



# CerCollettiva

Software opensource per la gestione delle comunità energetiche

07/06/2024

**Andrea Bernardi**

Venetiae Progetti STP S.r.l.

Via delle Macchine 9/F

30038, Spinea (Ve)

Distribuito secondo licenza MIT

# Indice della Documentazione per lo Sviluppo del Software

<b>1. Inquadramento del Progetto</b>	<b>3</b>
1.1. Obiettivo Principale	3
1.2. Contesto Normativo	4
1.3. Beneficiari	5
1.4. Funzionalità Chiave	7
1.5. Tecnologie e Infrastruttura	9
1.6. Open Source e Licenza	11
<b>2. Analisi dei Requisiti</b>	<b>13</b>
2.1. Requisiti Funzionali	13
2.1.1. Creazione e Adesione	13
2.1.2. Gestione Impianti e Misure	14
2.1.3. Calcolo Incentivi e Contributi	15
2.1.4. Amministrazione e Reporting	16
2.1.5. Requisiti Aggiuntivi (Parte III del Decreto CACER)	16
2.2. Requisiti Tecnici	18
<b>3. Progettazione dell'Architettura</b>	<b>22</b>
3.1. Architettura Generale	22
3.2. Modello dei Dati	24
3.4. API (Application Programming Interface)	28
<b>4. Sviluppo del Prototipo su Raspberry Pi</b>	<b>30</b>
4.1. Ambiente di Sviluppo	30
4.2. Linguaggio di Programmazione e Framework	32
Struttura del Database	34
Migrazioni	35
Seed	36
Ottimizzazione	36
4.5. Funzionalità del Prototipo	38
<b>5. Migrazione al Cloud e Deployment</b>	<b>40</b>
5.1. Scelta del Provider Cloud	40
5.2. Configurazione dell'Infrastruttura	40
5.3. Deployment del Software	40
5.4. Test e Monitoraggio	40
<b>6. Versione Open Source su GitHub</b>	<b>40</b>
6.1. Creazione del Repository	40
6.2. Gestione delle Versioni e dei Contributi	40
6.3. Documentazione	40
6.4. Community	40

<b>7. Manutenzione e Evoluzione</b>	<b>40</b>
7.1. Aggiornamenti Normativi	40
7.2. Nuove Funzionalità	40
7.3. Supporto agli Utenti	40
7.4. Monitoraggio e Risoluzione Problemi	40
<b>Appendice</b>	<b>40</b>
• A. Glossario dei Termini	40
• B. Riferimenti Normativi	40
• C. Diagrammi di Flusso (se necessario)	40
• D. Wireframe dell'Interfaccia Utente (se necessario)	40


# 1. Inquadramento del Progetto

## 1.1. Obiettivo Principale

L'obiettivo principale del progetto è sviluppare un software open-source per facilitare la creazione e la gestione delle Comunità Energetiche Rinnovabili (CER) e delle Comunità Energetiche dei Cittadini (CEC) in Italia. Il software si propone di:

1. **Semplificare l'adesione:** Rendere più semplice e accessibile il processo di creazione e adesione alle CER e CEC, guidando gli utenti attraverso i requisiti normativi e le procedure amministrative.
2. **Ottimizzare la gestione:** Fornire strumenti per la gestione efficiente degli impianti di produzione di energia rinnovabile, l'acquisizione e l'analisi dei dati di misura, il calcolo degli incentivi e dei contributi, e la rendicontazione delle attività.
3. **Promuovere la partecipazione:** Incoraggiare la partecipazione attiva dei cittadini, delle imprese e degli enti locali alla transizione energetica, facilitando la condivisione dell'energia prodotta da fonti rinnovabili e la creazione di comunità energetiche locali.
4. **Garantire la conformità normativa:** Assicurare che tutte le attività delle CER e CEC siano conformi alle normative vigenti, in particolare al Decreto CACER e al TIAD, semplificando la verifica dei requisiti e la gestione degli adempimenti amministrativi.
5. **Massimizzare i benefici:** Aiutare le CER e CEC a massimizzare i benefici economici derivanti dagli incentivi e dai contributi previsti dalla normativa, attraverso un calcolo preciso e trasparente dell'energia condivisa e dei corrispettivi spettanti.
6. **Contribuire alla sostenibilità:** Promuovere la produzione e l'autoconsumo di energia da fonti rinnovabili, contribuendo alla riduzione delle emissioni di gas serra e alla transizione verso un sistema energetico più sostenibile.

Il software sarà progettato per essere **modulare, scalabile e flessibile**, in modo da potersi adattare alle diverse esigenze delle comunità energetiche e alle evoluzioni del quadro normativo. La sua natura open-source favorirà la collaborazione e il contributo



della community, garantendo un continuo miglioramento e adattamento alle esigenze del settore.


## 1.2. Contesto Normativo

Il quadro normativo di riferimento per lo sviluppo del software è definito da due documenti principali:

1. **Decreto Ministeriale 7 dicembre 2023, n. 414 (Decreto CACER):** Questo decreto disciplina le modalità di incentivazione dell'energia elettrica prodotta da impianti a fonti rinnovabili inseriti in configurazioni di autoconsumo. Stabilisce inoltre i criteri e le modalità per la concessione di contributi in conto capitale previsti dal Piano Nazionale di Ripresa e Resilienza (PNRR) per lo sviluppo delle comunità energetiche e dell'autoconsumo collettivo nei comuni con meno di 5.000 abitanti.
2. **Deliberazione ARERA 727/2022/R/eel (TIAD):** Questa deliberazione definisce la regolazione economica e le modalità tecniche per l'autoconsumo diffuso. Stabilisce le tipologie di configurazioni ammesse (autoconsumatore individuale, gruppo di autoconsumatori, comunità energetiche, ecc.), le modalità di interazione con la rete elettrica, i requisiti per l'accesso agli incentivi e le modalità di calcolo dei contributi economici.

Il software dovrà tenere conto di tutti gli aspetti tecnici e legali previsti da questi due documenti, garantendo la conformità delle CER e CEC alle normative vigenti e la corretta applicazione dei meccanismi di incentivazione. In particolare, il software dovrà:

- **Verificare i requisiti di ammissibilità** delle CER e CEC, come definiti nel Decreto CACER e nel TIAD (ad esempio, tipologia di soggetti ammessi, requisiti degli impianti, ecc.).
- **Gestire le domande di accesso agli incentivi e ai contributi**, assicurando che la documentazione presentata sia completa e conforme ai requisiti.
- **Calcolare l'energia condivisa** in base alle misure fornite dai gestori di rete e alle regole del TIAD.

- 
- **Determinare la tariffa incentivante** per le CER e il contributo di valorizzazione per l'energia autoconsumata, applicando le formule e i criteri definiti nel Decreto CACER e nel TIAD.
  - **Gestire l'erogazione degli incentivi e dei contributi**, garantendo la corretta applicazione delle tempistiche e delle modalità di pagamento.

Inoltre, il software dovrà essere aggiornato in caso di modifiche normative, garantendo sempre la conformità alle disposizioni vigenti.


Il Decreto Ministeriale del 18 marzo 2024 introduce dei corrispettivi a carico dei soggetti che accedono ai benefici previsti dal Decreto CACER. Il software dovrà quindi tener conto anche di questi corrispettivi, gestendo la fatturazione e il pagamento degli stessi.

### 1.3. Beneficiari

Il software è progettato principalmente per i **referenti** delle Comunità Energetiche Rinnovabili (CER) e delle Comunità Energetiche dei Cittadini (CEC). Queste figure chiave sono responsabili della gestione tecnica e amministrativa delle comunità e necessitano di strumenti adeguati per svolgere le loro funzioni in modo efficiente e conforme alla normativa.

Il software offrirà ai referenti le seguenti funzionalità:

- **Monitoraggio dei flussi energetici:** Permette di visualizzare in tempo reale la produzione, il consumo e la condivisione dell'energia all'interno della comunità, consentendo di valutare l'efficienza del sistema e identificare eventuali criticità.
- **Identificazione anomalie:** Il software analizzerà i dati di produzione e consumo per individuare eventuali anomalie o comportamenti sospetti, come consumi eccessivi o produzione insufficiente da parte di alcuni membri.
- **Interventi tempestivi:** In caso di anomalie, il software invierà notifiche e avvisi al referente, consentendo di intervenire tempestivamente per risolvere i problemi e garantire il corretto funzionamento della CER/CEC.

- 
- **Gestione amministrativa:** Il software faciliterà la gestione degli aspetti amministrativi delle CER/CEC, come la tenuta dell'anagrafica dei membri, la gestione dei contratti, la fatturazione e la rendicontazione delle attività.
  - **Accesso agli incentivi:** Il software supporterà i referenti nella gestione delle domande di accesso agli incentivi e ai contributi previsti dal Decreto CACER e dal TIAD, semplificando le procedure e garantendo la conformità ai requisiti.

In una fase successiva, il software potrebbe essere esteso per includere funzionalità dedicate anche agli **utenti** delle CER e CEC, come la possibilità di visualizzare i propri dati di consumo e produzione, l'andamento della condivisione dell'energia e i benefici economici ottenuti.

### **Beneficiari Indiretti**

Oltre ai referenti, il software porterà benefici anche ad altri attori coinvolti nel sistema dell'autoconsumo diffuso:

- **Membri delle CER/CEC:** Avranno a disposizione informazioni chiare e trasparenti sul funzionamento della comunità energetica, sui propri consumi e sulla propria quota di energia condivisa.
- **Gestori di Rete:** Potranno beneficiare di una maggiore efficienza nella gestione dei dati di misura e nella comunicazione con le comunità energetiche.
- **GSE:** Potrà contare su un sistema più efficiente per la gestione delle domande di accesso agli incentivi e per il monitoraggio delle attività delle CER/CEC.

Il software, grazie alla sua natura open-source, potrà essere utilizzato e adattato da una vasta gamma di soggetti, contribuendo così a diffondere la cultura dell'autoconsumo e a promuovere la transizione energetica in Italia.

## 1.4. Funzionalità Chiave

Il software dovrà fornire le seguenti funzionalità chiave per supportare l'intero ciclo di vita delle CER e CEC, dalla creazione alla gestione operativa, fino alla rendicontazione e all'accesso agli incentivi:

- **Creazione e Adesione:**

- **Registrazione e Profilazione Utenti:** Creazione di profili utente per i membri delle CER/CEC (cittadini, imprese, enti locali), con raccolta di dati anagrafici, informazioni di contatto, ruolo nella comunità (produttore, consumatore, prosumer) e dati fiscali.
- **Verifica Automatica dei Requisiti:** Implementazione di controlli automatici per verificare l'ammissibilità dei membri e degli impianti in base ai requisiti del Decreto CACER e del TIAD.
- **Costituzione Guidata delle CER/CEC:** Una procedura guidata per la creazione delle comunità energetiche, che supporti la redazione dello statuto o dell'atto costitutivo in conformità alla normativa.
- **Gestione Documenti:** Raccolta e archiviazione di documenti importanti come statuti, contratti di mandato, liberatorie e altri documenti amministrativi.
- **Firma Digitale:** Integrazione con sistemi di firma digitale per la sottoscrizione dei documenti.

- **Gestione Impianti e Misure:**

- **Catalogo Impianti:** Creazione di un database degli impianti di produzione di energia rinnovabile, con informazioni dettagliate su tipologia, potenza, ubicazione, data di entrata in esercizio, componenti installati, ecc.
- **Verifica Requisiti Impianti:** Controlli automatici per verificare la conformità degli impianti ai requisiti tecnici e normativi (potenza massima, entrata in esercizio, conformità al principio DNSH, ecc.).
- **Acquisizione Dati di Misura:** Ricezione dei dati di misura dell'energia prodotta, immessa in rete e autoconsumata tramite il broker MQTT.



- **Profilazione dei Dati:** Elaborazione dei dati di misura per la creazione di profili di produzione e consumo, utili per l'analisi e la previsione dei flussi energetici.
- **Calcolo Energia Condivisa:** Calcolo dell'energia elettrica condivisa (autoconsumata e immessa in rete) in base alle regole del TIAD.
- **Visualizzazione Dati:** Rappresentazione grafica dei dati di produzione, consumo e condivisione dell'energia, con dashboard intuitive e personalizzabili.
- **Allarmi e Notifiche:** Invio di notifiche automatiche al referente in caso di anomalie nei flussi energetici o di superamento delle soglie previste dalla normativa.
- **Calcolo Incentivi e Contributi:**
  - **Determinazione Tariffa Incentivante:** Calcolo automatico della tariffa incentivante per le CER in base ai dati di produzione, ai prezzi di mercato e ai parametri del Decreto CACER.
  - **Calcolo Contributo di Valorizzazione:** Calcolo automatico del contributo per la valorizzazione dell'energia autoconsumata per CER e CEC, tenendo conto delle componenti tariffarie e del livello di tensione.
  - **Gestione Pagamenti:** Gestione dei pagamenti degli incentivi e dei contributi, con generazione di fatture e integrazione con sistemi di pagamento elettronico.
  - **Rendicontazione Economica:** Produzione di report dettagliati sui flussi economici, con indicazione degli incentivi e dei contributi ricevuti, delle spese sostenute e del bilancio complessivo della CER/CEC.
- **Amministrazione e Reporting:**
  - **Gestione Utenti e Ruoli:** Definizione di ruoli e permessi per i diversi utenti (referente, amministratore, membri), con livelli di accesso differenziati alle informazioni e alle funzionalità del software.
  - **Comunicazioni:** Invio di comunicazioni personalizzate ai membri delle CER/CEC (notifiche, avvisi, richieste di dati, ecc.).

- **Reporting:** Generazione di report periodici sulle attività della CER/CEC, sull'andamento degli incentivi e dei contributi, e sulla conformità ai requisiti normativi.
- **Archiviazione Documentale:** Conservazione digitale di tutti i documenti relativi alla CER/CEC, con possibilità di ricerca e consultazione.
- **Requisiti Aggiuntivi (Parte III del Decreto CACER):**
  - **Gestione Domande di Contributo:** Supporto alla compilazione e all'invio delle domande per i contributi in conto capitale previsti dal PNRR.
  - **Verifica Requisiti PNRR:** Controlli automatici per verificare l'ammissibilità ai contributi PNRR (localizzazione dell'impianto, data di avvio lavori, ecc.).
  - **Calcolo Contributo PNRR:** Calcolo dell'ammontare massimo del contributo in conto capitale spettante.
  - **Gestione Fasi di Erogazione:** Gestione delle richieste di anticipazione, quota intermedia e saldo del contributo PNRR, con verifica della documentazione e degli stati di avanzamento lavori.

Il software sarà sviluppato con un approccio modulare, consentendo di aggiungere o modificare funzionalità in base alle esigenze specifiche delle CER/CEC e alle evoluzioni normative. La sua architettura sarà progettata per garantire scalabilità, affidabilità e sicurezza, sia nell'ambiente iniziale su Raspberry Pi che nella successiva migrazione al cloud.

## 1.5. Tecnologie e Infrastruttura

L'ambiente di sviluppo iniziale sarà basato su un Raspberry Pi, un microcomputer economico e versatile, ideale per la prototipazione e lo sviluppo iniziale del software. Tuttavia, l'architettura del software sarà progettata per essere scalabile e facilmente migrabile su un'infrastruttura cloud, per garantire prestazioni e affidabilità elevate in un ambiente di produzione.



## Scelta delle Tecnologie

Le tecnologie scelte per lo sviluppo del software sono state selezionate per la loro affidabilità, scalabilità e ampia adozione nella community open source:

- **Linguaggio di Programmazione:** PHP, un linguaggio di scripting lato server ampiamente utilizzato per lo sviluppo web e particolarmente adatto per applicazioni che richiedono l'interazione con database e la gestione di dati.
- **Framework PHP:** Laravel, un framework PHP moderno e potente, che offre una struttura solida e strumenti avanzati per lo sviluppo rapido di applicazioni web.
- **Database:** MySQL o PostgreSQL, due sistemi di gestione di database relazionali open source molto diffusi, che offrono prestazioni elevate e affidabilità nella gestione dei dati.
- **Broker MQTT:** Mosquitto, un broker MQTT open source leggero e performante, ideale per la gestione della comunicazione in tempo reale tra i dispositivi IoT (Internet of Things) e il software.

## Infrastruttura Cloud

Per il deployment del software in un ambiente di produzione, si prevede la migrazione dell'intero ambiente di sviluppo, configurato sul Raspberry Pi, su un'infrastruttura cloud. Questa scelta garantirà:

- **Scalabilità:** La possibilità di adattare le risorse (CPU, memoria, storage) in base al carico di lavoro e al numero di utenti. 10
- **Affidabilità:** Alta disponibilità del software, grazie alla ridondanza dei server e ai sistemi di backup.
- **Sicurezza:** Protezione dei dati e delle informazioni sensibili attraverso meccanismi di sicurezza avanzati.
- **Manutenzione:** Semplificazione delle attività di manutenzione e aggiornamento del software.

## 1.6. Open Source e Licenza

Il software CerCollettiva sarà rilasciato come progetto open source, con il codice sorgente reso disponibile su GitHub. La licenza scelta per il progetto sarà la **MIT License**, una licenza permissiva che garantisce ampia libertà di utilizzo, modifica e distribuzione del software, sia per scopi personali che commerciali.

### Vantaggi dell'Open Source


La scelta di rilasciare il software come open source è motivata da diversi vantaggi:

- **Collaborazione:** L'open source incoraggia la collaborazione e il contributo della community di sviluppatori, che possono segnalare bug, proporre miglioramenti e aggiungere nuove funzionalità.
- **Trasparenza:** Il codice sorgente aperto garantisce la trasparenza del software, consentendo a chiunque di verificarne il funzionamento e l'affidabilità.
- **Flessibilità:** Gli utenti possono adattare il software alle proprie esigenze specifiche, modificando il codice sorgente o creando nuove funzionalità.
- **Innovazione:** L'open source favorisce l'innovazione, permettendo a sviluppatori e ricercatori di utilizzare il software come base per nuovi progetti e sperimentazioni.
- **Sostenibilità:** Il software open source è più sostenibile nel lungo periodo, grazie al contributo della community che ne garantisce la manutenzione e l'aggiornamento.

### MIT License

La MIT License è stata scelta per la sua semplicità e permissività. Essa consente a chiunque di utilizzare, copiare, modificare, unire, pubblicare, distribuire, concedere in sublicenza e/o vendere copie del software, a condizione che venga inclusa una copia della licenza in tutte le copie o parti sostanziali del software.

La licenza non impone restrizioni sull'uso commerciale del software e non richiede la condivisione delle modifiche apportate al codice sorgente. Questo rende la MIT License



particolarmente adatta per progetti che vogliono favorire la massima diffusione e adozione del software.

## GitHub

GitHub sarà la piattaforma utilizzata per ospitare il codice sorgente del progetto e gestire la collaborazione con la community. Su GitHub sarà possibile:

- **Creare il repository del progetto:** Un repository pubblico dove sarà disponibile il codice sorgente, la documentazione e le informazioni sul progetto.
- **Gestire le versioni e i contributi:** Utilizzare il sistema di controllo di versione Git per tracciare le modifiche al codice e gestire i contributi della community.
- **Collaborare con la community:** Segnalare bug, proporre miglioramenti e discutere nuove funzionalità attraverso il sistema di issue e pull request di GitHub.
- **Fornire documentazione:** Creare una documentazione completa e aggiornata per guidare gli utenti nell'installazione, configurazione e utilizzo del software.

Il progetto CerCollettiva, grazie alla sua natura open source e alla licenza MIT, si propone di creare un ecosistema collaborativo e innovativo per lo sviluppo e la diffusione di soluzioni software per le comunità energetiche in Italia.

## 2. Analisi dei Requisiti

### 2.1. Requisiti Funzionali


In questa sezione, vengono analizzati i requisiti funzionali, tecnici e legali che il software CerCollettiva dovrà soddisfare per garantire la corretta gestione delle Comunità Energetiche Rinnovabili (CER) e delle Comunità Energetiche dei Cittadini (CEC) in conformità con le normative vigenti (Decreto CACER e TIAD). L'obiettivo è definire le funzionalità chiave che il software dovrà implementare per supportare l'intero ciclo di vita delle comunità energetiche, dalla creazione e adesione alla gestione operativa, fino alla rendicontazione, all'accesso agli incentivi e al calcolo dei contributi.

Inoltre, verranno considerati i requisiti aggiuntivi previsti dalla Parte III del Decreto CACER, che disciplina l'accesso ai contributi in conto capitale del Piano Nazionale di Ripresa e Resilienza (PNRR) per le CER e i gruppi di autoconsumatori nei comuni con meno di 5.000 abitanti. L'analisi dei requisiti sarà fondamentale per la progettazione dell'architettura del software e per lo sviluppo del prototipo su Raspberry Pi, che verrà successivamente migrato su un'infrastruttura cloud per garantire scalabilità e affidabilità.

La definizione precisa dei requisiti consentirà di sviluppare un software completo, efficiente e conforme alle normative, in grado di soddisfare le esigenze dei referenti delle CER e CEC e di contribuire alla diffusione dell'autoconsumo di energia rinnovabile in Italia.

#### 2.1.1. Creazione e Adesione

- **Registrazione e Profilazione Utenti:** Il software dovrà consentire la registrazione e la creazione di profili utente per i membri delle CER/CEC (cittadini, imprese, enti locali). Sarà necessario raccogliere dati anagrafici, informazioni di contatto (inclusi PEC e SPID), ruolo nella comunità (produttore, consumatore) e dati fiscali (codice fiscale/partita IVA).
- **Verifica Automatica dei Requisiti:** Il software dovrà implementare controlli automatici per verificare l'ammissibilità dei membri e degli impianti in base ai requisiti del Decreto CACER (come ad esempio il rispetto del




principio DNSH) e del TIAD (come la verifica della potenza degli impianti e l'ubicazione all'interno della stessa cabina primaria). Questi controlli dovranno essere effettuati sia in fase di registrazione che durante la gestione della CER/CEC.

- **Costituzione Guidata delle CER/CEC:** Il software dovrà guidare gli utenti attraverso il processo di costituzione delle CER/CEC, supportando la redazione dello statuto o dell'atto costitutivo in conformità alla normativa. Dovrà essere fornita una lista di controllo per garantire che tutti i requisiti siano soddisfatti.
- **Gestione Documenti:** Il software dovrà consentire la raccolta, l'archiviazione e la gestione di documenti importanti come statuti, contratti di mandato, liberatorie e altri documenti amministrativi.
- **Firma Digitale:** Il software dovrà integrarsi con sistemi di firma digitale per consentire la sottoscrizione dei documenti in modo sicuro e legalmente valido.

#### 2.1.2. Gestione Impianti e Misure

- **Catalogo Impianti:** Il software dovrà includere un catalogo degli impianti di produzione di energia rinnovabile, con informazioni dettagliate su tipologia (fotovoltaico, eolico, idroelettrico, biogas, biomasse), potenza, ubicazione, data di entrata in esercizio, componenti installati (inclusi sistemi di accumulo), dati catastali e codice POD.
- **Verifica Requisiti Impianti:** Il software dovrà verificare automaticamente la conformità degli impianti ai requisiti tecnici e normativi, come la potenza massima (1 MW), la data di entrata in esercizio (successiva al 16 dicembre 2021), la conformità al principio DNSH, i criteri per gli impianti a biogas e biomasse e l'utilizzo di componenti nuovi o rigenerati.
- **Acquisizione Dati di Misura:** Il software dovrà acquisire i dati di misura dell'energia prodotta, immessa in rete e autoconsumata, tramite API messe a disposizione dai gestori di rete o direttamente dai dispositivi di misura (se supportati). I dati dovranno essere acquisiti con la frequenza



prevista dal Testo Integrato Misura Elettrica (TIME) e memorizzati in modo sicuro.

- **Profilazione dei Dati:** Il software dovrà elaborare i dati di misura per creare profili di produzione e consumo su base oraria e/o giornaliera. Questi profili saranno utili per l'analisi dell'efficienza energetica, la previsione dei flussi energetici e il calcolo degli incentivi.
- **Calcolo Energia Condivisa:** Il software dovrà calcolare automaticamente l'energia elettrica condivisa (autoconsumata e immessa in rete) in base alle regole del TIAD, tenendo conto delle diverse tipologie di configurazione (autoconsumatore individuale, gruppo di autoconsumatori, CER).

### 2.1.3. Calcolo Incentivi e Contributi

- **Determinazione Tariffa Incentivante:** Per le CER, il software dovrà calcolare automaticamente la tariffa incentivante in base ai dati di produzione, ai prezzi di mercato e ai parametri del Decreto CACER. Il calcolo dovrà essere effettuato su base oraria e mensile, tenendo conto anche delle eventuali decurtazioni previste in caso di cumulo con altri incentivi.
- **Calcolo Contributo di Valorizzazione:** Per le CER e le CEC, il software dovrà calcolare automaticamente il contributo per la valorizzazione dell'energia autoconsumata, tenendo conto delle componenti tariffarie (trasmissione, distribuzione, perdite di rete) e del livello di tensione.
- **Gestione Pagamenti:** Il software dovrà gestire i pagamenti degli incentivi e dei contributi, generando fatture elettroniche e integrandosi con sistemi di pagamento elettronico (ad esempio, PagoPA).
- **Rendicontazione Economica:** Il software dovrà generare report dettagliati sui flussi economici delle CER/CEC, con indicazione degli incentivi e dei contributi ricevuti, delle spese sostenute e del bilancio complessivo. Questi report dovranno essere personalizzabili in base alle esigenze dell'utente e esportabili in diversi formati.




#### 2.1.4. Amministrazione e Reporting

- **Gestione Utenti e Ruoli:** Il software dovrà consentire la gestione degli utenti e dei loro ruoli all'interno della CER/CEC. Dovranno essere definiti diversi livelli di accesso (referente, amministratore, membri) con permessi differenziati per la visualizzazione e la modifica dei dati.
- **Comunicazioni:** Il software dovrà permettere l'invio di comunicazioni personalizzate ai membri delle CER/CEC (notifiche, avvisi, richieste di dati, ecc.). Le comunicazioni potranno essere inviate tramite email, SMS o notifiche push, a seconda delle preferenze dell'utente.
- **Reporting:** Il software dovrà generare report periodici (mensili, trimestrali, annuali) sulle attività della CER/CEC, sull'andamento degli incentivi e dei contributi, e sulla conformità ai requisiti normativi. I report dovranno essere chiari, dettagliati e facilmente comprensibili.
- **Archiviazione Documentale:** Il software dovrà fornire un sistema di archiviazione documentale per conservare digitalmente tutti i documenti relativi alla CER/CEC (statuti, contratti, fatture, ecc.). L'archivio dovrà essere organizzato in modo efficiente e consentire la ricerca e la consultazione dei documenti.


#### 2.1.5. Requisiti Aggiuntivi (Parte III del Decreto CACER)

- **Gestione Domande di Contributo:** Il software dovrà supportare la compilazione e l'invio delle domande per i contributi in conto capitale previsti dal PNRR per le CER e i gruppi di autoconsumatori nei comuni con meno di 5.000 abitanti. Dovrà essere fornita una guida alla compilazione e un sistema di verifica della completezza e correttezza dei dati inseriti.
- **Verifica Requisiti PNRR:** Il software dovrà implementare controlli automatici per verificare l'ammissibilità delle CER e dei gruppi di autoconsumatori ai contributi PNRR, come la localizzazione dell'impianto in un comune con meno di 5.000 abitanti, la data di avvio lavori e gli altri requisiti previsti dal Decreto CACER.

- 
- **Calcolo Contributo PNRR:** Il software dovrà calcolare automaticamente l'ammontare massimo del contributo in conto capitale spettante in base ai costi ammissibili e ai massimali previsti dal Decreto CACER.
  - **Gestione Fasi di Erogazione:** Il software dovrà gestire le diverse fasi di erogazione del contributo PNRR (anticipazione, quota intermedia, saldo), verificando la documentazione presentata (fatture, bonifici, ecc.) e gli stati di avanzamento lavori. Dovrà inoltre supportare la comunicazione con il GSE e il Ministero dell'Ambiente e della Sicurezza Energetica per la richiesta e l'ottenimento dei contributi.


#### 2.1.6. Ripartizione Incentivi

- **Definizione Criteri di Ripartizione:** Il software dovrà consentire alla CER di definire i propri criteri di ripartizione degli incentivi, tenendo conto delle quote di energia autoconsumata e immessa in rete da ciascun membro, nonché delle quote destinate a coprire le spese della comunità energetica. Il software dovrà prevedere la possibilità di scegliere tra diversi modelli di ripartizione predefiniti (ad esempio, ripartizione proporzionale all'energia autoconsumata, ripartizione in parti uguali, ecc.) o di definire criteri personalizzati.
- **Personalizzazione delle Quote:** Il software dovrà permettere di personalizzare le quote di ripartizione per ciascun membro, in base a specifici accordi o regolamenti interni alla CER. Ad esempio, si potrà prevedere l'assegnazione di quote maggiori ai membri che hanno contribuito maggiormente alla realizzazione dell'impianto o che hanno particolari esigenze di consumo.
- **Calcolo Automatico:** Il software dovrà calcolare automaticamente l'ammontare degli incentivi spettanti a ciascun membro in base ai criteri definiti e ai dati di misura dell'energia. Il calcolo dovrà essere effettuato periodicamente (ad esempio, mensilmente) e dovrà tenere conto dell'eventuale energia immessa in rete e delle tariffe incentivanti previste dalla normativa.

- 
- **Trasparenza e Rendicontazione:** Il software dovrà garantire la trasparenza del processo di ripartizione, fornendo ai membri della CER un resoconto dettagliato dei calcoli e delle quote assegnate. Il resoconto dovrà essere facilmente accessibile e comprensibile, anche per utenti non tecnici. Dovrà inoltre essere possibile esportare i dati in diversi formati (ad esempio, PDF, CSV) per ulteriori analisi o per la condivisione con i membri della CER.
  - **Gestione delle Controversie:** Il software dovrà prevedere meccanismi per la gestione delle eventuali controversie tra i membri della CER in merito alla ripartizione degli incentivi. Ad esempio, si potrà prevedere la possibilità di segnalare eventuali errori o incongruenze nei calcoli o di richiedere una revisione dei criteri di ripartizione. Il software dovrà inoltre fornire strumenti per la mediazione e la risoluzione delle controversie, come la possibilità di coinvolgere un soggetto terzo o di ricorrere a un arbitro.


## 2.2. Requisiti Tecnici

- **Ambiente di Sviluppo Unificato:** Il software dovrà essere sviluppato utilizzando lo stesso ambiente di sviluppo sia sul Raspberry Pi (ambiente di test/sviluppo) che nel cloud (ambiente di produzione). Questo garantirà coerenza e faciliterà la migrazione e il deployment del software.
- **Scalabilità:** L'architettura del software dovrà essere progettata per essere scalabile, in modo da poter gestire un numero crescente di utenti e impianti senza compromettere le prestazioni. Si dovrà utilizzare un'architettura a microservizi o un'architettura serverless per garantire la massima flessibilità e scalabilità.
- **Affidabilità e Disponibilità:** Il software dovrà essere affidabile e garantire un alto livello di disponibilità, anche in caso di guasti o picchi di utilizzo. Si dovranno implementare meccanismi di ridondanza, backup e disaster recovery per garantire la continuità del servizio.

- 
- **Sicurezza:** Il software dovrà garantire la sicurezza dei dati degli utenti e delle informazioni sensibili, implementando misure di sicurezza adeguate come la crittografia dei dati, l'autenticazione degli utenti e la protezione da attacchi informatici.
  - **Interoperabilità:** Il software dovrà essere in grado di interagire con i sistemi esterni, come i portali del GSE e dei gestori di rete, per l'acquisizione dei dati di misura e la gestione delle pratiche amministrative. Si dovranno utilizzare API (Application Programming Interface) standard e protocolli di comunicazione sicuri.
  - **Usabilità:** L'interfaccia utente del software dovrà essere intuitiva e facilmente utilizzabile dai referenti delle CER/CEC, anche senza particolari competenze tecniche. Si dovrà prestare particolare attenzione alla progettazione dell'esperienza utente (UX) per garantire un'interazione semplice ed efficace con il software.
  - **Prestazioni:** Il software dovrà garantire prestazioni elevate, anche in caso di utilizzo intensivo, per offrire un'esperienza utente fluida e reattiva. Si dovranno ottimizzare le query al database, utilizzare tecniche di caching e adottare altre strategie per migliorare le prestazioni.
  - **Manutenibilità:** Il codice sorgente del software dovrà essere chiaro, organizzato e ben documentato, per facilitare la manutenzione e l'aggiornamento da parte della community di sviluppatori. Si dovranno seguire le migliori pratiche di sviluppo software e utilizzare strumenti di testing automatico per garantire la qualità del codice.

## 2.3. Requisiti Legali

- **Privacy (GDPR):** Il software dovrà essere conforme al Regolamento Generale sulla Protezione dei Dati (GDPR), garantendo la privacy degli utenti e la sicurezza dei loro dati personali. Dovrà essere implementata una chiara informativa sulla privacy, ottenere il consenso degli utenti per il trattamento dei dati e garantire i loro diritti (accesso, rettifica, cancellazione, ecc.).
- **Sicurezza delle Informazioni:** Il software dovrà adottare misure di sicurezza adeguate per proteggere i dati e le informazioni sensibili delle CER/CEC da accessi non autorizzati, perdite di dati e altre minacce informatiche.
- **Proprietà Intellettuale:** Il software dovrà rispettare le norme sulla proprietà intellettuale, garantendo che il codice sorgente e gli altri materiali protetti da copyright siano utilizzati in conformità con la licenza MIT. Si dovrà inoltre verificare che eventuali componenti di terze parti utilizzati nel software siano compatibili con la licenza open source del progetto.
- **Normativa Fiscale:** Il software dovrà essere conforme alla normativa fiscale italiana, gestendo correttamente la fatturazione degli incentivi e dei contributi e il pagamento dei corrispettivi dovuti al GSE. Dovranno essere implementati meccanismi per la generazione di fatture elettroniche e l'integrazione con i sistemi di pagamento elettronico (ad esempio, PagoPA).
- **Antimafia:** Nel caso in cui le CER/CEC siano soggette alla verifica antimafia, il software dovrà supportare la raccolta e la trasmissione al GSE della documentazione richiesta ai sensi del D. Lgs. 159/2011 e ss.mm.ii.
- **Trasparenza e Rendicontazione:** Il software dovrà garantire la trasparenza delle operazioni e delle informazioni relative alle CER/CEC, consentendo la generazione di report dettagliati e la verifica della



conformità ai requisiti normativi. Dovrà inoltre supportare la rendicontazione delle attività al GSE e al Ministero dell'Ambiente e della Sicurezza Energetica, secondo le modalità e le tempistiche previste dalla normativa.

- **Contributi PNRR:** Il software dovrà garantire il rispetto dei requisiti legali relativi ai contributi del PNRR, come il divieto di doppio finanziamento e la necessità di rendicontare le spese sostenute in modo trasparente e verificabile.

## 3. Progettazione dell'Architettura

### 3.1. Architettura Generale

L'architettura del software CerCollettiva sarà progettata per essere modulare, scalabile e flessibile, in modo da potersi adattare alle diverse esigenze delle comunità energetiche e alle evoluzioni del quadro normativo. Per raggiungere questo obiettivo, si utilizzerà un'architettura a microservizi, che consente di suddividere l'applicazione in componenti più piccoli e indipendenti, ognuno responsabile di una specifica funzionalità.

#### Microservizi

I microservizi comunicano tra loro attraverso API (Application Programming Interface) leggere e ben definite, come REST (Representational State Transfer) o GraphQL. Ogni microservizio può essere sviluppato, distribuito e scalato in modo indipendente dagli altri, garantendo una maggiore flessibilità e facilità di manutenzione. Inoltre, l'architettura a microservizi consente di utilizzare tecnologie diverse per ogni microservizio, scegliendo quelle più adatte per la specifica funzionalità. In particolare, il framework Laravel, basato su PHP, verrà utilizzato per la realizzazione dei microservizi, sfruttando le sue potenzialità in termini di struttura, sicurezza e gestione delle richieste.

#### Componenti Principali

I principali microservizi che comporranno l'architettura del software sono:

- **Microservizio Utenti:** Gestisce la registrazione, l'autenticazione e la profilazione degli utenti, la verifica dei requisiti di ammissibilità e l'assegnazione dei ruoli.
- **Microservizio Impianti:** Gestisce il catalogo degli impianti, la verifica dei requisiti tecnici e normativi, l'acquisizione dei dati di misura e il calcolo dell'energia condivisa.
- **Microservizio Incentivi:** Gestisce il calcolo degli incentivi e dei contributi, la generazione delle fatture e la gestione dei pagamenti.

- **Microservizio Amministrazione:** Gestisce gli aspetti amministrativi delle CER/CEC, come la gestione dei documenti, le comunicazioni e la rendicontazione.
- **Microservizio PNRR:** (Opzionale) Gestisce le domande di contributo PNRR, la verifica dei requisiti e il calcolo dell'ammontare del contributo.

## **Interazione tra Microservizi**

I microservizi interagiranno tra loro attraverso API RESTful o GraphQL, utilizzando un protocollo di comunicazione sicuro come HTTPS. Ad esempio, il microservizio Incentivi richiederà i dati di misura al microservizio Impianti per calcolare gli incentivi, mentre il microservizio Amministrazione richiederà i dati degli utenti al microservizio Utenti per inviare comunicazioni personalizzate.

## **Frontend**

Il frontend del software sarà sviluppato utilizzando tecnologie web standard come HTML, CSS e JavaScript. Si potrà valutare l'utilizzo di un framework JavaScript moderno come Vue.js o React per creare un'interfaccia utente reattiva e intuitiva. Il frontend comunicherà con i microservizi attraverso le API RESTful o GraphQL.

## **Infrastruttura Cloud**

L'architettura a microservizi è particolarmente adatta per il deployment su un'infrastruttura cloud, come Amazon Web Services (AWS), Google Cloud Platform (GCP) o Microsoft Azure. Ogni microservizio potrà essere eseguito in un container Docker, garantendo portabilità e facilità di gestione. L'infrastruttura cloud fornirà inoltre servizi di bilanciamento del carico, auto-scaling e monitoraggio per garantire prestazioni elevate e affidabilità.



## 3.2. Modello dei Dati

Il modello dei dati del software CerCollettiva sarà progettato per rappresentare in modo efficiente e accurato le informazioni relative alle Comunità Energetiche Rinnovabili (CER) e alle Comunità Energetiche dei Cittadini (CEC), nonché i dati necessari per la gestione degli impianti, il calcolo degli incentivi e dei contributi e gli adempimenti amministrativi.

### Entità Principali

Le principali entità del modello dei dati saranno:

- **Utente:** Rappresenta i membri delle CER/CEC (cittadini, imprese, enti locali).  
Attributi: ID, nome, cognome/ragione sociale, codice fiscale/partita IVA, indirizzo, contatti, ruolo (produttore, consumatore), dati bancari, consenso al trattamento dei dati, ecc.
- **Impianto:** Rappresenta gli impianti di produzione di energia rinnovabile. Attributi: ID, tipologia (fotovoltaico, eolico, ecc.), potenza, ubicazione (dati catastali), data di entrata in esercizio, componenti installati, dati di misura (produzione, immissione, autoconsumo), ecc.
- **Comunità Energetica (CER/CEC):** Rappresenta la comunità energetica. Attributi: ID, nome, tipologia (CER o CEC), data di costituzione, statuto/atto costitutivo, elenco dei membri, dati del referente, ecc.
- **Misura:** Rappresenta i dati di misura dell'energia (produzione, immissione in rete, autoconsumo) rilevati dai contatori. Attributi: ID, timestamp, valore, tipologia (produzione, immissione, autoconsumo), impianto di riferimento, ecc.
- **Incentivo/Contributo:** Rappresenta gli incentivi e i contributi spettanti alle CER/CEC. Attributi: ID, tipologia (tariffa incentivante, contributo di valorizzazione, contributo PNRR), importo, data di erogazione, impianto o configurazione di riferimento, ecc.

- **Documento:** Rappresenta i documenti amministrativi delle CER/CEC. Attributi: ID, nome, tipologia (statuto, contratto, fattura, ecc.), data, file, ecc.

## Relazioni

Le principali relazioni tra le entità saranno:

- **Utente - Comunità Energetica:** Un utente può essere membro di una o più comunità energetiche.
- **Impianto - Comunità Energetica:** Un impianto può appartenere a una sola comunità energetica.
- **Misura - Impianto:** Una misura è associata a un impianto specifico.
- **Incentivo/Contributo - Impianto/Comunità Energetica:** Un incentivo o contributo può essere associato a un impianto specifico o all'intera comunità energetica.
- **Documento - Comunità Energetica:** Un documento è associato a una comunità energetica.

## Considerazioni Aggiuntive

Il modello dei dati dovrà essere progettato per essere flessibile e facilmente estensibile, in modo da poter accogliere nuove funzionalità e requisiti in futuro. Si dovrà inoltre prestare particolare attenzione alla normalizzazione dei dati per evitare ridondanze e inconsistenze.

### 3.3. Interfaccia Utente (UI) e Esperienza Utente (UX)

L'interfaccia utente (UI) del software CerCollettiva sarà progettata per essere intuitiva, user-friendly e accessibile a utenti con diverse competenze tecniche. L'obiettivo è garantire un'esperienza utente (UX) positiva, che faciliti l'interazione con il software e la comprensione delle informazioni.



## Principi di Design


- **Semplicità:** L'interfaccia dovrà essere semplice e pulita, evitando sovraccarichi di informazioni e elementi grafici superflui.
- **Chiarezza:** Le informazioni dovranno essere presentate in modo chiaro e organizzato, utilizzando un linguaggio semplice e comprensibile.
- **Coerenza:** L'interfaccia dovrà essere coerente in termini di layout, colori, tipografia e iconografia, per facilitare la navigazione e l'apprendimento dell'utente.
- **Accessibilità:** L'interfaccia dovrà essere accessibile a utenti con disabilità, seguendo le linee guida WCAG (Web Content Accessibility Guidelines).
- **Responsività:** L'interfaccia dovrà adattarsi a diversi dispositivi (desktop, tablet, smartphone) per garantire un'esperienza utente ottimale su ogni schermo.

## Elementi dell'Interfaccia

- **Dashboard:** Una dashboard personalizzabile che fornisce una panoramica delle informazioni più importanti per l'utente (produzione di energia, consumi, incentivi, ecc.).
- **Navigazione:** Un menu di navigazione chiaro e intuitivo che consente di accedere alle diverse sezioni del software.
- **Moduli di Input:** Moduli intuitivi per l'inserimento dei dati, con validazione in tempo reale e suggerimenti per facilitare la compilazione.
- **Visualizzazione dei Dati:** Grafici, tabelle e mappe interattive per visualizzare i dati in modo chiaro e intuitivo.
- **Notifiche:** Un sistema di notifiche per informare l'utente di eventi importanti (anomalie, scadenze, aggiornamenti, ecc.).

## Test Utente

Durante la fase di sviluppo, verranno condotti test utente per raccogliere feedback e migliorare l'usabilità dell'interfaccia. I test coinvolgeranno utenti rappresentativi del



target di riferimento (referenti di CER/CEC) e saranno utilizzati per valutare l'efficacia del design, l'intuitività della navigazione e la chiarezza delle informazioni presentate.

## **API (Application Programming Interface)**

Le API (Application Programming Interface) saranno un elemento fondamentale dell'architettura del software CerCollettiva, consentendo la comunicazione tra i diversi microservizi e l'interazione con sistemi esterni.

### **API RESTful**

Si utilizzerà un'API RESTful (Representational State Transfer) per esporre le funzionalità dei microservizi. Le API RESTful sono basate su standard web come HTTP e JSON (JavaScript Object Notation) e offrono un'interfaccia semplice e flessibile per l'interazione con i dati.

### **Autenticazione e Autorizzazione**


Le API saranno protette da meccanismi di autenticazione e autorizzazione per garantire la sicurezza dei dati e l'accesso alle funzionalità solo agli utenti autorizzati. Si utilizzeranno token JWT (JSON Web Token) o OAuth 2.0 per l'autenticazione e si definiranno di ruoli e permessi per l'autorizzazione.

### **Documentazione**

Le API saranno documentate in modo dettagliato, utilizzando uno standard come OpenAPI (precedentemente noto come Swagger). La documentazione includerà informazioni su endpoint, metodi, parametri, risposte e codici di errore, facilitando l'integrazione con altri sistemi e lo sviluppo di applicazioni client.

### **API Gateway**

Si utilizzerà un API Gateway per gestire l'accesso alle API, il routing delle richieste, l'autenticazione, la limitazione della frequenza (rate limiting) e altre funzionalità di



sicurezza e gestione. L'API Gateway fungerà da punto di ingresso unico per tutte le richieste alle API, semplificando l'architettura e migliorando la scalabilità.

### 3.4. API (Application Programming Interface)

Le API (Application Programming Interface) saranno un elemento fondamentale dell'architettura del software CerCollettiva, consentendo la comunicazione tra i diversi microservizi e l'interazione con sistemi esterni. Le API RESTful, basate su standard web come HTTP e JSON, offriranno un'interfaccia semplice e flessibile per l'interazione con i dati. Saranno protette da meccanismi di autenticazione e autorizzazione per garantire la sicurezza, utilizzando token JWT o OAuth 2.0. La documentazione dettagliata delle API, seguendo standard come OpenAPI, faciliterà l'integrazione con altri sistemi e lo sviluppo di applicazioni client. Inoltre, un API Gateway gestirà l'accesso, il routing, l'autenticazione e altre funzionalità di sicurezza.

#### Endpoint Principali

Di seguito sono riportati alcuni esempi di endpoint che potrebbero essere esposti dalle API di CerCollettiva:

- **Microservizio Utenti**
  - **POST /api/users**: Registrazione di un nuovo utente
  - **POST /api/login**: Autenticazione di un utente
  - **GET /api/users/{id}**: Recupero dei dati di un utente specifico
  - **PUT /api/users/{id}**: Aggiornamento dei dati di un utente
- **Microservizio Impianti**
  - **POST /api/plants**: Registrazione di un nuovo impianto
  - **GET /api/plants/{id}**: Recupero dei dati di un impianto specifico
  - **PUT /api/plants/{id}**: Aggiornamento dei dati di un impianto
  - **POST /api/plants/{id}/readings**: Invio di dati di misura da parte di un impianto
- **Microservizio Incentivi**
  - **GET /api/incentives/{id}**: Calcolo dell'incentivo per una CER/CEC

- **POST /api/invoices:** Generazione di una fattura elettronica
- **Microservizio Amministrazione**
  - **GET /api/reports:** Generazione di report periodici
  - **POST /api/documents:** Caricamento di un documento

## API per l'Integrazione con Sistemi Esterni

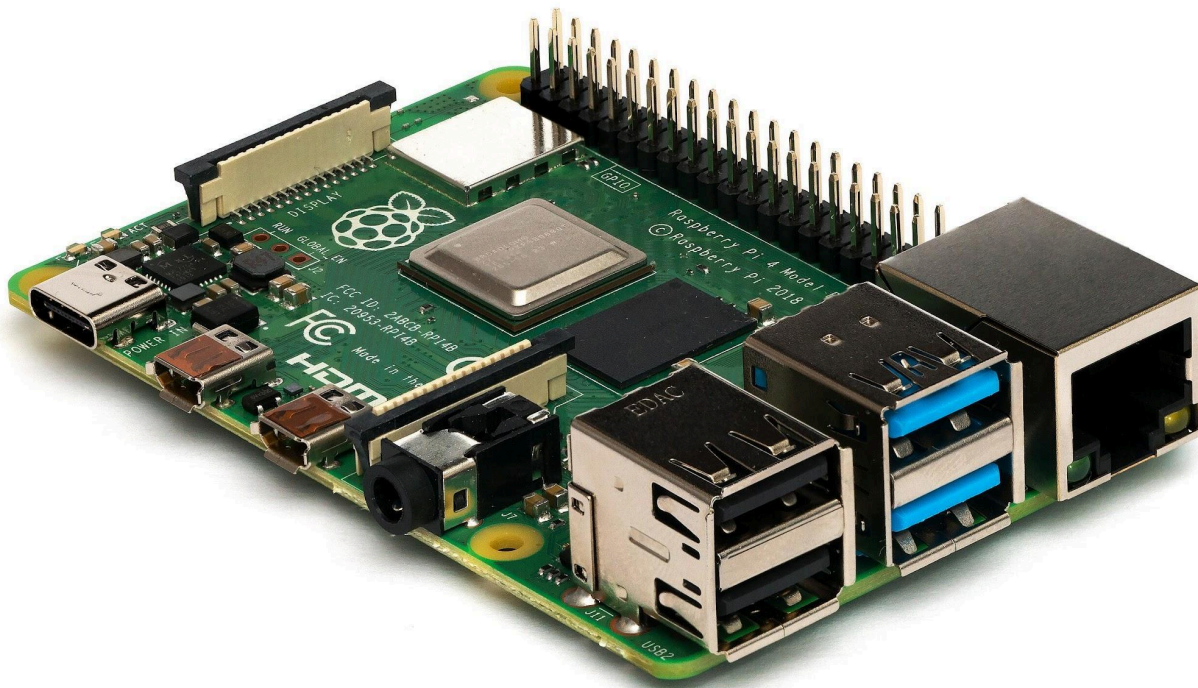
Oltre agli endpoint per l'interazione tra i microservizi, saranno sviluppate API specifiche per l'integrazione con sistemi esterni, come:

- **GSE:** Per l'invio delle domande di accesso agli incentivi e la rendicontazione delle attività.
- **Gestori di Rete:** Per l'acquisizione dei dati di misura e la gestione delle comunicazioni.
- **PagoPA:** Per la gestione dei pagamenti elettronici.
- **Mosquitto:** Per la creazione e la gestione degli account utente del broker MQTT.  
Un esempio di endpoint potrebbe essere **POST /api/mqtt/users** per la creazione di un nuovo account utente con le relative credenziali.

Queste API dovranno essere conformi agli standard e ai protocolli utilizzati dai sistemi esterni e garantire la sicurezza e l'integrità dei dati scambiati.

## 4. Sviluppo del Prototipo su Raspberry Pi

### 4.1. Ambiente di Sviluppo



Il Raspberry Pi 4 Model B sarà configurato con un ambiente di sviluppo LAMP (Linux, Apache, MariaDB, PHP), che fornirà tutti gli strumenti necessari per lo sviluppo e il test del software CerCollettiva. In particolare verranno installati:

- **Sistema Operativo:** Raspbian (Raspberry Pi OS), una distribuzione Linux ottimizzata per il Raspberry Pi, basata su Debian.
- **Web Server:** Apache, un web server open source ampiamente utilizzato per applicazioni web PHP.
- **Database:** MariaDB, un sistema di gestione di database relazionale open source, fork di MySQL, compatibile con quest'ultimo e ottimizzato per il Raspberry Pi.
- **PHP:** Il linguaggio di programmazione PHP nella versione 8.1, insieme all'estensione PHP-FPM (FastCGI Process Manager) per migliorare le prestazioni e la gestione delle risorse.

- **Composer:** Il gestore di pacchetti per PHP, per installare le dipendenze di Laravel e altre librerie necessarie.
- **Node.js e NPM:** Per la gestione delle dipendenze frontend (come moduli JavaScript, CSS e altri asset) e l'utilizzo di strumenti come Laravel Mix per la compilazione e l'ottimizzazione degli asset frontend.
- **Mosquitto:** Il broker MQTT per la gestione della comunicazione in tempo reale tra i dispositivi IoT (ad esempio, i contatori intelligenti) e il software CerCollettiva. Mosquitto è un broker leggero e performante, ideale per l'utilizzo su dispositivi embedded come il Raspberry Pi. Sarà configurato per supportare la ricezione dei dati di misura dai dispositivi IoT e la loro memorizzazione nel database.

## Configurazione dell'Ambiente

1. **Installazione del Sistema Operativo:** Installare il sistema operativo Raspbian (Raspberry Pi OS) sul Raspberry Pi seguendo le istruzioni ufficiali.
2. **Aggiornamento del Sistema:** Aggiornare il sistema operativo e i pacchetti installati utilizzando i comandi `sudo apt update` e `sudo apt upgrade`.
3. **Installazione di Apache, MariaDB e PHP:** Installare i pacchetti necessari utilizzando il comando `sudo apt install apache2 mariadb-server php8.1 php8.1-fpm libapache2-mod-php8.1`. Verificare il corretto funzionamento di Apache e PHP aprendo un browser e navigando all'indirizzo `http://localhost`. Dovrebbe essere visualizzata la pagina di benvenuto di Apache.
4. **Configurazione di MariaDB:** Impostare la password di root di MariaDB utilizzando il comando `sudo mysql_secure_installation`. Creare un nuovo database per CerCollettiva e un utente con i permessi necessari per accedere al database.
5. **Installazione di Laravel:** Installare Laravel globalmente utilizzando Composer con il comando `composer global require laravel/installer`. Creare un nuovo progetto Laravel nella directory `/var/www/html` utilizzando il comando `laravel new cercollettiva`.



6. **Installazione di Mosquitto:** Installare il broker MQTT Mosquitto utilizzando il comando `sudo apt install mosquitto mosquitto-clients`. Configurare Mosquitto per l'autenticazione degli utenti e la gestione dei topic. Sarà necessario creare un utente e una password per l'accesso al broker da parte del software CerCollettiva.
7. **Installazione di Node.js e NPM:** Installare Node.js e NPM utilizzando il comando `sudo apt install nodejs npm`. Utilizzare NPM per installare le dipendenze frontend del progetto Laravel (ad esempio, `npm install`).
8. **Configurazione di Apache:** Configurare Apache per utilizzare PHP-FPM e servire il progetto Laravel. Creare un nuovo file di configurazione per il virtual host in `/etc/apache2/sites-available/cercollettiva.conf` e abilitarlo utilizzando il comando `sudo a2ensite circollettiva.conf`.

Una volta completata la configurazione dell'ambiente di sviluppo, sarà possibile iniziare a sviluppare il prototipo del software CerCollettiva sul Raspberry Pi.

## 4.2. Linguaggio di Programmazione e Framework

Il linguaggio di programmazione principale sarà PHP nella versione 8.1, utilizzato in combinazione con il framework Laravel. Questa scelta è motivata da diversi fattori:

- **PHP:** PHP è un linguaggio di scripting lato server ampiamente utilizzato per lo sviluppo web, con una vasta community di sviluppatori e numerose risorse disponibili. È particolarmente adatto per applicazioni che richiedono l'interazione con database e la gestione di dati, come CerCollettiva. La versione 8.1 offre miglioramenti in termini di prestazioni, sicurezza e nuove funzionalità rispetto alle versioni precedenti.
- **Laravel:** Laravel è un framework PHP moderno, potente e open-source, che segue l'architettura MVC (Model-View-Controller) e offre una struttura solida e numerosi strumenti per lo sviluppo rapido di applicazioni web scalabili e sicure. Alcune delle caratteristiche principali di Laravel che lo rendono adatto per il progetto CerCollettiva sono:

- **Eloquent ORM:** Un potente ORM (Object-Relational Mapping) che semplifica l'interazione con il database, consentendo di definire modelli per le entità del modello dei dati (utenti, impianti, misure, ecc.) e di eseguire query in modo semplice ed espressivo.
- **Routing:** Un sistema flessibile per la definizione delle rotte dell'applicazione, che permette di mappare gli URL alle azioni dei controller.
- **Blade Templating:** Un motore di template semplice e intuitivo per la creazione delle viste, che consente di separare la logica di presentazione dalla logica dell'applicazione.
- **Artisan:** Un'interfaccia a riga di comando potente e versatile, che fornisce comandi per generare codice, eseguire migrazioni del database, gestire le attività pianificate e automatizzare altre operazioni.
- **Middleware:** Un meccanismo per filtrare le richieste HTTP in ingresso e in uscita, che permette di gestire l'autenticazione, l'autorizzazione, la registrazione dei log e altre funzionalità trasversali.
- **Sicurezza:** Laravel offre funzionalità integrate per la protezione da attacchi comuni come l'iniezione SQL, il cross-site scripting (XSS) e la falsificazione di richieste cross-site (CSRF).
- **Testing:** Laravel fornisce strumenti per scrivere e eseguire test unitari e di funzionalità, facilitando il processo di sviluppo e garantendo la qualità del codice.

## Vantaggi dell'utilizzo di Laravel

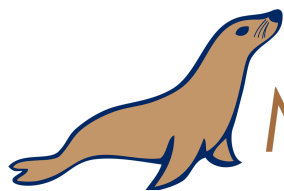
L'utilizzo di Laravel per lo sviluppo del software CerCollettiva offre numerosi vantaggi, tra cui:

- **Sviluppo rapido:** Laravel accelera lo sviluppo grazie alla sua struttura modulare, ai numerosi strumenti integrati e alla vasta community di sviluppatori.

- **Codice pulito e manutenibile:** Laravel incoraggia l'adozione di buone pratiche di sviluppo, come la separazione delle responsabilità e l'utilizzo di design pattern, che rendono il codice più leggibile, manutenibile e scalabile.
- **Sicurezza:** Laravel offre funzionalità integrate per la sicurezza delle applicazioni web, proteggendo da vulnerabilità comuni e facilitando l'implementazione di misure di sicurezza aggiuntive.
- **Scalabilità:** L'architettura di Laravel, basata su componenti indipendenti e facilmente sostituibili, rende il software più scalabile e adattabile alle esigenze future.
- **Community:** Laravel vanta una vasta community di sviluppatori che contribuiscono alla creazione di pacchetti, estensioni e risorse utili, offrendo supporto e soluzioni a problemi comuni.

In sintesi, la scelta di PHP come linguaggio di programmazione e di Laravel come framework si rivela strategica per lo sviluppo di CerCollettiva, garantendo un ambiente di sviluppo solido, efficiente e sicuro, in grado di supportare la crescita e l'evoluzione del progetto nel tempo.

### 4.3. Database



MariaDB

Il database MariaDB, installato sul Raspberry Pi, sarà utilizzato per memorizzare in modo strutturato e persistente tutti i dati dell'applicazione

CerCollettiva. Grazie all'utilizzo del framework Laravel e del suo ORM Eloquent, l'interazione con il database sarà semplificata e intuitiva. Eloquent consente infatti di definire modelli che rappresentano le entità del modello dei dati (utenti, impianti, comunità energetiche, misure, incentivi, documenti, ecc.) e di eseguire operazioni CRUD (Create, Read, Update, Delete) in modo efficiente e sicuro.

#### Struttura del Database



Le tabelle principali del database saranno le seguenti:

- **users:** Conterrà le informazioni sugli utenti registrati nel sistema, inclusi dati anagrafici, di contatto, **tipologia di utente (Produttore, Consumatore, Prosumer)**, ruolo nella comunità energetica e dati fiscali.
- **plants:** Conterrà le informazioni sugli impianti di produzione di energia rinnovabile, come tipologia, potenza, ubicazione, dati catastali, codice POD, componenti installati, ecc.
- **communities:** Conterrà le informazioni sulle comunità energetiche, inclusi nome, tipologia (CER o CEC), data di costituzione, statuto, elenco dei membri, dati del referente, ecc.
- **readings:** Conterrà i dati di misura dell'energia (produzione, immissione in rete, autoconsumo) rilevati dai contatori, associati a ciascun impianto.
- **incentives:** Conterrà le informazioni sugli incentivi e i contributi spettanti alle comunità energetiche e ai singoli membri, inclusi tipologia, importo, data di erogazione, ecc.
- **documents:** Conterrà i documenti amministrativi delle comunità energetiche, come statuti, contratti, fatture, ecc.

Oltre a queste tabelle principali, saranno create altre tabelle per gestire le relazioni tra le entità, come ad esempio tabelle di collegamento tra utenti e comunità energetiche, tra impianti e comunità energetiche, tra misure e impianti, ecc.

## Migrazioni

Laravel fornisce un sistema di migrazioni che consente di creare e modificare la struttura del database in modo versionato e controllato. Le migrazioni sono file PHP che definiscono le operazioni da eseguire sul database (creazione di tabelle, aggiunta di colonne, modifica di indici, ecc.). Grazie alle migrazioni, sarà possibile mantenere il database sincronizzato con il codice dell'applicazione e facilitare il deployment del software su diversi ambienti.

## Seed

Laravel fornisce anche un meccanismo di seed per popolare il database con dati iniziali. Ad esempio, si potranno creare seed per inserire nel database le tipologie di impianto ammesse, i codici POD dei gestori di rete, ecc. I seed saranno utili durante la fase di sviluppo e test del software.

## Ottimizzazione

Il database sarà ottimizzato per garantire prestazioni elevate, anche con un numero elevato di utenti e impianti. Si utilizzeranno indici appropriati per velocizzare le query, si adotteranno tecniche di caching per ridurre il carico sul database e si imposteranno parametri di configurazione ottimali per il server MariaDB.

### 4.4. Broker MQTT

Il broker MQTT Mosquitto verrà installato sul Raspberry Pi per gestire la comunicazione in tempo reale tra i dispositivi IoT (ad esempio, i contatori intelligenti) e il software CerCollettiva. Mosquitto è un broker leggero e performante, ideale per l'utilizzo su dispositivi embedded come il Raspberry Pi.

La configurazione di Mosquitto includerà:

- **Autenticazione degli utenti:** Ogni dispositivo IoT dovrà autenticarsi al broker utilizzando credenziali univoche (username e password) generate e gestite dal software CerCollettiva. Questo garantirà che solo i dispositivi autorizzati possano inviare dati di misura.
- **Gestione dei topic:** I dati di misura saranno pubblicati e sottoscritti su specifici topic MQTT, organizzati in base alla comunità energetica e all'impianto di riferimento. Questo permetterà di filtrare e indirizzare i dati in modo efficiente.
- **Persistenza dei messaggi:** I messaggi MQTT saranno resi persistenti, in modo che possano essere recuperati anche in caso di disconnessione temporanea del dispositivo IoT o del software CerCollettiva.

- **Sicurezza:** Mosquitto sarà configurato per utilizzare il protocollo TLS (Transport Layer Security) per crittografare la comunicazione tra i dispositivi IoT e il broker, garantendo la riservatezza e l'integrità dei dati.

Il software CerCollettiva interagirà con Mosquitto tramite una libreria client MQTT per PHP, come ad esempio `php-mqtt/client`. Questa libreria consentirà di connettersi al broker, sottoscrivere i topic di interesse e ricevere i messaggi contenenti i dati di misura. I dati ricevuti saranno quindi elaborati e memorizzati nel database MariaDB per essere utilizzati nelle funzionalità di calcolo dell'energia condivisa, determinazione degli incentivi e generazione di report.

### **Creazione e gestione degli account Mosquitto**


Il software CerCollettiva sarà responsabile della creazione e gestione degli account utente di Mosquitto. Per ogni dispositivo IoT che verrà registrato nell'impianto, CerCollettiva genererà automaticamente un account utente con credenziali univoche e sicure. Queste credenziali saranno memorizzate nel database in modo sicuro e associate all'impianto corrispondente. L'interfaccia utente di CerCollettiva fornirà una sezione dedicata alla gestione degli account Mosquitto, consentendo agli utenti autorizzati di visualizzare, modificare o eliminare le credenziali di accesso per ciascun dispositivo.

Inoltre, CerCollettiva implementerà meccanismi di controllo e monitoraggio per garantire la sicurezza degli account Mosquitto, rilevando eventuali anomalie o tentativi di accesso non autorizzato e adottando misure correttive appropriate.

## 4.5. Funzionalità del Prototipo

Il prototipo sviluppato su Raspberry Pi includerà le funzionalità chiave del software CerCollettiva, come:

1. **Registrazione e autenticazione degli utenti:** Gli utenti potranno creare un account, accedere al sistema e gestire il proprio profilo. L'autenticazione potrà avvenire tramite credenziali tradizionali (email e password) o utilizzando sistemi di identità digitale come SPID o CIE.
2. **Creazione e gestione delle comunità energetiche (CER/CEC):** Gli utenti potranno creare nuove comunità energetiche, invitare altri membri, definire le regole di funzionamento (come i criteri di ripartizione degli incentivi) e gestire gli aspetti amministrativi della comunità.
3. **Registrazione degli impianti e verifica dei requisiti:** Il backoffice, ovvero il personale autorizzato, registrerà gli impianti di produzione di energia rinnovabile, inserendo informazioni dettagliate sulla tipologia, la potenza, l'ubicazione e le caratteristiche tecniche. Il software verificherà automaticamente la conformità degli impianti ai requisiti tecnici e normativi previsti dal Decreto CACER e dal TIAD. Una volta registrato, l'impianto sarà assegnato all'utente corrispondente.
4. **Acquisizione e visualizzazione dei dati di misura:** Il software acquisirà i dati di misura dell'energia prodotta, immessa in rete e autoconsumata dai dispositivi IoT (contatori intelligenti) tramite il broker MQTT Mosquitto. I dati saranno visualizzati in tempo reale e su base storica attraverso grafici, tabelle e mappe interattive, consentendo agli utenti di monitorare l'andamento della produzione e del consumo energetico della comunità.
5. **Calcolo dell'energia condivisa:** Il software calcolerà automaticamente l'energia condivisa (autoconsumata e immessa in rete) in base alle regole del TIAD, tenendo conto delle diverse tipologie di configurazione (autoconsumatore individuale, gruppo di autoconsumatori, CER) e delle diverse tipologie di utenti (produttori, consumatori, prosumer). Il calcolo dell'energia condivisa sarà



fondamentale per la determinazione degli incentivi e dei contributi spettanti alla comunità energetica e ai singoli membri.

6. **Definizione dei criteri di ripartizione degli incentivi:** Il software consentirà alla CER di definire i propri criteri di ripartizione degli incentivi, scegliendo tra modelli predefiniti o personalizzati. Sarà possibile assegnare quote diverse a ciascun membro in base al suo ruolo nella comunità e al suo contributo energetico. Il software calcolerà automaticamente l'ammontare degli incentivi spettanti a ciascun membro in base ai criteri definiti e ai dati di misura dell'energia.
7. **Gestione degli account Mosquitto:** Il software si occuperà della creazione e gestione degli account utente del broker MQTT Mosquitto. Per ogni dispositivo IoT registrato, genererà automaticamente un account con credenziali univoche e sicure, memorizzandole nel database e associandole all'impianto corrispondente. L'interfaccia utente fornirà una sezione dedicata alla gestione degli account Mosquitto, consentendo di visualizzare, modificare o eliminare le credenziali.

Il prototipo non includerà tutte le funzionalità previste per la versione finale del software, ma sarà sufficiente per testare l'architettura, il modello dei dati e l'interazione tra i microservizi. Inoltre, il prototipo servirà come base per lo sviluppo della versione cloud del software, che verrà successivamente distribuita su un'infrastruttura scalabile e affidabile.



## 5. Migrazione al Cloud e Deployment

- 5.1. Scelta del Provider Cloud
- 5.2. Configurazione dell'Infrastruttura
- 5.3. Deployment del Software
- 5.4. Test e Monitoraggio

## 6. Versione Open Source su GitHub

- 6.1. Creazione del Repository
- 6.2. Gestione delle Versioni e dei Contributi
- 6.3. Documentazione
- 6.4. Community

## 7. Manutenzione e Evoluzione

- 7.1. Aggiornamenti Normativi
- 7.2. Nuove Funzionalità
- 7.3. Supporto agli Utenti
- 7.4. Monitoraggio e Risoluzione Problemi

## Appendice

- A. Glossario dei Termini
- B. Riferimenti Normativi
- C. Diagrammi di Flusso (se necessario)
- D. Wireframe dell'Interfaccia Utente (se necessario)