



Agenzia nazionale per le nuove tecnologie,
l'energia e lo sviluppo economico sostenibile



M S E



Ricerca di Sistema elettrico

Progettazione piattaforma raccolta di dati energetici residenziali

Chiara Foglietta, Dario Masucci, Federica Pascucci, Stefano Panzieri



Report RdS/PTR(2019)/003

PROGETTAZIONE PIATTAFORMA RACCOLTA DI DATI ENERGETICI RESIDENZIALI

Chiara Foglietta, Dario Masucci, Federica Pascucci, Stefano Panzieri (Università Roma Tre)

Dicembre 2019

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dello Sviluppo Economico - ENEA

Piano Triennale di Realizzazione 2019-2021 - I annualità

Obiettivo: Tecnologie

Progetto: Tecnologie per la penetrazione efficiente del vettore elettrico negli usi finali

Work package 1: Local Energy District

Linea di attività 6: Progettazione piattaforma raccolta di dati energetici residenziali

Responsabile del Progetto: Claudia Meloni, ENEA

Responsabile del Work package: Claudia Meloni, ENEA

Il presente documento descrive le attività di ricerca svolte all'interno dell'Accordo di collaborazione "*Progettazione piattaforma raccolta di dati energetici residenziali*"

Responsabile scientifico ENEA: Ing. Fabio Moretti

Responsabile scientifico Università Roma Tre: Prof. Stefano Panzieri

Indice

SOMMARIO	5
1 INTRODUZIONE	6
1.1 OBIETTIVI.....	6
1.2 LA SOLUZIONE PROPOSTA	6
1.3 LE ATTIVITÀ SVOLTE	6
2 ENGINE – FOGLIO DI CALCOLO.....	8
2.1 DESCRIZIONE DELLE FUNZIONALITÀ	8
2.2 DATI DI INPUT	8
2.3 DATI DI OUTPUT.....	9
3 APPLICATION - INTERFACCIA UTENTE.....	10
3.1 ACCESSO ALLA PIATTAFORMA	11
3.2 CREAZIONE UTENZA	11
3.3 CREAZIONE UTENZA	12
3.4 COMPILAZIONE.....	12
3.5 INVIO DEL QUESTIONARIO.....	13
3.6 VISUALIZZAZIONE DEI RISULTATI	13
4 STORAGE - STRUTTURA DEL DATABASE.....	15
4.1 TABELLE E CAMPI	15
4.2 TABELLE E CAMPI	16
5 ORCHESTRATOR - IL DEMONE INFORMATICO.....	19
6 REQUISITI DI PROGETTAZIONE	21
6.1 SPECIFICA DI UN REQUISITO	21
6.2 REQUISITI DELLA PIATTAFORMA	22
6.3 REQUISITI DELL'ELEMENTO APPLICATION.....	23
6.4 REQUISITI DELL'ELEMENTO STORAGE.....	24
6.5 REQUISITI DELL'ELEMENTO ORCHESTRATOR	25
6.6 REQUISITI DELL'ELEMENTO ENGINE.....	26
7 CONCLUSIONI E RISULTATI.....	27
RIFERIMENTI	30

Elenco delle figure

Fig. 1 - Schema logico dei processi gestiti dal web service	10
Fig. 2 - Layout pagina di Login a) all'avvio, b) accesso di un utente registrato, c) accesso di un nuovo utente	11
Fig. 3 - Layout pagina Home	12
Fig. 4 - Layout pagine di compilazione del questionario	13
Fig. 5 - Esempio di layout a) pagina Submit, b) pagina Answer	13
Fig. 6 - Layout pagina Output	14
Fig. 7 - Diagramma EER del database progettato	16
Fig. 8 - Schema di criptazione a chiave simmetrica	17
Fig. 9 - Sintassi query MYSQL per la criptazione AES	17
Fig. 10 - Sintassi query MYSQL criptazione AES con codice hash	18
Fig. 11 - Schema di criptazione a chiave simmetrica e codice hash.....	18
Fig. 12 - Processi gestiti e controllati dal demone informatico	20
Fig. 13 - Schema accessi al database.....	20
Fig. 14 - Schema procedurale della piattaforma informatica	28
Fig. 15 - Modello logico del sistema ICT di supporto al processo di Audit Energetico	28

Elenco delle tabelle

Tabella 1 - Template per la descrizione di un requisito	21
Tabella 2 - Classificazione dei requisiti.....	22

Sommario

L'attività riguarda la progettazione di una piattaforma informatica che deve essere in grado di:

- Acquisire i dati di interesse sui consumi energetici di utenti residenziali;
- Effettuare elaborazioni ad alto livello dei dati a disposizione;
- Rendere fruibili per l'utente le informazioni ottenute a valle dell'elaborazione.

L'architettura informatica della piattaforma prevede due livelli distinti: la parte di *back-end* e la parte di *front-end*. L'obiettivo di questa annualità è di progettare:

- La sezione di gestione, *back-end*, che dovrà elaborare e supervisionare la produzione dei contenuti;
- La sezione applicativa, *front-end*, che dovrà gestire il processo di immissione e visualizzazione dei contenuti da parte dell'utente.

In particolare, l'elaborazione dei dati energetici dei singoli utenti viene effettuata mediante un foglio di calcolo *XLSM* messo a disposizione da ENEA, mentre il processo di supervisione della produzione dei contenuti sarà il focus principale su cui si concentreranno gli sforzi.

Al termine di questa attività ci si aspetta di aver progettato la parte di *back-end* e quella di *front-end* per la corretta scrittura e lettura dei dati sia in una struttura di archiviazione che nel file *XLSM* che elabora i risultati di output. Saranno quindi definiti:

- Il sistema di archiviazione dei dati. In particolare, si dovranno gestire non solo le informazioni riguardanti l'uso dell'energia, ma anche i dati sensibili di ogni utente che decide di partecipare all'audit energetico;
- Il processo informatico che, ciclicamente, è in grado sia di prelevare i dati di interesse dal database e inserirli correttamente nel foglio di calcolo per l'elaborazione dati, che di leggere correttamente i dati prodotti dal foglio di calcolo e procedere alla loro scrittura sul database.

Una volta strutturato il database e definito il processo di gestione delle informazioni, l'obiettivo sarà di progettare il *front-end* della *web application* che avrà il ruolo di interfaccia con l'utente. Si dovrà quindi:

- Definire un insieme di pagine web dinamiche caratterizzate da una serie di *form di inserimento dati* semplici e intuitive;
- Definire una pagina web dinamica tramite la quale l'utente potrà visualizzare i risultati di output ottenuti a valle dell'elaborazione del file *XLSM*.

Gli output previsti per questa attività riguardano le specifiche di progetto che risultano essere preliminari alla successiva fase di implementazione della piattaforma, e fondamentali per una corretta realizzazione dei processi informatici.

Le specifiche rilasciate riguarderanno:

- La strutturazione di un database, che sia compliant con le norme di gestione della riservatezza, per l'archiviazione e la manipolazione dei dati;
- La definizione di un processo informatico per la gestione e supervisione del flusso di informazioni tra il database e il foglio di calcolo;
- La definizione di una *web application* per la raccolta dei dati e la loro successiva visualizzazione.

1 Introduzione

La collaborazione con la Divisione Energia del C.R. ENEA La casaccia è incentrata sullo sviluppo di tecnologie e soluzioni informatiche per la pianificazione strategica degli interventi di riqualificazione energetica degli edifici. Lo strumento che si intende realizzare si rivolge alle province autonome, alle regioni, ma anche ai privati cittadini interessati al servizio.

Si vuole quindi realizzare una piattaforma nazionale per la condivisione di macro-dati e la loro analisi comparativa ad alto livello.

1.1 Obiettivi

Gli obiettivi di questa progettualità risultano essere molteplici e riassumibili come segue:

- Acquisire le informazioni riguardanti i consumi energetici degli utenti.
- Individuare e segnalare le zone geografiche o gli edifici con maggiore necessità di interventi di riqualificazione energetica.
- Fornire un feedback all'utente tramite statistiche, benchmark di riferimento e informazioni sui consumi elettrici.

Oltre a rendere omogenei i dati provenienti da tutti gli utenti e consentire una loro corretta elaborazione e valutazione statistica, questo si può rivelare un utile strumento di programmazione dei controlli ed accertamenti sugli *Attestati di Prestazione Energetica* (APE) raccordandosi ai catasti regionali degli impianti termici.

Si è deciso, quindi, di procedere nella definizione di un ecosistema informatico costituito da processi automatici in grado di acquisire, gestire ed elaborare i dati energetici inseriti dagli utenti.

Per questo motivo, l'obiettivo della presente annualità è di elaborare le linee guida di progettazione della piattaforma software, propedeutiche per la successiva fase di implementazione.

1.2 La soluzione proposta

La proposta di soluzione consiste in una piattaforma, modulare e interoperabile, che dal punto di vista strutturale deve essere suddivisa nei seguenti elementi principali.

- Un elemento per l'elaborazione, *Engine*, che deve elaborare i dati di input e produrre le informazioni di output. In continuità con le annualità precedenti, si fa riferimento allo strumento ottenuto a seguito della ricerca riportata nel Report RdS/PAR2016/009, ossia un complesso foglio di calcolo XLSM.
- Un elemento per l'archiviazione, *Storage*, che deve mantenere in memoria i dati energetici degli utenti e le informazioni elaborate dall'Engine. Questi devono essere resi accessibili e manipolabili.
- Un elemento di gestione delle informazioni, *Orchestrator*, che deve supervisionare il flusso di dati esistente tra Storage ed Engine.
- Un elemento di interfaccia con l'utente, *Application*, così organizzato:
 - Una sezione applicativa, front end, che deve gestire il processo di immissione dei dati e visualizzazione dei contenuti da parte dell'utente. Si farà riferimento alla realizzazione di un applicativo web che prevede un percorso guidato all'inserimento dei dati tramite delle form intuitive e di semplice compilazione.
 - Una sezione di gestione, back end, che deve supervisionare la produzione dei contenuti e gestire il flusso di informazioni tra front end e storage.

1.3 Le attività svolte

In questo ambito progettuale le attività su cui si devono concentrare le attenzioni risultano essere quattro e sono riassumibili come segue.

1. Analizzare il foglio di calcolo XLSM, responsabile dell'elaborazione dei dati, per snellirne la complessità e renderlo conforme ai requisiti del progetto.
2. Definire e progettare il sistema di archiviazione dei dati in base al numero e tipo di input e output di interesse. Si devono gestire le informazioni riguardanti l'uso dell'energia e, eventualmente, anche i dati anagrafici degli utenti che partecipano all'audit.

3. Progettare un processo informatico che ciclicamente sia in grado di:
 - Prelevare i dati di interesse dal database di cui sopra.
 - Inserire correttamente i dati nel file XLSM per l'elaborazione degli output.
 - Leggere correttamente i dati di output prodotti dal foglio XLSM e procedere alla loro scrittura sul database.
4. Progettare il front-end e il back-end della web application che ha il ruolo di interfaccia con l'utente. Si deve definire un insieme di pagine web dinamiche per guidare l'utente nel processo di inserimento dei dati e per la visualizzazione delle informazioni di output.

Di seguito, ognuna delle attività affrontate è caratterizzata e descritta in modo sufficientemente dettagliato per gli scopi di progettazione previsti.

2 Engine – Foglio di calcolo

Poiché esula dalle progettualità assegnate a Roma Tre, del componente Engine viene data una descrizione sommaria con lo scopo di fornire una visione completa della piattaforma.

2.1 Descrizione delle funzionalità

Il componente della piattaforma denominato Engine risulta fondamentale per la valutazione del potenziale di aggregabilità di utenze residenziali, in vista della loro possibile trasformazione in smart home.

Le principali attività espletate da questo modulo sono:

- La definizione delle informazioni di interesse, riguardanti l'utente;
- La raccolta dei dati inseriti dall'utente;
- L'elaborazione dei risultati.

In particolare, l'obiettivo è quello di:

- Predisporre un modulo di aggregazione dei dati provenienti da utenze residenziali;
- Definire una procedura di simulazione volta alla previsione dei consumi energetici reali delle abitazioni associate ai diversi utenti.

Per fare questo verrà definito un foglio di calcolo, realizzato in ambiente Excel [1] e che prevede l'implementazione di macro e funzioni scritte in VBA [2] (Visual Basic for Applications), in grado di caratterizzare la tipologia di utenza ed effettuare un'analisi di benchmarking rispetto all'utenza tipo.

Quindi, per rendere la procedura fruibile anche da un utente non esperto, si dovranno limitare al minimo le richieste di informazioni tecniche e si dovranno individuare le informazioni necessarie per caratterizzare l'utente dal punto di vista elettrico e termico.

Una volta raccolti i dati necessari, si prevede una simulazione in regime dinamico semplificato che consente di stimare i consumi annui, rilevati dalle bollette dei rispettivi fornitori, per un confronto ed un feedback all'utente. Il confronto consente un giudizio sulla tipologia di utente, riferito sia all'entità dei consumi (alti/medi/bassi) sia alla stabilità degli stessi (variabili/costanti, mensili/stagionali).

Il foglio di calcolo permetterà quindi di valutare il potenziale di aggregazione dell'abitazione, sia in termini di utenze già elettriche sia in termini di utenze termiche che possono passare all'elettrico.

2.2 Dati di input

Per poter effettuare l'elaborazione dei dati è necessario avere a disposizione le informazioni dell'utente, non solo riguardanti i consumi registrati in bolletta, ma anche sullo stato e le caratteristiche dell'edificio residenziale in analisi. Devono essere, inoltre, tenuti in considerazione gli usi e le abitudini dell'utente e l'eventuale presenza di impianti tecnologici di qualità.

I principali dati di input, di cui necessita il foglio di calcolo per effettuare la simulazione sono suddivisi nelle macro-classi riassunte di seguito.

- Informazioni generali:
 - La geolocalizzazione dell'edificio (inserendo provincia e comune);
 - Le modalità di occupazione dell'edificio, per fasce orarie.
- Informazioni sull'edificio e sull'abitazione:
 - Altezza e numero di piani dell'edificio, nonché l'epoca di costruzione e la tipologia di rivestimento;
 - Dimensioni e pianta dell'abitazione e il suo collocamento all'interno dell'edificio.
- Informazioni sugli impianti tecnologici:
 - Definizione dell'impianto di riscaldamento (autonomo o centralizzato) e i principali dati che lo caratterizzano;
 - Definizione dell'impianto di raffrescamento tramite dispositivi elettrici (climatizzatori e ventilatori);

- Definizione degli impianti per la preparazione dell'acqua calda sanitaria (caldaia ed impianti elettrici a scaldabagno);
- Informazioni sugli elettrodomestici:
 - Definizione dei grandi elettrodomestici di uso comune (lavatrice, lavastoviglie, frigorifero, forno);
 - Enumerazione e tipologia dei punti luce utilizzati per l'illuminazione dell'abitazione;
 - Enumerazione e tipologia di dispositivi elettronici come televisioni, monitor e personal computer.

2.3 *Dati di output*

I risultati dell'elaborazione effettuata tramite il foglio di calcolo dovranno essere resi disponibili al componente Application della piattaforma informatica che avrà il compito di presentarli all'utente sottoforma di quadri sinottici e grafici.

Per questo l'Engine dovrà essere in grado di elaborare quei feedback e benchmark, riguardanti i consumi energetici, utili all'utente per orientarsi verso un comportamento energetico più consapevole.

I principali dati di output che dovrà elaborare il foglio di calcolo riguardano:

- La ripartizione dei consumi di energia primaria in base al servizio (riscaldamento, raffrescamento, ecc.);
- Tipologia dei carichi elettrici (accumulabili, differibili, non differibili);
- Confronti con benchmark di riferimento;
- Segnalazione di possibili migliori tariffe di fornitura di elettricità e gas;
- Potenziale di risparmio energetico dell'abitazione.

3 Application - Interfaccia utente

Per prima cosa è opportuno fare una distinzione tra ciò che riguarda il front end e ciò che riguarda il back end. Per front end si intende tutta quella parte di web application che è rivolta all'utente, ossia tutte le pagine e le funzionalità ad esse associate che vengono rese disponibili all'utilizzatore. Mentre per back end si intende tutta la struttura della web application che si occupa della gestione e della realizzazione delle funzionalità dell'interfaccia.

La distinzione tra front end e back end è fondamentale in quanto aiuta ad individuare i linguaggi di programmazione che devono essere utilizzati. Questa scelta risulta avere un forte impatto a livello progettuale in quanto determina e condiziona la fattibilità della successiva fase di implementazione.

Per realizzare la parte visibile agli utenti e quella con cui essi interagiscono si utilizzano essenzialmente tre linguaggi di programmazione:

- HTML [3], permette di creare la struttura e il contenuto costruire una web application.
- CSS [4], permette di dotare la pagina web di colori, stile, font e immagini di sfondo.
- JavaScript [5], permette di dare alla pagina maggiore dinamicità e possibilità d'interazione.

Per realizzare la parte di gestione della web application e di memorizzazione dei dati inseriti si suggerisce di utilizzare il linguaggio di programmazione PHP [6] in quanto molto versatile e permette una immediata interazione con gli strumenti di gestione dei database. Grazie all'uso di una sapiente programmazione è possibile integrare i diversi linguaggi di programmazione e realizzare in modo efficiente i comportamenti voluti.

A valle delle considerazioni fatte e tenendo ben presente l'obiettivo dell'attività, si vuole progettare un'interfaccia con l'utente che si occupi della gestione dei seguenti processi:

- Registrazione di una nuova utenza, che deve prevedere la generazione di un codice univoco per il riconoscimento dell'utente.
- Controllo degli accessi alla piattaforma, che deve permettere solo ad un utente registrato di usufruire del servizio di Audit Energetico.
- Compilazione del questionario, che deve guidare l'utente nel processo di compilazione informandolo sulle funzionalità disponibili.
- Scrittura e lettura dal database, che deve gestire l'inserimento delle risposte e la lettura dei risultati dell'Audit.
- Visualizzazione output, che rende disponibili i risultati in un formato di facile e immediata comprensione per l'utente.

I processi sono schematicamente rappresentati in Fig.1.



Fig. 1 - Schema logico dei processi gestiti dal web service

Seguendo il processo di Audit energetico, dalla fase di accesso al portale fino alla visualizzazione dei risultati, viene di seguito introdotta la struttura del front end e le funzionalità definite in back end.

3.1 Accesso alla piattaforma

Si devono prevedere due punti di accesso al processo di Audit Energetico in base al profilo che caratterizza l'utente. In particolare, si possono distinguere due casi:

- L'utente partecipa al progetto "Smart Home" (per maggiori dettagli vedi report RdS/PTR(2019)/002 "Aggiornamento modulo per la consapevolezza energetica di utenti residenziali") e possiede già delle credenziali di accesso all'Audit. Nella pagina del portale "Smart Home" riguardante le applicazioni disponibili, deve essere aggiunto un meccanismo, un pulsante o un link, per poter richiamare il questionario.
L'utente viene quindi indirizzato alla pagina *Home* della piattaforma che si sta progettando, portando con sé, come parametro della chiamata, il codice univoco identificativo che gli è stato assegnato dal portale "Smart Home".
- L'utente non partecipa al progetto "Smart Home" ed ha la possibilità di usufruire dell'Audit in forma anonima creando delle credenziali di accesso. Sulla pagina principale dell'Agenzia, www.enea.it, deve essere prevista una sezione dedicata al processo di compilazione del questionario. Da qui l'utente viene indirizzato alla pagina *Login* della web application.

3.2 Creazione utenza

Gli utenti anonimi devono seguire una procedura particolare per il riconoscimento o la creazione di una nuova utenza. Devono essere indirizzati sulla pagina Login, di cui è riportato un esempio in Fig.2, in cui si forniscono all'utente due possibili opzioni, mutuamente esclusive, e illustrate tramite box testuali:

- Se l'utente è già in possesso di un codice identificativo, utente registrato alla piattaforma, questo può essere digitato nell'apposita form. In questo caso deve essere previsto un processo per controllare che il codice inserito sia effettivamente presente nel database. Nel caso in cui non esiste viene comunicata l'incongruenza, altrimenti l'utente viene indirizzato alla pagina Home portando come parametro il suo codice identificativo.
- Se l'utente non è in possesso del codice identificativo, si deve prevedere una procedura per la generazione automatica di un codice valido. In particolare, la procedura deve essere in grado di generare un codice alfanumerico che non sia già presente nel database e fornirlo in output all'utente. Questo potrà visualizzarlo e inserirlo nell'apposita form. Una volta inserito il codice, l'utente viene indirizzato alla pagina Home portando come parametro il suo codice identificativo.

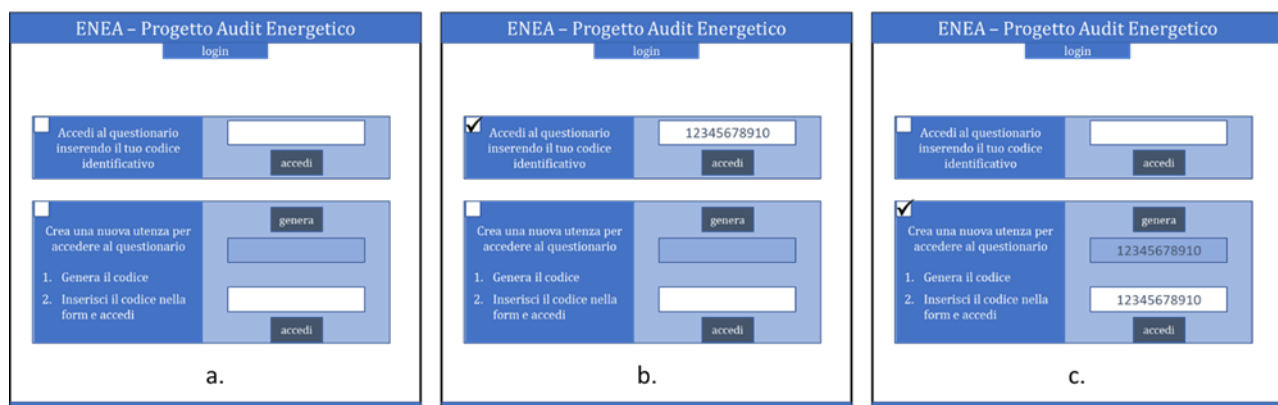


Fig. 2 - Layout pagina di Login a) all'avvio, b) accesso di un utente registrato, c) accesso di un nuovo utente

3.3 Creazione utenza

L'accesso alla pagina Home, il cui layout di esempio è riportato in Fig.3, deve essere permesso solo alle utenze registrate e, al suo avvio, deve essere in grado di effettuare un riconoscimento automatico dell'utente. Si deve quindi prevedere un processo che controlla se il codice identificativo, passato come parametro, è già presente nel database. In questo modo si può distinguere se l'utenza è al suo primo accesso oppure ha già avviato il processo di Audit.

Il riconoscimento dell'utenza risulta essere un'informazione molto utile per predisporre prontamente le funzionalità che le sono permesse. Infatti:

- Se viene rilevata una nuova utenza, questa deve avere la possibilità di iniziare il questionario e quindi essere indirizzata alla prima pagina di Compilazione, contraddistinta dall'id pari a 1.
- Se viene rilevata un'utenza registrata, che ha già iniziato il questionario, devono essere fornite diverse funzionalità:
 - Visualizzare le informazioni inserite, in risposta alle domande del questionario, ed inviate al database.
 - Scegliere, in modo puntuale, da quale macro-argomento riprendere il questionario.
 - Annullare tutte le risposte inserite e iniziare un nuovo questionario.
 - Visualizzare i risultati ottenuti, nel caso l'utente abbia già terminato e inviato le risposte dell'intero questionario.

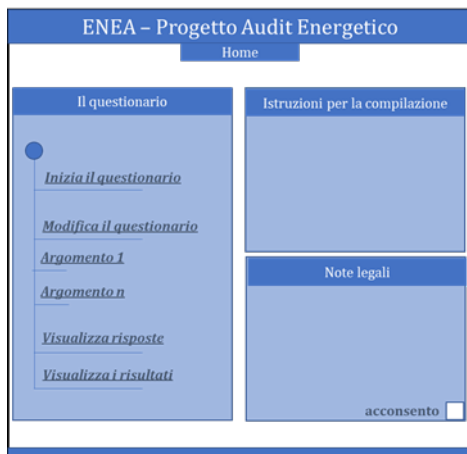


Fig. 3 - Layout pagina Home

3.4 Compilazione

Le pagine web adibite per la compilazione del questionario, *Compilazione_n*, devono avere tutte un layout fisso e caratterizzato dalla presenza di form di inserimento testuale e anche form per l'inserimento dati a risposta chiusa.

Ogni pagina di compilazione deve essere associata ad un macro-argomento riguardante l'Audit Energetico e deve contenere le domande corrispondenti. Inoltre, deve essere caratterizzata da un id progressivo uguale al codice assegnato al macro-argomento, in accordo con quanto riportato nel secondo capitolo.

Le risposte di ogni macro-argomento vengono salvate sul database solo dopo che l'utente le ha tutte terminate e inviate. I dati vengono quindi memorizzati tenendo traccia del macro-argomento appena trattato. L'utenza viene quindi indirizzata alla successiva pagina di compilazione, ovviamente portando con sé il codice identificativo.

Una volta completate le domande dell'ultima pagina di compilazione del questionario l'utente deve essere indirizzato alla pagina *Submit*. Un esempio di possibile layout per le pagine questionario è proposto in Fig.4.



Fig. 4 - Layout pagine di compilazione del questionario

3.5 Invio del questionario

Terminato il questionario l'utente deve avere la possibilità di:

- Visualizzare ancora una volta tutti i dati inseriti prima di procedere con l'elaborazione dei risultati. È opportuno prevedere una pagina dedicata a questo scopo, *Answer* (Fig.5b).
- Inviare le risposte del questionario e aspettare l'elaborazione dei risultati. Quindi viene indirizzato alla pagina *Output*.
- Terminare il processo di Audit Energetico per procedere all'elaborazione dei risultati in un secondo momento.

Queste funzionalità devono essere disponibili nella pagina *Submit* riportata in Fig.5a.

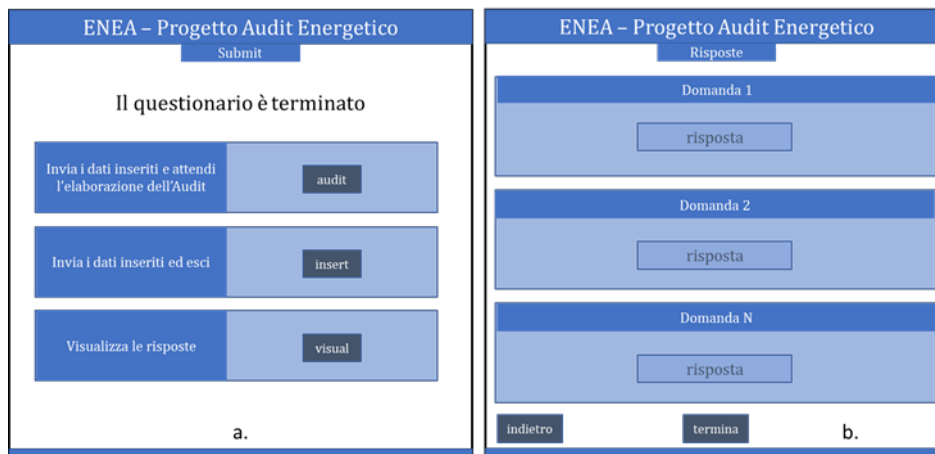


Fig. 5 - Esempio di layout a) pagina Submit, b) pagina Answer

3.6 Visualizzazione dei risultati

La pagina *Output* permette all'utente di visualizzare i risultati ottenuti a valle dell'elaborazione del questionario da parte dell'Engine. Il layout, proposto in Fig.6, deve quindi prevedere la presenza di quadri sinottici e grafici intuitivi per mostrare in modo semplice e immediato quali sono le criticità e i suggerimenti di intervento.

Le funzionalità utilizzabili dall'utente che devono essere previste in questa pagina sono:

- Visualizzare e salvare il questionario compilato.
- Salvare i risultati di output associati al questionario.
- Tornare alla pagina Home della web application.



Fig. 6 - Layout pagina Output

4 Storage - Struttura del database

Considerando la natura pressoché omogenea dei dati da archiviare, si prevede di considerare un database SQL [7] (Structure Query Language). Questo permette la facile integrazione con le strutture dati preesistenti nell'ecosistema informatico implementato al C.R. La Casaccia. Tramite le interrogazioni, o *query*, proprie di questo linguaggio avviene l'interazione con il database e la possibilità di compiere tutte le operazioni di lettura, scrittura e aggiornamento sulle tabelle.

Va da sé che le query devono essere scritte obbligatoriamente seguendo regole ben precise di sintassi, pena il fallimento dell'interrogazione. Per realizzare questo compito è stato scelto uno tra i più diffusi software per la gestione di database, ossia MySQL [8] che costituisce una soluzione flessibile e affidabile per la gestione degli accessi e per eseguire le interrogazioni sul database.

In particolare, MySQL appartiene alla classe di software RDBMS (Relational DataBase Management System) che opera in aderenza con la teoria relazionale definita da Codd secondo cui il sistema deve operare su dati strutturati, all'interno del database, in differenti tabelle, interconnesse tra loro e organizzate secondo la relazione chiave-valore.

MySQL consente di creare connessioni con il database tramite l'indirizzo IP ed eseguire i comandi per la manipolazione dei dati.

Per mantenere compatte e di facile accesso le risorse di archiviazione, si è definita una struttura organizzata in diverse tabelle che compongono il database relazionale conservato all'interno di un server dedicato.

4.1 Tabelle e campi

In base alle informazioni progettuali definite nel capitolo 2, la struttura del database deve prevedere le tabelle riassunte di seguito.

- CREDE: questa tabella contiene, in formato criptato i dati anagrafici di ogni utente che si registrerà tramite il Web Service e si avvarrà del servizio di audit energetico. I campi in cui è suddivisa la tabella sono:
 - *user_id*, funge da chiave primaria della tabella e rappresenterà il codice progressivo di memorizzazione degli utenti.
 - *user_name*, funge da chiave primaria della tabella e rappresenta il codice univoco identificativo dell'utente.
 - *user_sens_data_nm*, dati anagrafici dell'utente (ad esempio *user_address*, per memorizzare la geolocalizzazione dell'utente) in cui *m* e *n* sono i riferimenti rispettivamente al numero di sottosezione e al numero di domanda dell'audit.
- DIN: questa tabella contiene i dati energetici inseriti dall'utente che verranno utilizzati per l'elaborazione dei risultati di output. I campi in cui è suddivisa la tabella sono:
 - *auditIn_id*, è la chiave primaria e rappresenta il codice progressivo identificativo dell'audit preso in carico.
 - *auditUser_id*, per memorizzare il codice identificativo dell'utente associato all'audit.
 - *check_level*, per segnalare il livello di progresso dell'audit. Questo campo risulta fondamentale per coordinare sia l'attivazione del demone informatico sia il processo di inserimento dati tramite interfaccia.
 - *in_11 ... in_mn*, per memorizzare i dati inseriti dall'utente. *m* e *n* sono i riferimenti rispettivamente al numero di sottosezione e al numero di domanda dell'audit.
- DOUT: questa tabella contiene i risultati di output ottenuti a valle dell'elaborazione. I campi in cui è suddivisa la tabella sono:
 - *auditOut_id*, è la chiave primaria e rappresenterà il codice univoco identificativo dell'audit di cui sono stati elaborati gli output.
 - *auditUser_id*, per memorizzare il codice identificativo dell'utente associato all'audit.
 - *out_11 ... out_xy*, per memorizzare i dati ottenuti a valle dell'elaborazione. *x* e *y* sono i riferimenti rispettivamente al numero di sottosezione e del numero di output calcolato.

Tra le diverse tabelle si possono evidenziare le seguenti relazioni e corrispondenze:

- Relazione *uno a uno* tra:
 - il campo *user_name* della tabella CREDE e il campo *auditUser_id* della tabella DIN.
 - il campo *user_name* della tabella CREDE e il campo *auditUser_id* della tabella DOUT.
 - il campo *auditUser_id* della tabella DIN e il campo *auditUser_id* della tabella DOUT.
- Relazione *uno a uno* tra:
 - il campo *user_id* della tabella CREDE e il campo *auditIn_id* della tabella DIN.
 - il campo *user_id* della tabella CREDE e il campo *auditOut_id* della tabella DOUT.
 - il campo *auditIn_id* della tabella DIN e il campo *auditOut_id* della tabella DOUT.

Nella Fig.7 è riportato il diagramma EER in cui sono rappresentate le correlazioni tra le diverse tabelle interessate dal processo che si vuole realizzare.

La scelta di adottare una soluzione che prevede l'implementazione di tre diverse tabelle è stata dettata dalla necessità di ridurre al minimo le latenze dovute al sovraccarico della risorsa condivisa, ossia il database. Infatti, gli accessi contemporanei da parte del *thread* informatico e delle routine associate al Web Service ad un'unica tabella porterebbero senza dubbio ad un sensibile aumento del tempo computazionale impiegato nelle fasi di lettura e scrittura dei dati.

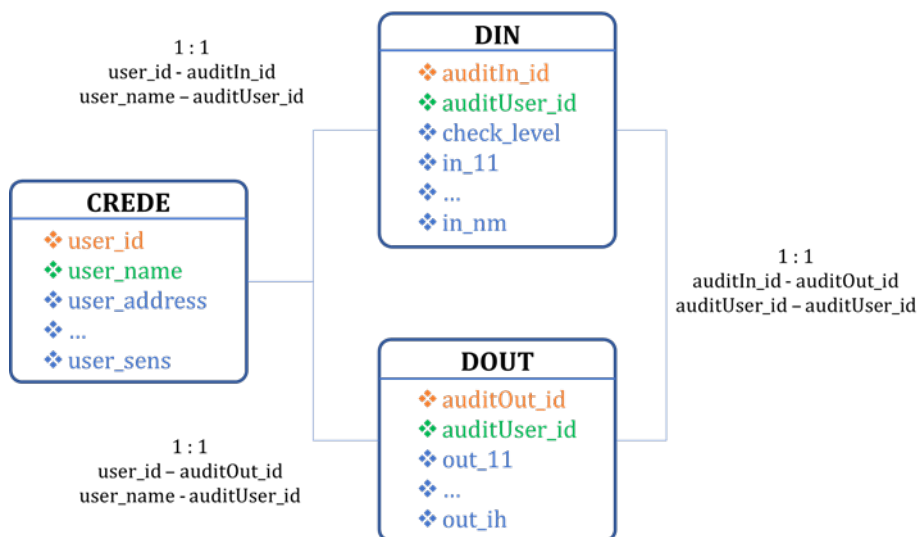


Fig. 7 - Diagramma EER del database progettato

4.2 Tabelle e campi

Con il continuo progresso tecnologico, la diffusione degli attacchi informatici è in costante aumento. Per questo motivo, grazie al loro significativo impatto sulla privacy degli utenti e delle informazioni, le funzioni di sicurezza e di crittografia dei dati e dei messaggi scambiati all'interno delle comunicazioni tra dispositivi hanno ormai una larga diffusione. Quindi, per evitare che i dati siano esposti ai cosiddetti *cyber attacks* provenienti da terze parti, questi vengono crittografati, ossia riorganizzati in sequenze di numeri apparentemente senza senso, denominate *testo cifrato*. Per accedere alle informazioni, è necessario avere una *chiave di decriptazione* che consente di decifrare il messaggio e riportarlo al suo formato originale.

I metodi di crittografia che sono attualmente i più sicuri e più diffusi, vengono utilizzati da banche e note piattaforme social, prevedono una chiave a 256 bit o a 128 bit. Indicativamente, infatti, più è lunga la chiave di decriptazione e più è difficile scoprirla.

Le principali strategie di criptazione dati sono:

- Crittografia a *chiave simmetrica*: viene utilizzata la stessa chiave per codificare e decodificare le informazioni. La chiave segreta è condivisa solo con il mittente e il destinatario (Fig.8).

- Crittografia a *chiave asimmetrica*: vengono utilizzate chiavi diverse per la codifica e la decodifica dei dati. Esiste una chiave privata, conosciuta solo dal particolare processo di gestione delle comunicazioni, e una chiave pubblica, condivisa da tutti gli utenti sulla rete. I dati codificati tramite una particolare chiave pubblica possono essere correttamente decodificati solo dalla corrispondente chiave privata.

Per gli scopi di questa attività, si è deciso di proteggere le informazioni anagrafiche riguardanti gli utenti ricorrendo ad uno dei più comuni metodi di crittografia dei dati, ossia il metodo *Advanced Encryption Standard* [9] (AES). La scelta è ricaduta su questo algoritmo in quanto risulta essere uno strumento di sicurezza estremamente avanzato e potente. Utilizzato dalle maggiori Aziende e Organizzazioni, possiamo citare tra tutte Apple e Microsoft, non è stato trovato, ad oggi, un modo per violarlo. Il rilascio di AES, avvenuto nel 1999 a cura del National Institute of Standard and Technology (NIST), fu necessario quando divenne evidente che lo standard DES, utilizzato fino a quel momento, era diventato vulnerabile agli attacchi *brute force*.

Il DES era l'algoritmo di crittografia utilizzato per venti anni dal governo degli USA finché *distributed.net* in collaborazione con la *Electronic Frontier Foundation* hanno forzato e violato il DES in meno di 24 ore. Il nuovo algoritmo, l'AES, è di tipo a chiave simmetrica e memorizza le informazioni utilizzando un algoritmo di cifratura a blocchi, i quali stabiliscono l'input di testo in chiaro e l'output del testo cifrato. Ogni cifrario AES codifica e decodifica i dati in blocchi di 128 bit prevedendo dieci round di cifratura.

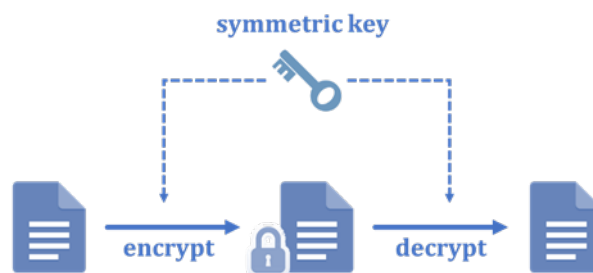


Fig. 8 - Schema di crittazione a chiave simmetrica

Inoltre, l'AES, è stato progettato per essere facilmente implementabile e integrabile in differenti contesti software. Anche MYSQL mette a disposizione una serie di funzioni che si occupano di crittografia e codifica delle informazioni utilizzate proprio per la conservazione in memoria di password, qualora il database faccia parte di un sistema di autenticazione.

I metodi di MYSQL che implementano l'algoritmo AES sono *aes_encrypt* per crittografare e *aes_decrypt* per decrittare l'informazione.

In particolare, la sintassi prevista per l'utilizzo di queste funzionalità è mostrata nelle query MYSQL riportate in Fig.9.

```
1 INSERT INTO data_table
2 VALUES ("field", AES_ENCRYPT("data","key"))
1 SELECT "field", AES_DECRYPT("field","key")
2 FROM data_table
```

Fig. 9 - Sintassi query MYSQL per la crittazione AES

In cui:

- *data_table* è la tabella del database in cui si vuole inserire il dato.
- *field* è il campo, all'interno della tabella, in si vuole inserire il dato.
- *data* è l'informazione che si vuole criptare.
- *key* è la chiave simmetrica utilizzata per la crittografia del dato.

Come già anticipato, viene applicata una cifratura simmetrica, per cui si utilizza la stessa chiave per codificare e per leggerli in chiaro. Con l'obiettivo di aggiungere un ulteriore livello di sicurezza, si deve applicare alla

chiave di criptazione una forma di crittografia a senso unico utilizzando il metodo *hash* messo a disposizione da MYSQL.

Grazie a questa funzione è possibile ricavare una stringa di lunghezza fissa e apparentemente incomprensibile, detta *codice hash*, a partire da una stringa qualsiasi, facendo in modo però che l'operazione inversa sia impossibile.

In questo modo, la chiave di criptazione non sarà leggibile in modo diretto nel caso in cui si verifichi un attacco informatico mirato a leggere le informazioni sensibili memorizzate nel database.

L'implementazione della funzione hash che si è stabilito di adottare poiché estremamente adattabile con le funzioni AES è l'MD5 [10]. Questa, infatti, calcola una *checksum* da 128 bit, a partire da una stringa qualsiasi, e restituisce un codice esadecimale di 32 caratteri. Un esempio di utilizzo della funzione è riportato di seguito.

```

1 INSERT INTO data_table
2 VALUES ("field", AES_ENCRYPT("data",UNHEX(MD5("key"))))
1 SELECT "field", AES_DECRYPT("field",UNHEX(MD5("key")))
2 FROM data_table
    
```

Fig. 10 - Sintassi query MYSQL criptazione AES con codice hash

La schematizzazione finale tramite la quale è possibile rappresentare il processo di protezione dei dati che si è progettato può essere quella riportata in Fig.11.



Fig. 11 - Schema di criptazione a chiave simmetrica e codice hash

5 Orchestrator - Il demone informatico

In relazione agli obiettivi di questa ricerca si considera di implementare l'algoritmo tramite il linguaggio di programmazione orientato agli oggetti Java [11]. Tale linguaggio è stato creato a partire da ricerche effettuate alla Stanford University agli inizi degli anni Novanta raggiungendo oggi una maturità consolidata. Uno dei principi fondamentali del linguaggio è espresso dal motto WORA (write once, run anywhere), ossia "scrivi una volta, esegui ovunque". Infatti, il codice compilato che viene eseguito su una piattaforma è in un formato chiamato *bytecode* che risulta eseguibile da una qualunque implementazione della Java Virtual Machine, un processore virtuale, e perciò non deve essere ricompilato.

Ciò permette, non solo una più agevole integrazione con il sistema di supervisione preesistente, ma anche una gestione immediata del database relazionale con cui il processo di controllo deve interagire.

Per realizzare questo tipo di controllo si sfrutta l'utilizzo di un componente Java molto importante, ossia il thread. Il linguaggio java infatti consente di creare applicazioni in grado di utilizzare la corretta logica dei processi, garantendo la condivisione in parallelo dello spazio di memoria tra i vari processi.

Dal punto di vista dell'applicazione i *thread* rappresentano una serie di processi logici che, da una parte condividono la stessa memoria della procedura che li ha creati, dall'altra concorrono con il processo principale al meccanismo di assegnazione della CPU.

Per rendere il codice scalabile e riutilizzabile si prevede l'uso di librerie Java open source:

- *mysqlConnector*, messa a disposizione da Oracle che fornisce supporto alla procedura durante le operazioni di connessione e disconnessione con il database. Permette di far dialogare MySQL con Java in modo tale da poter eseguire le query tramite uno script Java fornendo la possibilità di agire direttamente sul database. Ciò consente la lettura automatica delle tabelle in ambiente MySQL e l'esecuzione delle operazioni di CRUD senza alcuna interazione da parte dell'utente.
- *POI* [12], messa a disposizione da Apache che fornisce supporto alla procedura per leggere e scrivere file in formati Microsoft Office, come Word, PowerPoint ed Excel. In particolare, il sottocomponente che verrà utilizzato è XSSF (XML Spreadsheet Format) per la manipolazione di file con formato Open Office XML (XLSX, XLSM).

Tramite queste librerie è possibile far dialogare un database MySQL e un documento XLSM con Java. In questo modo si realizza l'esecuzione delle operazioni di CRUD sulle tabelle in ambiente MySQL e la manipolazione dei file Excel senza alcuna interazione da parte dell'utente.

Le operazioni che in ciclicamente deve effettuare il demone informatico possono essere riassunte tramite lo schema procedurale che segue:

1. Acquisizione delle informazioni di interesse dalla tabella DIN:
 - Apertura della connessione con il db.
 - Verifica dell'esistenza di un nuovo record non ancora elaborato:
 - Se non esiste, chiusura della connessione con il db e si riprende dal punto 9.
 - Altrimenti si prosegue.
 - Estrazione dei dati dal db.
 - Chiusura della connessione con il db.
2. Immissione dei dati nel documento XLSM per l'elaborazione:
 - Apertura dello stream di dati con il file .xslm.
 - Scrittura dei dati all'interno di specifiche celle del file .xslm.
 - Chiusura dello stream di dati con il file .xslm.
3. Attesa per l'elaborazione dei dati da parte del documento XLSM.
4. Acquisizione dei dati risultanti dall'elaborazione:
 - Apertura dello stream di dati con il file .xslm.
 - Lettura dei dati all'interno di specifiche celle del file .xslm.
 - Chiusura dello stream di dati con il file .xslm.

5. Immissione dei risultati dell’elaborazione nella tabella DOUT:

- Apertura della connessione con il db.
- Scrittura dei dati nel db.
- Chiusura della connessione con il db.

Le funzioni che ciclicamente effettua il demone sono riportate graficamente in Fig.12.

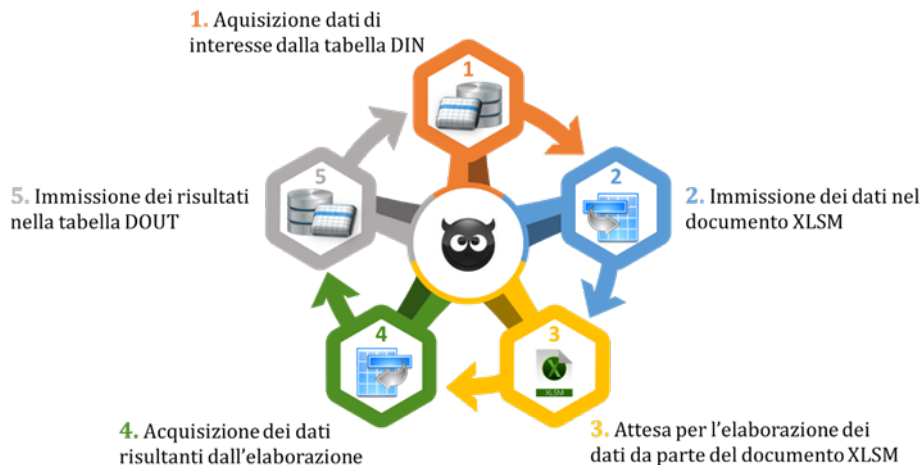


Fig. 12 - Processi gestiti e controllati dal demone informatico

Si può facilmente osservare che la procedura implementata occupa il canale di connessione solo il tempo strettamente necessario. Ciò ha un impatto positivo sull’efficienza del processo in quanto vengono limitati gli accessi multipli e le risorse database e documento XLSM non vengono sovraccaricate di richieste. Gli accessi, in scrittura e lettura, al database da parte delle routine associate al Web Service e al thread informatico si possono riassumere come segue:

- CREDE, accesso in scrittura e lettura da parte delle routine associate al Web Service.
- DIN, accesso in scrittura e lettura da parte delle routine associate al Web Service e accesso solo in lettura e scrittura da quelle associate al demone.
- DOUT, accesso in scrittura e lettura da parte delle routine associate al demone e accesso solo in lettura da parte di quelle associate al Web Service.

La relativa schematizzazione è riportata in Fig.13.

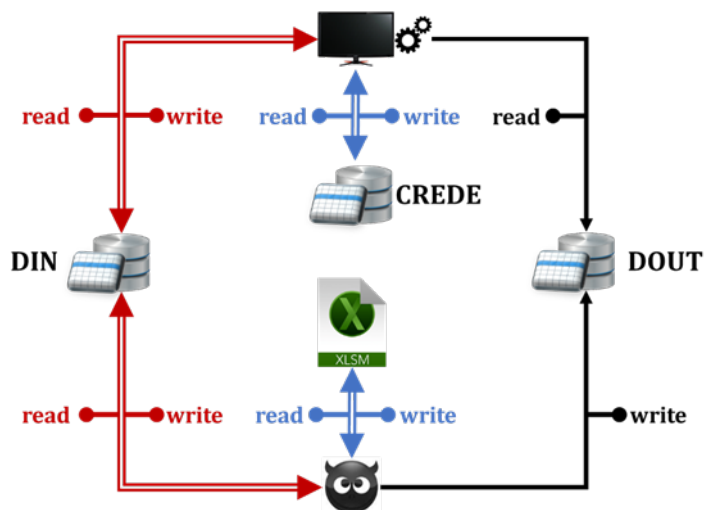


Fig. 13 - Schema accessi al database

6 Requisiti di progettazione

Lo scopo di questa sezione è di elencare e raggruppare una serie di caratteristiche che si rendono utili o necessarie per poter attuare quanto progettato. Queste caratteristiche vengono dette requisiti e possono essere principalmente i due tipi:

- Una descrizione astratta e generica di un servizio o un vincolo del sistema.
- Una specifica concreta e dettagliata di una funzionalità del sistema.

Sulla base di questi contenuti sarà possibile indirizzare meglio la successiva fase di realizzazione e implementazione di un prototipo funzionale della piattaforma off-line. Il sistema complessivo, come detto, deve includere:

- Un'interfaccia per l'utente finale con lo scopo principale di raccolta dati e visualizzazione dei risultati e suggerimenti.
- Un servizio per l'ingestione e analisi dei dati composto da una struttura di archiviazione e un elemento dedicato alla loro elaborazione.

6.1 Specifica di un requisito

I requisiti individuati saranno divisi in base all'elemento del sistema a cui si riferiscono, e potranno essere di due tipologie:

- Requisiti funzionali, che descrivono le funzionalità da fornire agli elementi del sistema.
- Requisiti non funzionali, che non descrivono direttamente delle funzionalità, ma piuttosto le caratteristiche del sistema che sono orizzontali alle funzionalità (scalabilità, riusabilità, ecc.).

Nella tabella 1 è riportato il template di esempio che si utilizzerà per la definizione delle specifiche.

ID:	Nome:	Importanza:
Descrizione:		

Tabella 1 - Template per la descrizione di un requisito

Si può osservare che il template è composto dai seguenti campi:

- ID, il codice univoco di identificazione del particolare requisito. Il codice si costruisce considerando il formato *Progetto_Classificazione_numero*, in cui:
 - Il progetto di Audit Energetico si abbrevia come *AE*;
 - I codici inerenti alla classificazione sono:
 - *GEN*, requisiti del Sistema;
 - *APP*, requisiti riguardanti l'elemento Application;
 - *STO*, requisiti riguardanti la struttura di archiviazione;
 - *ORC*, requisiti riguardanti il processo informatico per la gestione del flusso di dati tra la struttura di archiviazione e quella di elaborazione dei risultati;
 - *ENG*, requisiti riguardanti la struttura di elaborazione dei risultati.
 - Il numero progressivo per definire univocamente il requisito.

I differenti codici per identificare i requisiti sono illustrati nella tabella 2.

- Nome, il nome assegnato al requisito;
- Importanza, che descrive il livello di importanza del requisito, e può essere:
 - Richiesto (RI), ossia obbligatorio e assolutamente richiesto;
 - Raccomandato (RA), quando può esistere un ragionevole motivo per ignorarlo ma è necessario valutare attentamente le conseguenze che si possono generare;
 - Opzionale (OP), ossia un requisito consigliato ma non obbligatorio.

Per riconoscere il livello di importanza del requisito si può fare anche riferimento al verbo che viene utilizzato nella sua descrizione. Infatti, verranno utilizzati:

- *Deve*, per riferirsi ad un requisito Richiesto e quindi obbligatorio;
- *Dovrebbe*, per riferirsi ad un requisito Raccomandato;
- *Potrebbe*, per riferirsi ad un requisito Opzionale.

Classificazione	Codice categoria
Generale	AE_GEN_<num>
Applicazione	AE_APP_<num>
Storage	AE_STO_<num>
Orchestrator	AE_ORC_<num>
Engine	AE_ENG_<num>

Tabella 2 - Classificazione dei requisiti

Di seguito verranno riportati i requisiti funzionali della piattaforma, raggruppati per differenti macro-funzionalità individuate come segue:

- Applicazione per la gestione del front end e back end dell’interfaccia grafica per l’utente;
- Struttura di archiviazione dei dati, sia le informazioni in input dall’utente che i risultati di output dell’elaborazione;
- Processo informatico per la gestione del flusso di informazioni tra la struttura di archiviazione e il foglio di calcolo adibito all’elaborazione;
- Processo di elaborazione delle informazioni inserite in input dall’utente per ottenere i risultati e i suggerimenti energetici.

Verranno per prima cosa presentati i requisiti funzionali e non funzionali riguardanti l’architettura e il sistema di Audit Energetico in generale.

6.2 Requisiti della piattaforma

ID: AE_GEN_01	Nome: Main Authentication	Importanza: RI
Descrizione: Il sistema <i>deve</i> permettere l’accesso degli utenti alla piattaforma solo a seguito di autenticazione.		
ID: AE_GEN_02	Nome: User Identification	Importanza: RI
Descrizione: Tutti gli utenti del servizio di Audit <i>devono</i> essere autenticati ed individuabili tramite il codice univoco assegnatogli o dal sistema stesso o dal portale Smart Home (RdS/PTR(2019)/001 “Tecnologie per le Smart Homes”).		
ID: AE_GEN_03	Nome: Data Encryption	Importanza: RA

Descrizione: Il sistema <i>dovrebbe</i> supportare l'uso di protocolli criptati per la connessione da remoto, come HTTPS e SSH.

ID: AE_GEN_04	Nome: System Availability	Importanza: RA
Descrizione: Il sistema <i>dovrebbe</i> garantire un'alta disponibilità del servizio tradotta in bassi tempi di latenza.		

ID: AE_GEN_05	Nome: Data Integrity Standard	Importanza: RA
Descrizione: Il sistema <i>dovrebbe</i> utilizzare standard e protocolli di comunicazione che garantiscono l'integrità dell'informazione.		

ID: AE_GEN_06	Nome: Data Type	Importanza: RI
Descrizione: <i>Devono</i> essere definiti i dati, inseriti dall'utente, da considerare anagrafici, e quindi da criptare.		

ID: AE_GEN_07	Nome: Feature Filtering	Importanza: RI
Descrizione: Il sistema <i>deve</i> permettere all'utente di usufruire delle funzionalità disponibili, filtrate nell'accessibilità e visualizzazione, sulla base del progresso nella compilazione del questionario.		

ID: AE_GEN_08	Nome: Interface Usability	Importanza: RA
Descrizione: Il sistema <i>dovrebbe</i> prevedere un'interfaccia verso l'utente che sia intuitiva, immediata e di facile utilizzo per una gamma di utenti il più possibile vasta.		

ID: AE_GEN_09	Nome: Prototyping Documentation	Importanza: RA
Descrizione: Il prototipo che verrà realizzato in una fase successiva <i>dovrebbe</i> essere adeguatamente documentato.		

ID: AE_GEN_10	Nome: Prototyping Testability	Importanza: RI
Descrizione: Sul prototipo che verrà realizzato <i>devono</i> essere effettuati test di validazione inerenti ai requisiti funzionali.		

6.3 Requisiti dell'elemento Application

ID: AE_APP_01	Nome: Data Exchange	Importanza: RI
-------------------------	-------------------------------	--------------------------

Descrizione: L'interfaccia Application <i>deve</i> essere in grado di strutturare un flusso massivo di interscambio dati con la struttura di archiviazione (Storage).

ID: AE_APP_02	Nome: User Authentication	Importanza: RI
Descrizione: L'interfaccia Application <i>deve</i> prevedere un processo per l'autenticazione dell'utente ad ogni accesso al portale di Audit.		

ID: AE_APP_03	Nome: Data Acquisition	Importanza: RI
Descrizione: L'interfaccia Application <i>deve</i> acquisire le informazioni inserite dall'utente tramite semplici form, testuali o a opzione multipla.		

ID: AE_APP_04	Nome: Compilation Assistance	Importanza: RA
Descrizione: L'interfaccia Application <i>dovrebbe</i> presentare all'utente informazioni testuali utili che possano guidarlo nell'inserimento dei dati, prima e durante l'esecuzione del questionario.		

ID: AE_APP_05	Nome: Feature Visibility	Importanza: RI
Descrizione: L'interfaccia Application <i>deve</i> rendere disponibili le funzionalità all'utente sulla base delle limitazioni imposte dal progresso di compilazione dell'utente stesso.		

ID: AE_APP_06	Nome: Report Presentation	Importanza: RI
Descrizione: L'interfaccia Application <i>deve</i> presentare all'utente finale i risultati dell'elaborazione utilizzando rappresentazioni grafiche, diagrammi, numeriche e testuali.		

ID: AE_APP_07	Nome: Report Download	Importanza: OP
Descrizione: L'interfaccia Application <i>potrebbe</i> permettere all'utente di scaricare sul proprio dispositivo i risultati dell'Audit (formato PDF, DOC o TXT).		

6.4 Requisiti dell'elemento Storage

ID: AE_STO_01	Nome: Storage Protection	Importanza: RI
Descrizione: L'accesso in scrittura e lettura alla struttura di archiviazione Storage <i>deve</i> essere protetto e consentito solo tramite password.		

ID: AE_STO_02	Nome: Credential Storage	Importanza: RI
Descrizione: La struttura di archiviazione Storage <i>deve</i> prevedere una tabella dedicata esclusivamente alla memorizzazione delle credenziali di accesso.		

ID: AE_STO_03	Nome: Encrypted Data Storage	Importanza: RI
Descrizione: La struttura di archiviazione Storage <i>deve</i> prevedere una tabella dedicata alla memorizzazione, in formato criptato, delle informazioni anagrafiche dell'utente.		

ID: AE_STO_04	Nome: Storage Architecture	Importanza: OP
Descrizione: La struttura di archiviazione Storage <i>potrebbe</i> prevedere una sola tabella dedicata alla memorizzazione dei dati inseriti dall'utente e dei risultati ottenuti a seguito dell'elaborazione.		

ID: AE_STO_05	Nome: Storage Architecture	Importanza: RA
Descrizione: La struttura di archiviazione Storage <i>dovrebbe</i> essere realizzata utilizzando un database relazionale SQL, poiché compliant con l'ecosistema informatico in cui verrà immerso.		

6.5 Requisiti dell'elemento Orchestrator

ID: AE_ORC_01	Nome: Stored Data Exchange	Importanza: RI
Descrizione: Il processo informatico Orchestrator <i>deve</i> essere in grado di strutturare un flusso massivo di interscambio dati con la struttura di archiviazione (Storage).		

ID: AE_ORC_02	Nome: Elaborated Data Exchange	Importanza: RI
Descrizione: Il processo informatico Orchestrator <i>deve</i> essere in grado di strutturare un flusso massivo di interscambio dati con la struttura di elaborazione dati (Engine).		

ID: AE_ORC_03	Nome: Data Recognition	Importanza: RI
-------------------------	----------------------------------	--------------------------

Descrizione: Il processo informatico Orchestrator <i>deve</i> essere in grado di riconoscere la presenza di un nuovo set di dati pronto per le elaborazioni.

ID: AE_ORC_04	Nome: Data Extraction	Importanza: RI
Descrizione: Il processo informatico Orchestrator <i>deve</i> processare un set di dati da elaborare alla volta.		

ID: AE_ORC_05	Nome: Implementation	Importanza: RA
Descrizione: Il processo informatico Orchestrator <i>dovrebbe</i> essere scritto nel linguaggio di programmazione Java, poichè compliant con l’ecosistema informatico in cui verrà immerso.		

6.6 Requisiti dell’elemento Engine

ID: AE_ENG_01	Nome: Engine Accessibility	Importanza: RI
Descrizione: La struttura di elaborazione Engine <i>deve</i> permettere un facile accesso in lettura e scrittura da parte del processo informatico Orchestrator.		

ID: AE_ENG_02	Nome: Stored Data Exchange	Importanza: RI
Descrizione: La struttura di elaborazione Engine <i>deve</i> fare riferimento a quanto prodotto nel Report RdS/PAR2016/009.		

7 Conclusioni e risultati

In questo ambito progettuale le attività su cui si sono concentrate le attenzioni risultano essere quattro e sono riassumibili come segue.

Per prima cosa è stato analizzato il foglio di calcolo XLSM, responsabile dell'elaborazione dei dati, per renderlo conforme ai requisiti del progetto. Sono stati evidenziati gli output di interesse e, di conseguenza, identificati le funzioni e gli input coinvolti. Ciò ha permesso di eliminare le latenze snellendo di gran lunga la procedura di calcolo.

Quindi, definite la numerosità e la tipologia di dati necessari all'elaborazione, si è passati alla progettazione del front-end della web application. Questa ha il ruolo di interfaccia con l'utente ed accede al database per la memorizzazione dei dati. Perciò si sono definite:

- Pagine web dinamiche caratterizzate da una serie di form di inserimento dati semplici e intuitive che guideranno l'utente nel processo di inserimento delle informazioni necessarie per il calcolo dell'audit energetico;
- Pagine web dinamiche tramite la quale l'utente potrà visualizzare i risultati di output ottenuti a valle dell'elaborazione del file XLSM. Visualizzazione che potrà essere sia testuale che grafica tramite quadri sinottici e diagrammi. Si vuole lasciare l'opportunità all'utente di poter scaricare un file pdf nel quale sono sia riassunte le informazioni inserite che riportati gli output ottenuti.

A valle della definizione del foglio di calcolo ci si è soffermati sulla parte di back-end che si occupa della corretta scrittura e lettura dei dati sia in memoria che nel file XLSM. Si sono quindi definite le funzionalità del processo informatico che ciclicamente è in grado di:

- Prelevare i dati di interesse dal database di cui sopra.
- Inserire correttamente i dati nel file XLSM per l'elaborazione degli output.
- Leggere correttamente i dati di output prodotti dal foglio XLSM e procedere alla loro scrittura sul database.

E' possibile riassumere il processo come segue:

- I dati vengono inseriti dall'utente tramite delle form di acquisizione disponibili dall'interfaccia del Web Service.
- Il Web Service memorizza i dati in due diverse tabelle all'interno di un database:
 - Nella prima CREDE vengono salvati, in formato criptato, i dati personali dell'utente.
 - Nella seconda DIN vengono salvati, in formato non criptato i dati inseriti dall'utente.
- Il demone ciclicamente:
 - Acquisisce dalla tabella DIN i record ancora non elaborati.
 - Scrive all'interno delle celle del file xml i dati inseriti dall'utente.
 - Attende il tempo di elaborazione del file XLSM.
 - Legge all'interno delle celle del file xml i risultati ottenuti.
 - Scrive nella tabella DOUT del database i dati di output.
- Il Web Service interroga la tabella DOUT in base alle richieste degli utenti e fornisce in output i risultati ottenuti.

Schematizzando, in Fig.14 si ha la rappresentazione procedurale della piattaforma proposta.

Alla luce di quanto visto detto, l'architettura generale della piattaforma di supporto per il processo di Audit Energetico si compone dei seguenti building blocks:

- Un portale web (Application) dedicato agli utenti finalizzato a fornire loro informazioni di feedback a partire dai dati da loro stessi immessi.
- Una struttura di archiviazione (Storage) in grado di memorizzare i dati in modo permanente e che permetta un agile e veloce accesso ad essi.
- Un servizio (Engine) per l'elaborazione dei dati inseriti dall'utente.
- Un servizio (Orchestrator) per la gestione dello scambio dei dati tra la struttura di archiviazione e il servizio di elaborazione dell'Audit.

In Fig.15 è riportata una rappresentazione semplificata del sistema, con evidenziati anche i collegamenti tra i vari componenti sopradescritti.

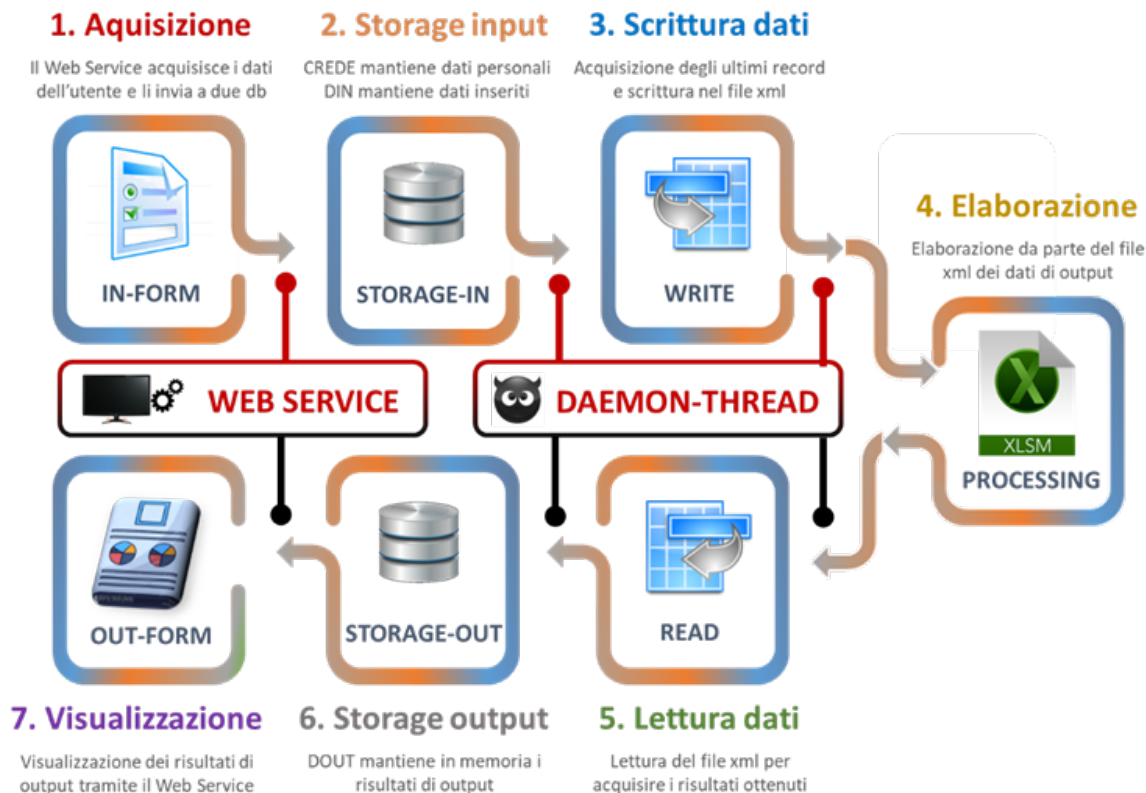


Fig. 14 - Schema procedurale della piattaforma informatica

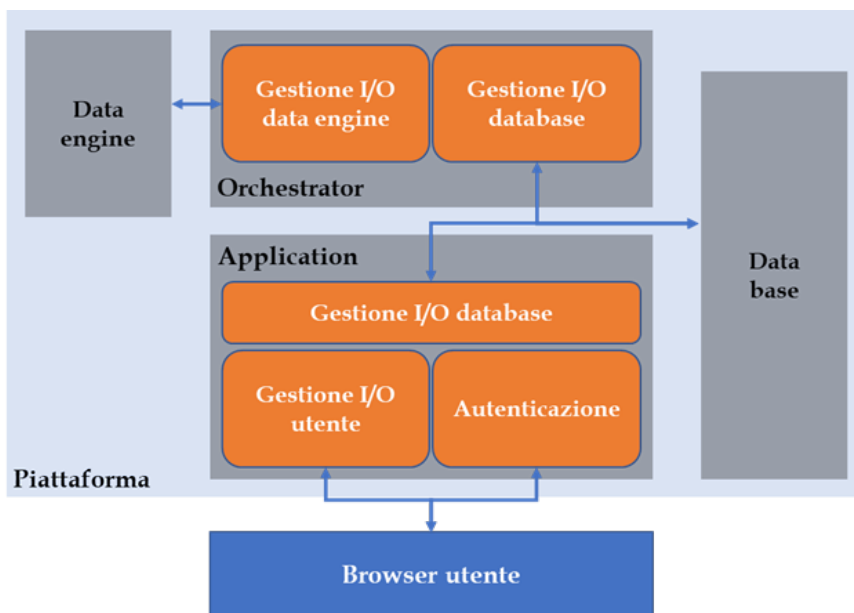


Fig. 15 - Modello logico del sistema ICT di supporto al processo di Audit Energetico

La piattaforma così progettata, benché abbia una struttura complessa e articolata, risulta essere disaccoppiata, presentando una spiccata modularità. Questa caratteristica assicura una buona portabilità e riusabilità dello strumento. Infatti, si può notare che, se correttamente trattate, le operazioni di inserimento e sostituzione di uno degli elementi della piattaforma non pregiudica la consistenza della stessa e rende le operazioni sopracitate poco invasive.

Nel presente elaborato di progettazione si sono descritte nel dettaglio quelle che sono le prime scelte tecnologiche ed architettoniche propedeutiche alla successiva fase di realizzazione della piattaforma ICT. Sono anche presenti i primi tentativi di *make up* delle interfacce grafiche per indirizzare al meglio la loro implementazione.

Per quanto questo documento sia da considerare in forma definitiva, ci si aspetta che durante la fase di implementazione del primo prototipo off-line si facciano delle revisioni per incorporare aspetti che richiedono chiarimenti o che non sono stati valutati in questa fase.

Riferimenti

- [1] "Professional Excel Development: The Definitive Guide to Developing ... - Stephen Bullen, Rob Bovey, John Green - Google Libri." [Online]. Available: https://books.google.it/books?hl=it&lr=&id=L4BtHsMe0EwC&oi=fnd&pg=PR7&dq=microsoft+excel+definitive+guide&ots=XYQ_yn4sFT&sig=8lJ4fgBDwJNR0n4r0DRCw607vGk&redir_esc=y#v=onepage&q=microsoft excel definitive guide&f=false. [Accessed: 02-Jan-2020].
- [2] "Mastering Microsoft VBA - Guy Hart-Davis - Google Libri." [Online]. Available: https://books.google.it/books?hl=it&lr=&id=n-2NtSJNHBYC&oi=fnd&pg=PR3&dq=microsoft+VBA&ots=GJMnInL93v&sig=Tq1Y7r8BimFgpLj1g1kxUaYKXFI&redir_esc=y#v=onepage&q=microsoft VBA&f=false. [Accessed: 02-Jan-2020].
- [3] C. Musciano and B. Kennedy, *HTML, the definitive guide*. O'Reilly & Associates, 1996.
- [4] "Extending Bootstrap - Christoffer Niska - Google Libri." [Online]. Available: https://books.google.it/books?hl=it&lr=&id=x9ciAwAAQBAJ&oi=fnd&pg=PT7&dq=bootstrap+css&ots=GLtreHNhpP&sig=qNJSvc4OW99f90xgqLo8L7uU-AU&redir_esc=y#v=onepage&q=bootstrap css&f=false. [Accessed: 02-Jan-2020].
- [5] "JavaScript: The Definitive Guide - David Flanagan - Google Libri." [Online]. Available: https://books.google.it/books?hl=it&lr=&id=k0CbAgAAQBAJ&oi=fnd&pg=PT6&dq=javascript&ots=O3szeolBtR&sig=Z8aUcFibNe3ZO31R8BuaJ1LyFdo&redir_esc=y#v=onepage&q=javascript&f=false. [Accessed: 02-Jan-2020].
- [6] L. E. Ullman, *PHP For The World Wide Web*.
- [7] J. Melton, "Database Language SQL," in *Handbook on Architectures of Information Systems*, Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 105–132.
- [8] P. DuBois, *MySQL*. New Riders, 1999.
- [9] J. Nechvatal *et al.*, "Report on the development of the Advanced Encryption Standard (AES)," *J. Res. Natl. Inst. Stand. Technol.*, vol. 106, no. 3, pp. 511–577, 2001.
- [10] R. Rivest, "Network Working Group Request for Comments: 1321 The MD5 Message-Digest Algorithm," 1992.
- [11] D. Reilly, "Inside Java: The Java Programming Language," *Inside Java*, 2006. [Online]. Available: <https://dl.acm.org/citation.cfm?id=1051069>. [Accessed: 19-Dec-2018].
- [12] K. L. Bawankule and N. B. Raut, "Design and implementation of massive MYSQL data intelligent export system to excel by using Apache-POI libraries," Ver. IV.