisposizione delle infrastrutture di laboratorio per l'esecuzione di test previsti dalle attività della LA1.16.

In relazione alle facilities di laboratorio per la simulazione/emulazione/caratterizzazione elettrica, sono state condotte le seguenti attività:

- **Verifica dei dispositivi di laboratorio:** È stata effettuata una ricognizione delle attrezzature fisicamente presenti nei laboratori della NanoGrid SGRE dell'ENEA di Portici, valutandone l'adeguatezza rispetto alle procedure di caratterizzazione ipotizzate, e determinando la necessità di integrare le stesse con unità o moduli aggiuntivi.
- **Identificazione degli accoppiamenti dispositivi/tecnologie da emulare:** Nella LA1.1 sono state definite alcune tecnologie di particolare impatto nello studio di adeguatezza, sicurezza e resilienza delle reti elettriche, quali i sistemi di generazione fotovoltaica ed eolica, i sistemi basati sull'idrogeno (gli elettrolizzatori e le fuel cell PEM), ed i sistemi di accumulo elettrico mediante batterie. Gli accoppiamenti dei dispositivi con le tecnologie da studiare sono riportati in Tabella A.1.
- **Configurazione dei dispositivi:** È stata definita la configurazione di ogni dispositivo al fine di simulare efficacemente la tecnologia ad esso assegnato. Per ogni tecnologia è stato individuato il dispositivo, la modalità con la quale questo deve essere connesso alla rete di laboratorio (AC/DC, LV/MV, etc.) e i parametri funzionali del dispositivo stesso necessari a permettere l'adeguata emulazione della tecnologia assegnata.
- **Adeguamento del sistema di Hardware-In-the-Loop:** La necessità di connettere in tempo reale tali dispositivi, al fine di simulare una rete caratterizzata da un avanzato grado di complessità, ha reso necessario l'estensione del sistema Typhoon HIL presente nel laboratorio SGRE della sede ENEA di Portici attraverso l'acquisto di un Core Extended HIL604 della Typhoon HIL.
- **Creazione della rete di collegamento:** Una volta configurato il sistema per l'Hardware-In-the-Loop, si è reso necessario eseguire il collegamento fisico dello stesso con i dispositivi dedicati alla simulazione delle unità di rete attraverso collegamenti sia di tipo analogico (per garantire una adeguata velocità di settaggio dei set-point dei dispositivi) che attraverso protocollo TCP-IP (dedicati alla configurazione dei parametri operativi delle unità). In aggiunta, per assicurare una accurata misurazione di tensione, corrente e potenza in ingresso e/o in uscita dalle unità di simulazione, sono stati utilizzati dei dispositivi di energy meter della Yokogawa di tipo monofase WT310EH e trifase WT333E, collegati attraverso connessioni TCP-IP al computer di simulazione.
- **Configurazione delle interfacce di comunicazione:** Sono state sviluppate apposite interfacce SCADA in ambiente Typhoon HIL capaci di importare i parametri di input e le grandezze di output da monitorare. Le interfacce SCADA, specifiche per ogni unità di emulazione, sono state realizzate nell'ambiente di sviluppo model-based disponibile all'interno del software di modellazione di Typhoon HIL. Lo sviluppo dell'algorithm di interfaccia è risultato particolarmente oneroso, a causa della eterogeneità dei protocolli di comunicazione dei dispositivi: in particolare, si è reso necessario utilizzare protocolli di comunicazione ModBus e SCPI, oltre che i protocolli proprietari della Typhoon HIL per i segnali analogici, creando apposite classi di comunicazione con le unità hardware.

Il dettaglio delle configurazioni dei dispositivi di emulazione per le tecnologie considerate è riportato in Tabella A.1.

- **Test di verifica del setup:** L'efficacia delle configurazioni e dei relativi sistemi di comunicazione è stata verificata attraverso test dedicati, volti a valutare l'effettivo invio dei comandi imposti dal sistema HIL e le letture delle grandezze di interesse, nonché dei tempi transitori tipici dei dispositivi. La procedura definita per ognuno dei dispositivi utilizzati, costituita dai seguenti step:
  1. Configurazione di una rete elementare, contenente l'unità da esaminare, sull'ambiente Typhoon HIL;
  2. Configurazione dei dispositivi per l'emulazione della tecnologia di interesse;
  3. Inserimento dell'algoritmo delle interfacce SCADA per il controllo e monitoraggio dell'unità da esaminare, e per l'acquisizione dei dati dal relativo energy meter;
  4. Verifica delle connessioni elettriche di potenza dell'hardware di emulazione;
  5. Impostazione della potenza prelevata / immessa dall'unità hardware da esaminare;
  6. Verifica delle letture di misurazione dell'unità e dell'energy meter, e confronto (ove possibile) delle stesse;
  7. Valutazione dei tempi transitori.

I test condotti per le diverse unità hanno confermato l'idoneità del setup realizzato, producendo una eccellente ripetibilità delle misure.

In relazione alla **caratterizzazione termica**, che prevede il monitoraggio dell'andamento delle prestazioni di un determinato componente se soggetto a cicli termici (o, più generalmente, climatici) definiti da apposite procedure sperimentali, è stata eseguita una ricognizione delle apparecchiature presenti nella sede ENEA di Portici, funzionali a tali test. Nello specifico, è stata individuata una camera climatica SU340 della ACS, con la quale condizionare temperatura e umidità dell'ambiente di test, oltre che dispositivi per la rivelazione puntuale della temperatura, della corrente e della tensione o eventuali altre grandezze di interesse.

## 7.2 Definizione delle caratteristiche della connessione NanoGrid ENEA – LabZERO POLIBA

Al fine di realizzare una connessione efficace tra i laboratori NanoGrid/ENEA e LabZERO/POLIBA per supportare le attività di emulazione congiunta, è stata identificata e definita la modalità di collegamento tra i due laboratori, prevedendo un Tunnel VPN IPSec che permetta lo scambio in sicurezza di messaggi UDP tra i simulatori real-time. Il protocollo UDP è stato scelto sulla base delle esperienze pregresse di POLIBA nella co-simulazione remota e real-time. Il protocollo UDP è un protocollo asincrono, leggero e adatto alla comunicazione real-time poiché non prevede controlli sulla ricezione dei pacchetti da parte del destinatario. La co-simulazione prevede un dialogo tra un simulatore Typhoon (ENEA) e un simulatore OPAL RT (Politecnico di Bari). È stata quindi ravvisata la necessità di un middleware per tradurre i messaggi inviati dal Typhoon nel formato utilizzato da OPAL e viceversa. Nello specifico, è stata individuata la piattaforma VILLASframework, capace di connettere di più nodi in strutture di co-simulazione remota multi-sito.

## 7.3 Definizione dell'architettura del tool

Il software che verrà sviluppato nella LA1.15 permetterà lo studio di adeguatezza, affidabilità e resilienza dei sistemi di potenza, mettendo a disposizione degli utenti i modelli sviluppati nelle linee di attività precedenti, nonché i relativi output. Lo strumento deve avere una interfaccia semplice ed esaustiva che renda chiare ed accessibili tutte le sue funzionalità. Inoltre, un approccio open-source, consentirà l'eventuale espansione con ulteriori funzionalità sviluppate in futuro.

È stata quindi ideata una struttura di massima del software capace di consentire una programmazione modulare molto agevole: l'architettura (Figura A.2) individua i macro-moduli principali del software, e la rete dei flussi logici ad essi inerente.

Una simile architettura permette un rapido accesso a tutte le risorse della rete in esame da parte delle singole funzionalità del software nonché delle azioni di interfaccia utente; inoltre facilita lo scambio dei risultati tra i diversi moduli e le diverse funzionalità del software, permettendo così l'esecuzione di studi con un alto grado di complessità.

Una forma semplificata dell'albero di navigazione del software è riportata in Figura A.3. All'avvio del software, l'utente dovrà scegliere l'origine della rete da studiare: caricare una rete già salvata, crearne una ex novo, o scegliere una rete benchmark disponibile; in quest'ultimo caso, sarà richiesta anche la selezione di uno scenario. Il software, quindi, popolerà il dizionario con le caratteristiche della rete. L'utente potrà poi modificare la rete, se necessario. L'utente ora sceglie l'operazione da effettuare (LoadFlow, Optimal Power Flow, Calcolo di affidabilità e adeguatezza della rete, Grid management, Optimal Network Reconfiguration). Fatta eccezione per il LoadFlow, che darà i relativi risultati, a seconda dello studio scelto, sarà necessario inserire i relativi parametri peculiari del tipo di studio (per il singolo componente e/o per l'intera rete e/o globali); i parametri verranno memorizzati nel dizionario. Lo studio selezionato potrà necessitare dell'esecuzione di un LoadFlow per eseguire la procedura di calcolo. I risultati saranno mostrati all'utente e saranno memorizzati nel dizionario. Sarà possibile esportare il report della rete studiata, effettuare ulteriori studi (indipendenti dal precedente, o includendone quanto ottenuto), o terminare il software.

Tutti i dati relativi alla rete ed ai suoi componenti, nonché i risultati dei diversi studi, saranno salvati in un "dizionario" di rete, che contiene le caratteristiche della rete e dei relativi studi. In particolare, il dizionario conterrà:

- informazioni riguardo gli elementi presenti nel modello di rete da analizzare, quali i parametri, le connessioni, le caratteristiche elettriche e fisiche;
- informazioni inerenti alle condizioni simulative dello studio, quali set-point puntuali o profili dei carichi e dei generatori, o parametri specifici degli altri studi;
- risultati degli studi specifici applicati alla rete in esame

Il dizionario ha una struttura ad albero, in cui al primo livello sono elencati tutti gli elementi della rete. Per ogni elemento, saranno definiti dei sotto-dizionari atti a caratterizzarli all'interno della rete e degli studi effettuati.

Il sotto-dizionario "par" definisce le caratteristiche topologiche e i parametri specifici del componente, nonché le sue connessioni con gli altri elementi. Gli altri sotto-dizionari afferiscono agli studi specifici integrati all'interno del software; in particolare, per ogni sotto-dizionario (e quindi per ogni studio) l'elemento disporrà di una sezione in cui inserire i parametri dello studio, ed una in cui verranno riportati i risultati dello studio. L'utilizzo dei dizionari degli elementi facilita l'esportazione dello use-case configurato, nonché dei risultati in esso ottenuto.

## 7.4 Analisi e identificazione dei moduli Open Source per l'esecuzione dei Power Flow

Lo studio della simulazione delle reti elettriche sarà affidato a componenti esterni integrabili all'interno del tool. La loro funzione sarà quella di assicurare calcoli di LoadFlow puntuali o su intervalli temporali per scenari che spaziano dalle reti di trasmissione alle reti di distribuzione fino alle smart grid. Sono stati individuati tre tool open-source capaci di soddisfare le esigenze richieste: PandaPower, PyPSA e OpenDSS.

**PyPSA** è uno strumento operante in ambiente Python, capace di simulare ed ottimizzare sistemi energetici moderni e le loro funzionalità avanzate. È capace di trattare reti multi-vettoriali, permettendo una facile conversione tra i diversi vettori energetici.

**PandaPower** è uno strumento implementato ed operante in ambiente Python, largamente impiegato anche nello studio di sistemi sia di distribuzione simmetrici che di trasmissione. È in grado di risolvere power flow e optimal power flow di reti bilanciate e sbilanciate, per studi puntuali e su un dominio temporale.

**OpenDSS** è uno strumento completo per la simulazione di sistemi elettrici. Sviluppato per l'esecuzione stand-alone, dispone di librerie atte a poter essere integrato in diversi ambienti di lavoro (Python, VBA, C#, MatLab, Typhoon HIL, etc.) Riesce a simulare funzionalità avanzate dei componenti di rete e permette valutazioni

particolarmente complesse quali l'analisi delle armoniche ed inter-armoniche, della risposta della frequenza nei trasformatori, e delle perdite nelle reti sbilanciate.

Alla luce delle caratteristiche dei diversi moduli, riassunte in Tabella A.2, la scelta del tool di riferimento è ricaduta su OpenDSS sia perché risulta già largamente diffuso presso i partner del progetto, sia per la migliore gestione delle reti sbilanciate, sia per la capacità di gestire in maniera nativa gli elementi di storage, sia per la sua flessibilità di integrazione, in particolare con i sistemi Typhoon-HIL, permettendo una rilevante possibilità di espansione futura per il software che verrà sviluppato.

## 7.5 Analisi e identificazione dei linguaggi di programmazione Open Source per lo sviluppo del tool

Sono state analizzate diverse piattaforme Open Source per lo sviluppo del software:

**Python** è un linguaggio di programmazione, di “alto livello”, multi-paradigma, a tipizzazione dinamica. Il forte supporto di una solida e numerosa comunità di sviluppatori ha permesso un'ampia varietà di librerie (p.e., panda, numpy, scipy, etc.), ambienti di sviluppo, ecosistemi (p.e., Anaconda), affermandosi fortemente fra la comunità di ricercatori come strumento per il calcolo scientifico, soprattutto in contrapposizione a soluzioni commerciali a pagamento.

**Julia** è un linguaggio di programmazione di “alto livello”, multi-paradigma, a tipizzazione statica. È specificamente orientato al calcolo scientifico ed è altamente interoperabile con altri linguaggi di programmazione. La non elevata maturità tecnologica lo penalizza nel numero di librerie “native”.

**Java** è un linguaggio di programmazione di “alto livello”, multi-paradigma, a tipizzazione statica, largamente diffuso per la realizzazione di applicativi. Non pensato per applicazioni di calcolo scientifico, è caratterizzato da alcuni limiti (numeri complessi, e interi senza segno, array multidimensionali).

**C** è un linguaggio di programmazione di “medio livello”, procedurale, a tipizzazione statica. Ha una elevata maturità tecnologica ed una amplissima diffusione in molti ambiti grazie alle funzionalità che permettono di sfruttare le caratteristiche della macchina su cui viene compilato. Nonostante la disponibilità di numerose librerie dedicate, l'utilizzo di C per il calcolo scientifico non è consigliato (tranne che per esigenze particolari). Alla luce delle caratteristiche dei diversi linguaggi di programmazione, riassunti nella Tabella A.3, è stato individuato il Python come linguaggio di programmazione più idoneo, per sfruttare il comprovato background dei partner, i moduli già sviluppati all'interno di OraTool, e la facilità di integrazione con OpenDSS e più in generale con tutti i tool di calcolo DSS.

## 8 Contributo delle eventuali consulenze alle attività sopra descritte

L'attività non ha previsto il ricorso a consulenze.

## 9 Pubblicazioni scientifiche

### Elenco delle pubblicazioni scientifiche eventualmente risultanti dall'attività svolta

Le attività della LA1.14 sono state oggetto di disseminazione nel seguente lavoro scientifico:

*Antonio Ricca, Giovanna Adinolfi, Roberto Ciavarella, Giorgio Graditi, Maria Valenti. Innovative software for reliability, resilience, security and adequacy assessment of AC and DC grids and microgrids. AEIT 2023 Conference Proceedings. ISBN 978-88-87237-60-3.*

L'articolo è stato predisposto a giugno 2023 e, a valle di revisioni conseguenti alle richieste di revisione da parte del Comitato Scientifico della Conferenza, sottomesso nella sua versione definitiva il 15 settembre 2023.

## 10 Eventi di disseminazione

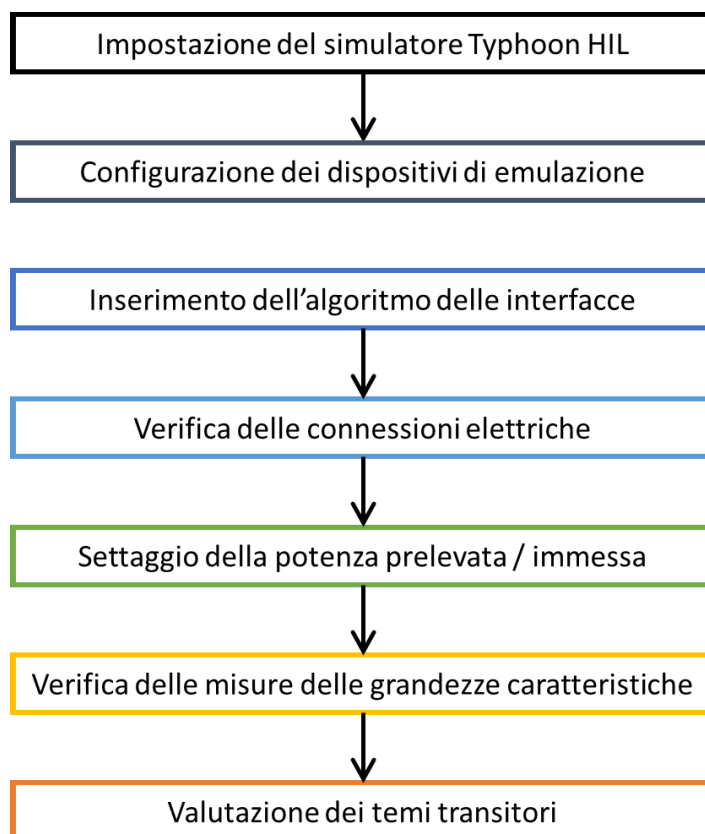
### Lista degli eventi di disseminazione eventualmente scaturiti dall'attività svolta

L'attività svolta non è stata oggetto di eventi di disseminazione nel SAL e i relativi risultati sono stati diffusi attraverso la pubblicazione di cui al precedente paragrafo.

## A. Appendice

In Figura A.1 è riportata la sequenza delle operazioni necessarie alla validazione delle configurazioni di ogni dispositivo al fine di verificare la capacità di emulazione delle tecnologie assegnate. La procedura è costituita da 7 step:

1. Impostazione del simulatore Typhoon HIL: l'ambiente di emulazione viene predisposto, impostando una rete elementare contenente la tecnologia da esaminare;
2. Configurazione dei dispositivi di emulazione: agendo sia sulla parte hardware che sulla parte software (menu di configurazione), il dispositivo viene predisposto per poter dialogare in maniera efficace sia con l'ambiente Typhoon sia con gli altri elementi di controllo;
3. Inserimento dell'algoritmo delle interfacce: all'interno dell'ambiente di emulazione, viene inserito l'algoritmo delle interfacce SCADA in modo da avere il controllo ed il monitoraggio dell'unità in esame;
4. Verifica delle connessioni elettriche: tutte le connessioni tra l'unità in esame e gli altri dispositivi viene controllata mediante una serie di operazioni elementari (alimentazione, invio segnale, ricezione segnale, etc.);
5. Settaggio della potenza prelevata/immessa: vengono impostati i parametri di test;
6. Verifica delle misure delle grandezze caratteristiche: vengono acquisite le letture delle grandezze di interesse mediante appositi energy meter confrontandole con i valori attesi e, laddove possibile, con i valori letti dagli strumenti in esame;
7. Valutazione dei tempi transitori: vengono stimati i tempi di risposta dello strumento.



**Figura A.1: Schema della verifica della connessione delle unità hardware di simulazione al simulatore Typhoon HIL**

La Figura A.2 rappresenta lo schema dell'architettura del software. Il core dell'architettura è rappresentato dal dizionario, in cui vengono registrati tutti i parametri topologici ed operativi degli elementi della rete in esame, nonché i parametri ed i risultati delle singole funzionalità, ed eventuali altri parametri globali della rete. Tutte le altre sezioni del software afferiscono direttamente al dizionario, prelevando le informazioni necessarie e apportando le modifiche impostate dall'utente (attraverso l'interfaccia grafica) o derivanti dalle valutazioni degli studi effettuati (funzionalità).

L'interfaccia grafica permette l'accesso alla rete da studiare (consentendone la creazione, o permettendo di aprire una rete benchmark o un modello di rete precedentemente salvato). Le caratteristiche della rete selezionata/creata vengono salvate sul dizionario. L'interfaccia grafica permette quindi di visualizzare le informazioni della rete in esame, e di apportare eventuali modifiche funzionali allo studio da effettuare.

Le funzionalità disponibili nel software (anch'esse accessibili tramite l'interfaccia grafica) prelevano i parametri della rete, nonché quelli strettamente correlati alla funzionalità stessa, dal dizionario, ed effettuano l'analisi prevista dal modello corrispondente. I risultati dello studio saranno a loro volta registrati nel dizionario, distinguendo tra i risultati relativi ai singoli componenti di rete e i risultati globali della rete. Tale approccio facilita l'interazione tra i diversi studi, offrendo la possibilità di implementare logiche multiple. Laddove necessario, le singole funzionalità possono accedere alle risorse offerte dal modulo OpenSource DSS per l'esecuzione dello studio del LoadFlow.

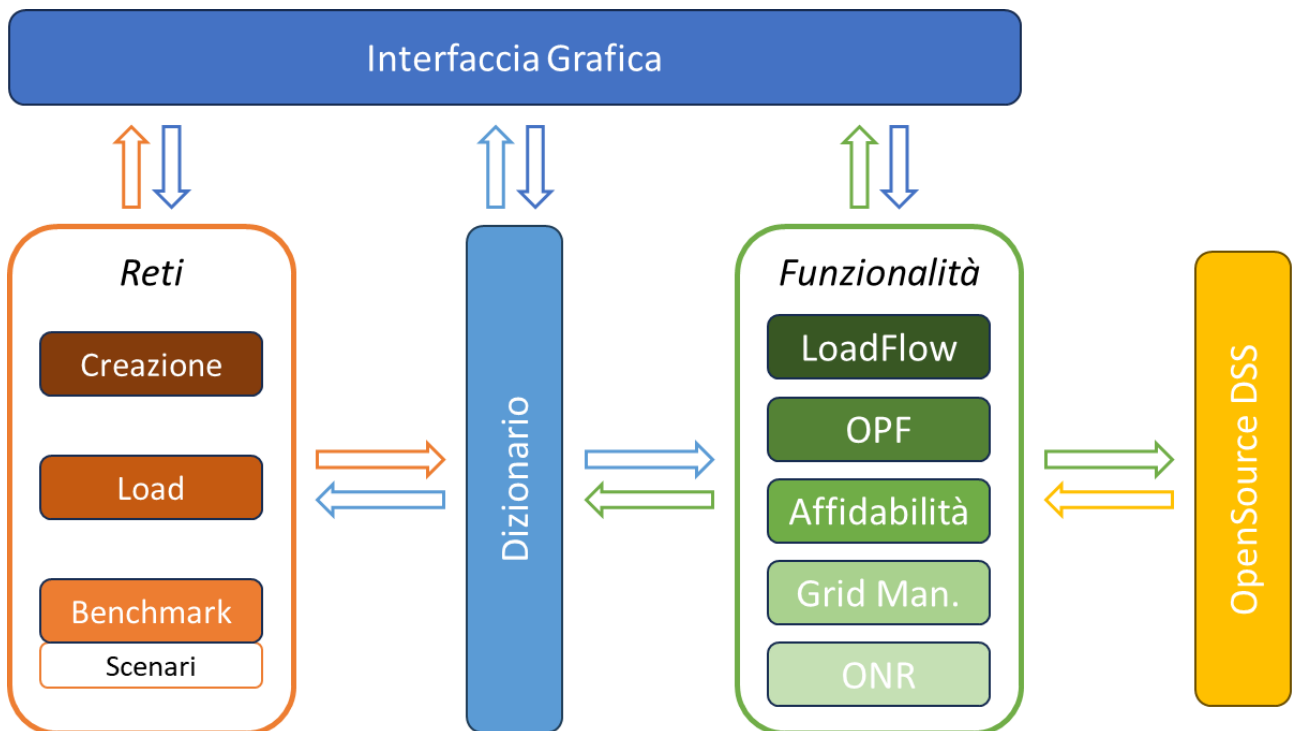
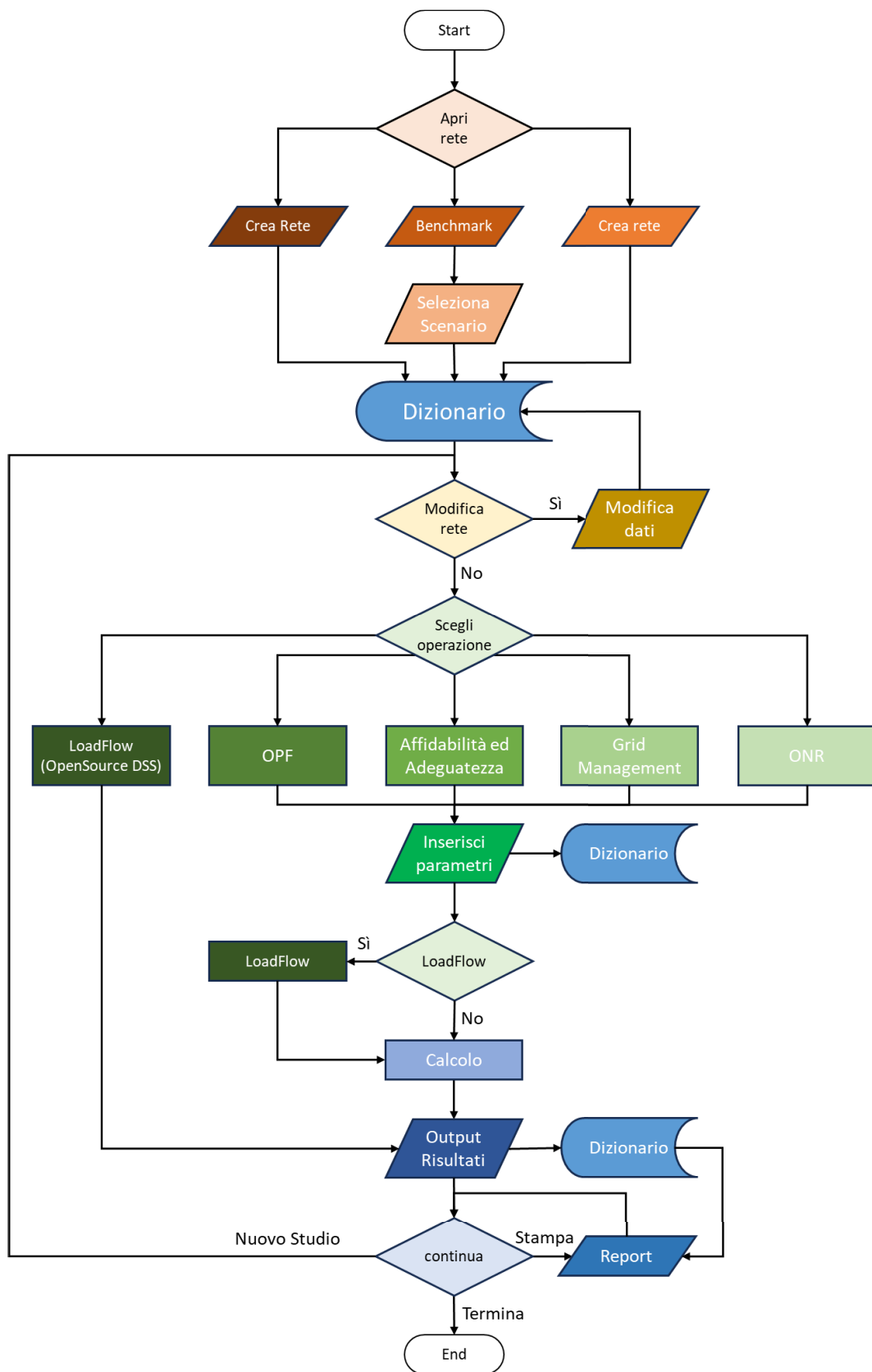


Figura A.2: Schema dell'architettura del software.



**Figura A.3: Albero di navigazione del software.**

## A.1 Tabelle

**Tabella A.1: Accoppiamento delle tecnologie con i dispositivi di laboratorio, caratteristiche di comunicazione e principali parametri di configurazione.**

<i>Tecnologia</i>	<i>Dispositivo di emulazione</i>	<i>Collegamento alla rete</i>	<i>Connessione analogica</i>	<i>Altre connessioni</i>	<i>Parametri principali</i>	<i>Grandezze in input</i>	<i>Grandezze in output</i>
<b>Generazione fotovoltaica</b>	<b>TerraSAS ETS/600/8(</b>	Rete AC	-10 / + 10 V	TCP-IP	<ul style="list-style-type: none"> <li>○ Potenza dei moduli;</li> <li>○ Arrangiamento dei moduli</li> </ul>	<ul style="list-style-type: none"> <li>○ Profilo di temperatura;</li> <li>○ Profilo di irraggiamento</li> </ul>	<ul style="list-style-type: none"> <li>○ Potenza</li> <li>○ Corrente</li> <li>○ Tensione</li> </ul>
<b>Generazione eolica</b>	<b>AMP Technologies SP300VAC5000W</b>	RETE AC	-10 / + 10 V	UDP SCPI	<ul style="list-style-type: none"> <li>○ Rating</li> <li>○ Protezioni</li> <li>○ Priorità di esercizio</li> </ul>	<ul style="list-style-type: none"> <li>○ Profilo di potenza</li> </ul>	<ul style="list-style-type: none"> <li>○ Potenza</li> <li>○ Corrente</li> <li>○ Tensione</li> </ul>
<b>Batterie</b>	<b>CINERGIA EL-15</b>	Rete DC Rete AC	-	Modbus TCP-IP	<ul style="list-style-type: none"> <li>○ Modalità di funzionamento</li> <li>○ Impedenza caratteristica</li> <li>○ Protezioni (I, V)</li> </ul>	<ul style="list-style-type: none"> <li>○ Erogazione / Assorbimento</li> <li>○ Potenza assorbita/generata</li> </ul>	<ul style="list-style-type: none"> <li>○ Potenza</li> <li>○ Tensione</li> <li>○ Corrente</li> </ul>
<b>Batterie</b>	<b>ITECH 6018C</b>	Rete DC	-10 / + 10 V	TCP-IP SCPI	<ul style="list-style-type: none"> <li>○ Modalità di funzionamento</li> <li>○ Protezioni (I, V)</li> </ul>	<ul style="list-style-type: none"> <li>○ Erogazione / Assorbimento</li> <li>○ Potenza assorbita/generata</li> </ul>	<ul style="list-style-type: none"> <li>○ Potenza</li> <li>○ Tensione</li> <li>○ Corrente</li> </ul>
<b>Elettrolizzatore</b>	<b>ITECH 6018C</b>	Rete DC	-10 / + 10 V	TCP-IP SCPI	<ul style="list-style-type: none"> <li>○ Protezioni (I, V)</li> </ul>	<ul style="list-style-type: none"> <li>○ Potenza assorbita</li> </ul>	<ul style="list-style-type: none"> <li>○ Potenza</li> <li>○ Tensione</li> <li>○ Corrente</li> </ul>
<b>Fuel Cell</b>	<b>ITECH 6018C</b>	Rete DC	-10 / + 10 V	TCP-IP SCPI	<ul style="list-style-type: none"> <li>○ Protezioni (I, V)</li> </ul>	<ul style="list-style-type: none"> <li>○ Potenza generata</li> </ul>	<ul style="list-style-type: none"> <li>○ Potenza</li> <li>○ Tensione</li> <li>○ Corrente</li> </ul>
<b>Carichi AC</b>	<b>ITECH 8615</b>	RETE AC	-10 / + 10 V	TCP-IP SCPI	<ul style="list-style-type: none"> <li>○ Protezioni (I, V)</li> </ul>	<ul style="list-style-type: none"> <li>○ Potenza assorbita</li> </ul>	<ul style="list-style-type: none"> <li>○ Potenza</li> <li>○ Tensione</li> <li>○ Corrente</li> </ul>
<b>Carichi DC</b>	<b>ITECH 6018C</b>	Rete DC	-10 / + 10 V	TCP-IP SCPI	<ul style="list-style-type: none"> <li>○ Protezioni (I, V)</li> </ul>	<ul style="list-style-type: none"> <li>○ Potenza assorbita</li> </ul>	<ul style="list-style-type: none"> <li>○ Potenza</li> <li>○ Tensione</li> <li>○ Corrente</li> </ul>

**Tabella A.2: Comparazione delle caratteristiche più rilevanti tra i software**

<i>Caratteristica</i>	<i>PyPSA</i>	<i>PandaPower</i>	<i>OpenDSS</i>
<b>Gestione elementi DC</b>	NO	SÌ	SÌ
<b>Gestione degli storage</b>	NO	NO	SÌ
<b>Facilità di programmazione<sup>1</sup></b>	3°	2°	1°
<b>Precisione dei risultati<sup>1</sup></b>	1°	3°	2°
<b>Tempo di calcolo<sup>1</sup></b>	3°	2°	1°
<b>Power Flow sbilanciati</b>	NO	SÌ	SÌ
<b>Integrazione con Typhoon-HIL</b>	NO	NO	SÌ

<sup>1</sup> L. Giorgi, A. Obushevs and P. Korba, "Open Source Module for the Investigation of the Impact of Electric Vehicles in a Low Voltage Grid," 2021 IEEE 62nd International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTU CON), Riga, Latvia, 2021, pp. 1-10, doi: 10.1109/RTU CON53541.2021.9711593.

**Tabella A.3: Comparazione delle caratteristiche più rilevanti tra i linguaggi di programmazione**

<i>Parametri</i>	<i>Python</i>	<i>Julia</i>	<i>Java</i>	<i>C</i>
<b>Curva di apprendimento</b>	Beginner friendly	Ripida per le funzionalità più avanzate	Intermedia	Ripida (necessità di gestione aspetti di basso livello)
<b>Facilità d'uso</b>	Sintassi semplice e leggibile	Sintassi semplice e leggibile	Sintassi verbosa, esperienza richiesta	Sintassi poco naturale, esperienza richiesta
<b>Supporto</b>	Ampia community; grande disponibilità di documentazione online	Community limitata; poche risorse disponibili	Community molto ampia, estesa documentazione disponibile online	Community ampia, discreta disponibilità di risorse online
<b>Librerie per calcolo scientifico</b>	Molte librerie disponibili (NumPy, SciPy, etc.);	Limitata disponibilità di librerie dedicate;	Non progettato per il calcolo scientifico; Limitata disponibilità di librerie dedicate;	Disponibilità alta di librerie dedicate;
<b>Velocità nel calcolo scientifico</b>	Non trascurabile impegno della CPU	Elevata velocità di esecuzione	Velocità di esecuzione potenzialmente alta	Velocità di esecuzione molto elevata ed efficiente
<b>Integrazione con strumenti di laboratorio</b>	Facilitata dalla vasta disponibilità di librerie; Real-Time processing limitato dalla natura del linguaggio; Limitazioni nei processi multi-threading	Limitata disponibilità di librerie dedicate; Adatto al real-time	Limitata disponibilità di librerie dedicate Facilità di realizzare applicazioni multi-piattaforma;	Poco adatto alla prototipazione rapida; Necessaria una conoscenza avanzata della macchina sottostante; Processi real-time facilitati dalla interazione a basso livello con la strumentazione di laboratorio;
<b>Integrazione con applicativi DSS</b>	<ul style="list-style-type: none"> <li>• PowerFactory</li> <li>• Neplan</li> <li>• PyDSS</li> <li>• PandaPower</li> <li>• OpenDSS</li> </ul>	<ul style="list-style-type: none"> <li>• PowerFactory</li> <li>• PyDSS</li> <li>• PandaPower</li> <li>• OpenDSS</li> </ul>	<ul style="list-style-type: none"> <li>• OpenDSS</li> </ul>	-
<b>Realizzazione di GUI</b>	Disponibilità di numerose librerie (Tkinter, PyQt, Kivy); Possibilità di cross-piattaforma	Non dispone di librerie dedicate alla realizzazione delle GUI; Le GUI devono essere realizzate mediante altri linguaggi	Ampia disponibilità di librerie; Possibilità di cross-piattaforma	Realizzazione di GUI customizzate; Possibilità di cross-piattaforma