



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

M



S

E



Ricerca di Sistema elettrico

Smart energy in sistemi pubblici: analisi di affidabilità e qualificazione dei dati per ridurre le incertezze di sistema

Fabio Leccese, Mariagrazia Leccisi



Report RdS/PTR2019/006

SMART ENERGY IN SISTEMI PUBBLICI: ANALISI DI AFFIDABILITÀ E QUALIFICAZIONE DEI DATI PER RIDURRE LE INCERTEZZE DI SISTEMA

Fabio Leccese, Mariagrazia Leccisi (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: 15 - Analisi affidabilistica di Smart Building e supporto al consolidamento della sezione front-end del PELL

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 *"Smart Energy in Sistemi Pubblici: analisi di affidabilità e qualificazione dei dati per ridurre le incertezze di sistema"*

Responsabile scientifico ENEA: Francesco Pieroni

Responsabile scientifico Università Roma Tre: Prof.re Fabio Leccese

Indice

SOMMARIO	4
1 INTRODUZIONE	5
2 IL PROGETTO PELL	6
2.1 IL PROGETTO PELL – FRAMEWORK UTILIZZATO	7
2.2 STRUTTURA DELL'APPLICAZIONE: HOME PAGE	14
2.3 STRUTTURA DELL'APPLICAZIONE: PAGINA ILLUMINAZIONE PUBBLICA	25
2.3.1 <i>Utente anonimo –form login</i>	29
2.3.2 <i>Utente autenticato – menu laterale</i>	32
2.4 STRUTTURA DELL'APPLICAZIONE: AMMINISTRAZIONE	34
2.4.1 <i>Gruppi</i>	38
2.5 STRUTTURA DELL'APPLICAZIONE: LA SCHEDA CENSIMENTO	43
2.6 STRUTTURA DELL'APPLICAZIONE: VALUTAZIONI ECONOMICO-FINANZIARIE	71
2.6.1 <i>Sezione investimenti</i>	72
2.6.2 <i>Sezione gestione impianto</i>	72
2.6.3 <i>Nuova simulazione</i>	73
2.7 STRUTTURA DELL'APPLICAZIONE: PRESTAZIONI ILLUMINOTECNICHE	75
2.8 STRUTTURA DELL'APPLICAZIONE: PELL APPLICATION	77
2.8.1 <i>PELL Application</i>	77
2.8.2 <i>Archiviazione dei dati</i>	78
2.8.3 <i>Integrazione Hadoop – Frontend</i>	81
3 CONCLUSIONI	83
4 APPENDICE A.....	84
5 RIFERIMENTI BIBLIOGRAFICI	86
6 ABBREVIAZIONI ED ACRONIMI	87
7 APPENDICE: BREVE CURRICULUM SCIENTIFICO DEL GRUPPO DI LAVORO IMPEGNATO NELL'ATTIVITÀ.....	88

Sommario

L'attività svolta dall'unità dell'Università degli Studi "Roma Tre" nell'ambito del progetto "Smart Energy in Sistemi Pubblici: analisi di affidabilità e qualificazione dei dati per ridurre le incertezze di sistema", è tesa alla gestione e presa in carico del sistema PELL - Public Energy Living Lab con particolare riferimento alla parte di front-end.

Lo scopo del sistema è quello di avere una piattaforma in grado di censire, gestire e monitorare gli impianti di illuminazione pubblica di tutta Italia al fine di fornire utili informazioni circa un possibile risparmio energetico ed economico da parte delle amministrazioni comunali, fornendo loro la possibilità di migliorare le strutture urbane che gestiscono migliorando i benefici per tutti i cittadini e per l'ambiente.

In particolare il lavoro di cui ci stiamo occupando prevede lo studio teso al miglioramento del sistema in termini di fruibilità da parte degli operatori, intervenendo sia aggiornando le schede censimento e le sezioni ad esse legate strettamente, sia valutando e risolvendo eventuali anomalie rilevate durante l'uso, sia cercando di rendere l'interfaccia sempre più agevole agli operatori, fornendo loro vari strumenti utili.

Per fare questo è stato necessario un approfondito studio dell'applicazione dal punto di vista informatico, anche attraverso l'utilizzo di appositi "use case" per verificare le funzionalità del sistema e rilevare eventuali problematiche e/o criticità.

Il lavoro che stiamo effettuando al momento che integra i dati presenti sulla scheda censimento con i dati di consumo provenienti dagli smart meters, pone le basi per il lavoro successivo che ha il fine di acquisire una grande quantità di informazioni e permettere la valutazione dei consumi degli impianti di illuminazione pubblica al fine di migliorarne le prestazioni in termini economici ed energetici.

1 Introduzione

Questo documento descrive le attività inerenti la presa in carico e lo studio di una apposita piattaforma per la raccolta, gestione, organizzazione e valutazione di dati strategici statici e dinamici relativi alle infrastrutture energivore urbane chiamata PELL - Public Energy Living Lab, con particolare riferimento alla parte di front-end, per la quale è stata effettuata un'attività di valutazione di nuove funzionalità ed il testing delle funzionalità già presenti al fine di migliorarne la fruibilità da parte degli utenti.

La gran parte dell'attività lavorativa è stata volta:

- Allo studio del framework utilizzato per sviluppare l'applicazione;
- All'analisi dell'applicazione da un punto di vista del codice e del database;
- Ad aumentare la confidenza dell'operatore verso l'utilizzo del sistema cercando di facilitare l'utente all'inserimento consapevole e corretto dei dati attraverso valutazioni di miglioramento della leggibilità di schede di raccolta dati o della visualizzazione di grafici di monitoraggio accessibili dal portale; in particolare questo punto è importante in quanto l'incertezza finale con la quale si produrranno i dati sintetici prestazionali dell'impianto è fortemente dipendente dalla confidenza con cui l'utente inserisce correttamente e consapevolmente i dati
- Alla verifica e risoluzione di anomalie di sistema.

A questo scopo l'applicazione è stata oggetto di studio approfondito, una sorta di "reverse engineering" per conoscere, da un punto di vista tecnico, sia l'applicazione stessa che il suo database, al fine di poter operare autonomamente valutando la possibilità di migliorie ed efficientamento.

Infatti l'applicazione è in continua evoluzione secondo le necessità degli utenti finali e l'aggiornamento delle tecnologie, pertanto si è reso necessario un approccio di "quality engineering" per cercare di far emergere situazioni dell'applicazione che potevano essere migliorate sia a livello di use case applicativi, che a livello di codice vero e proprio.

2 Il progetto PELL

Il progetto PELL è stato avviato nel 2014 da ENEA, su finanziamento del MISE (Ministero dello Sviluppo Economico), con l'obiettivo di raccogliere, gestire, organizzare e valutare i dati strategici statici e dinamici relativi alle infrastrutture energivore urbane (pubblica illuminazione, edifici pubblici) al fine di fornire una valutazione costante e dinamica delle loro prestazioni funzionali, energetiche ed economiche per migliorarne la gestione e permettere l'avvio di processi di riorganizzazione urbana in chiave smart city.

Il progetto ha impegnato molti enti pubblici e privati, tra cui CONSIP, l'Acquirente unico, associazioni di categoria, ma anche imprese internazionali del settore dell'energia come ENEL.

Il PELL è stato sviluppato quale strumento operativo del progetto Lumière [1], progetto framework per lo sviluppo di un modello di management efficiente ed efficace della pubblica illuminazione (<http://progettolumiere.enea.it/>).

L'illuminazione pubblica è un punto fondamentale dello sviluppo di una entità urbana, ed è anche un argomento che si presta a migliorie ed innovazioni tecnologiche, infatti è un ambito che spesso, per mancanza di risorse da parte delle amministrazioni, inefficienza energetica degli impianti, scarse competenze gestionali da parte di chi dovrebbe gestirli, viene relegato in secondo piano; invece, nel mercato sono attualmente presenti delle tecnologie che possono sicuramente migliorare questo settore, rendendo le città realmente "smart" [2].

È molto importante sviluppare il concetto di "illuminare"

- dove serve
- come serve
- quanto serve
- quando serve

sia dal punto di vista di risparmio di risorse che dal punto di vista di una vera e propria riqualificazione tecnologica, economica, sociale ed urbanistica della città.

Il progetto è destinato ad una grande varietà di persone, a partire dall'amministratore comunale per arrivare al libero cittadino, ai gestori di rete e agli operatori, perché la riqualificazione urbana è un qualcosa che impatta su tutta la comunità in termini di risparmio di risorse, ma anche di incremento di nuovi servizi e di miglioramento di servizi già esistenti.

In particolare, i servizi che offre il sistema sono pensati per le seguenti categorie di utenti:

- Amministrazioni locali e centrali
- Gestori
- Operatori settoriali
- Operatori comparto bancario
- Governance
- Cittadino.

E la loro fruizione è condizionata dalla registrazione dell'utente al Portale. I servizi disponibili variano a seconda della categoria di utente e dal livello di registrazione per garantire a ciascun utente ciò di cui ha più bisogno. È prevista anche una integrazione con la piattaforma WebGis da cui è possibile accedere ai servizi offerti dal PELL.

Il PELL può essere utilizzato da varie tipologie di utenti, ad esempio la Pubblica Amministrazione dovrebbe essere la maggior fruitrice di questo strumento; infatti, attraverso il monitoraggio degli impianti e usando gli

algoritmi di valutazione di una eventuale riqualificazione, il PELL può studiare ulteriori scenari di ricerca e sviluppo per rendere più sicuri ed efficienti gli impianti. Tale efficienza potrà avvenire, ad esempio,

- utilizzando nuove tecnologie che siano più efficienti (molti impianti sono datati) come ad esempio l'uso di lampade a LED piuttosto che a incandescenza o ai vapori di mercurio
- ripensando la gestione dell'impianto nella sua globalità anche attraverso nuove tecnologie che permettano di utilizzare correttamente l'energia in modalità "on demand", cioè dove serve e quando serve

Al fine di raggiungere questi scopi, è importante per prima cosa conoscere gli impianti attualmente presenti e funzionanti nel territorio, sia da un punto di vista tecnico che da un punto di vista di consumi energetici; infatti, gli impianti di illuminazione pubblica sono considerati energivori, cioè consumatori di una grande quantità di energia elettrica.

A tal fine l'adesione al PELL implica che la Pubblica Amministrazione realizzi il censimento degli impianti secondo un data model PELL ben definito (<https://geodati.gov.it>)[3], e l'adozione di smart meter su tutti i Quadri Elettrici per consentire il monitoraggio continuo e costante dei consumi.

L'utilizzo di una piattaforma dedicata permette l'informatizzazione dei dati tecnici degli impianti e dei consumi di ciascuno di essi; questo è molto importante, perché una volta che i dati sono salvati su un database, è sempre possibile accedervi e, attraverso processi e strumenti più o meno complicati, effettuare delle valutazioni in modo molto semplice, rapido, e trasparente.

Ad esempio, avere memorizzati su un database i consumi di un impianto permette di valutare gli stessi secondo algoritmi semplici o complessi (es. reti neurali), dando visibilità anche grafica dell'andamento dei consumi e dei costi nel passato, nel presente e anche, attraverso algoritmi previsionali, nel futuro.

La piattaforma permette:

- il censimento degli impianti e dei pod;
- l'analisi (benchmarking, diagnostica, prestazioni) delle grandezze dei singoli pod;
- il calcolo illuminotecnico sulla base dell'algoritmo OXYTECH;
- la valutazione economico-finanziaria di un singolo impianto;
- di gestire dati statici e dinamici con l'obiettivo di fornire una valutazione costante e dinamica tanto dello stato di fatto dell'impianto quanto delle sue prestazioni funzionali, energetiche ed economiche.

2.1 Il progetto PELL – Framework utilizzato

Nella Figura 1 viene visualizzato lo schema relativo al funzionamento dell'infrastruttura; il suo sviluppo è pensato attraverso due fasi distinte:

- una fase statica: relativa al censimento delle infrastrutture (PIATTAFORMA ICT LIVING LAB) al fine di conoscerne lo stato attuale effettivo e, eventualmente, valutarne la riqualificazione;
- una fase dinamica: relativa all'acquisizione ed al monitoraggio dinamico dell'infrastruttura e dei suoi consumi, attraverso vari strumenti di analisi e diagnostica.

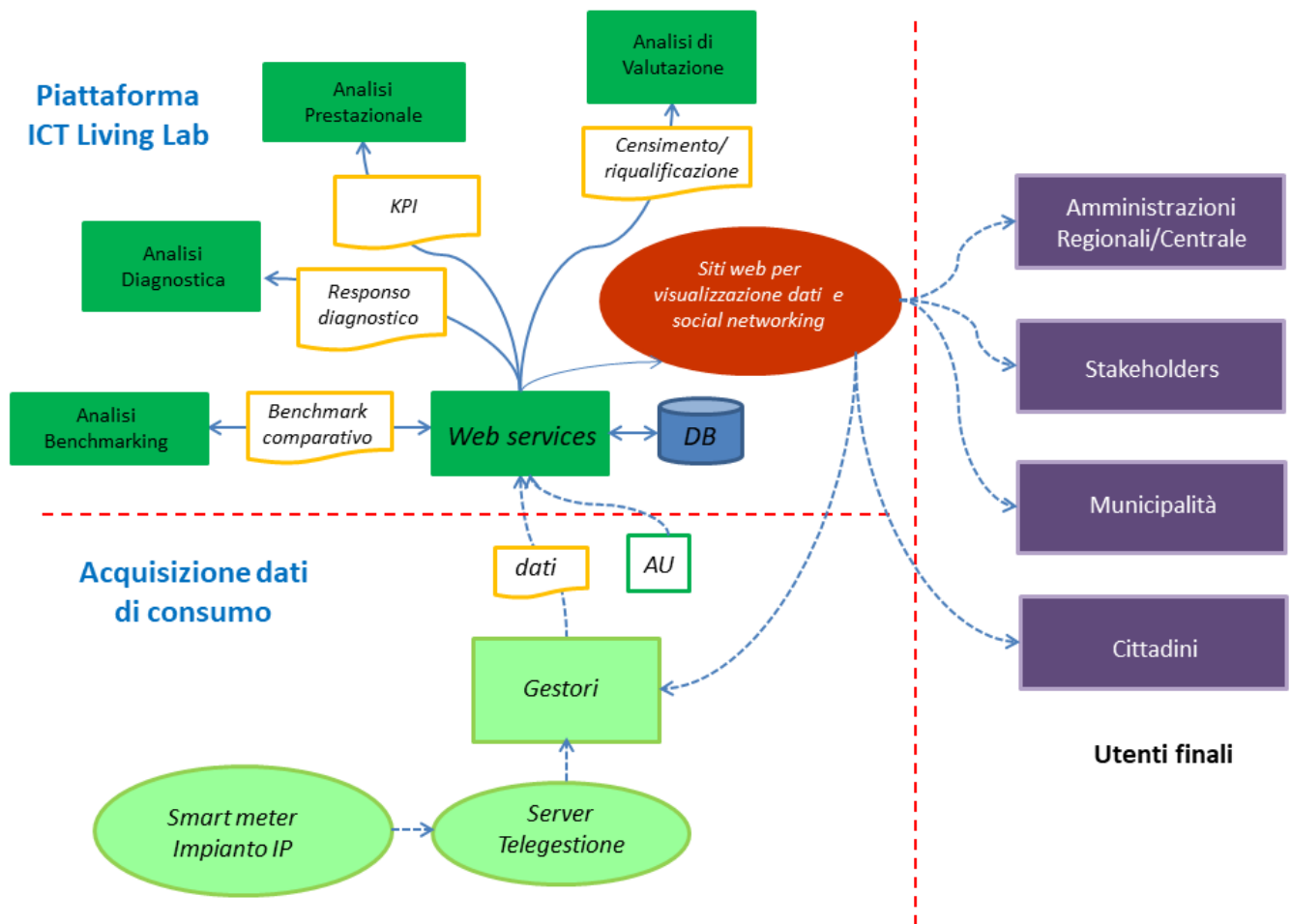


Figura 1- piattaforma PELL

Nei paragrafi successivi descriveremo le varie sezioni dell'applicazione da un punto di vista dell'utente e dello sviluppatore, evidenziando lo studio approfondito che è stato fatto dell'applicazione.

Per lo sviluppo del PELL è stato utilizzato il framework php open source CodeIgniter, per lo sviluppo di applicazioni web in linguaggio PHP.

È un sistema basato sul paradigma MVC (model-view controller), che consiste nella divisione di un progetto in 3 sezioni distinte e separate:

- Model
- View
- Controller

Il model è la parte di applicazione che descrive la struttura dei dati; in esso sono presenti:

- Tutti i dati che permettono la definizione delle tabelle del database;
- Tutte le chiamate da e verso il database, comprese le funzioni base come INSERT, UPDATE, SELECT, ma anche viste, stored procedures ecc.

La parte relativa alla view comprende tutto ciò che riguarda l'interfaccia di visualizzazione che mostra i dati agli utenti, cioè le GUI. Queste possono essere di vario tipo, può essere una pagina web intera o può essere

anche una porzione di essa, ad esempio un header o un footer, o anche una pagina RSS. Contiene tutto il codice legato alle pagine, quindi html, jsp ecc....

Il controller si occupa di processare le varie richieste http e richiamare la view, passando i dati caricati dal model; in pratica è uno strato intermedio tra il model e il view che gestisce le richieste vere e proprie.

L'architettura del framework è descritta di seguito in Figura 2:

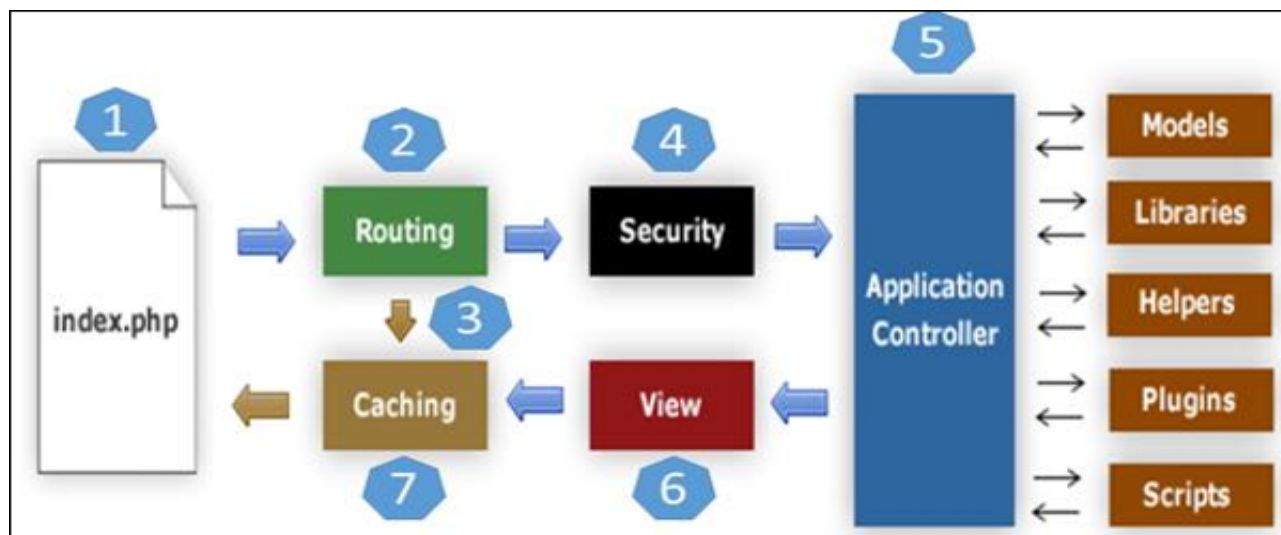


Figura 2 – Schema generale di CodeIgniter

Quando viene effettuata la richiesta da parte del browser, viene invocato il file index.php, che inizializza tutte le risorse e le librerie che necessitano a CodeIgniter per poter funzionare; esso instrada la chiamata verso i controlli di sicurezza, oppure restituisce direttamente la risposta all'utente, se la pagina richiesta è già in cache.

Nel caso la pagina non fosse in cache, la richiesta viene instradata verso il controller che si occupa di caricare tutto ciò che serve al sistema in base alla richiesta effettuata e, in ultimo, di passare i dati alla view che viene caricata dal browser per la visualizzazione e, eventualmente, salvata in cache.

Il framework è contenuto in un file.zip (scaricabile dal sito <https://codeigniter.com/>) che, decompresso, contiene:

- Un file index.php
- Directory application
- Directory system
- Directory user_guide
- Directory test.

Di seguito illustreremo le cartelle e i files in esse contenuti, con particolare riferimento alle directories "application" e "system".

La directory **system** contiene le cartelle e i files di sistema di CodeIgniter. Esse comprendono:

- core: il core del framework, che contiene le classi essenziali per il funzionamento di CodeIgniter;
- database: i files relativi al database, come ad esempio driver, query builder ecc..;
- fonts: contiene tutti i files legati alla gestione dei font;
- helpers: contiene gli helpers di base di CodeIgniter come ad esempio cookie, array, data ecc..;
- language: contiene i file di lingua;
- libraries: contiene le librerie di base di CodeIgniter.

La directory **application** contiene invece tutta l'applicazione, cioè tutti i dati che utilizza il framework ma anche tutto ciò che viene creato dallo sviluppatore; contiene infatti tutti i controller, il model e le viste relative al PELL.

Di seguito una rapida descrizione di alcune delle sub directories di application:

- Cache: Quando una pagina viene caricata per la prima volta, il file della cache viene scritto in questa cartella. Al caricamento delle pagine successive il file viene recuperato dalla stessa cartella e inviato al browser dell'utente richiedente. Se il file è scaduto, viene eliminato e aggiornato prima di essere inviato al browser.
- Config: la cartella contiene tutte le configurazioni dell'applicazione, di seguito i files più significativi:
 - config.php: descrive il base site URL, il file index.php, la lingua predefinita e il set di caratteri, percorso dei logs ecc.... CodeIgniter permette di aggiungere uno o più files di configurazione per dare la possibilità di customizzare le configurazioni; in Pell è stato aggiunto un file custom config-local.php che contiene tutta la configurazione dell'applicazione.
 - database.php: descrive la configurazione del database (Figura 3), e come per il file config.php in PELL è stato definito un file database-local.php custom.

```
$active_group = 'default';
$query_builder = TRUE;

$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => 'password',
    'database' => 'nm3_pell_v3',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);
```

Figura 3 – configurazione del database in config.php

- email.php: per la configurazione del server SMTP;
- permissions.php: per la configurazione dei permessi, in cui vengono definiti i percorsi che bypassano l'autenticazione (es. pagina di login, pagine statiche visibili agli utenti anonimi ecc.);
- routes.php: la configurazione dei percorsi, che permette di indirizzare le richieste verso il controller corretto.

Alcuni percorsi sono riservati, come ad esempio i seguenti:

```
$route['default_controller'] = 'Welcome';
$route['404_override']       = 'welcome/page_404';
$route['500_override']       = 'welcome/page_500';
$route['translate_uri_dashes'] = TRUE;
```

Mentre i seguenti sono i percorsi attualmente definiti:

```
$route['enea'] = 'enea/welcome/index';
$route['il-progetto'] = 'enea/welcome/il_progetto';
$route['la-piattaforma'] = 'enea/welcome/la_piattaforma';
$route['gli-stakeholders'] = 'enea/welcome/gli_stakeholders';
$route['pell-lumiere'] = 'enea/welcome/pell_lumiere';
$route['videolezioni'] = 'enea/welcome/videolezioni';
$route['come-aderire'] = 'enea/welcome/come_aderire';
$route['contatti'] = 'enea/welcome/contatti';
$route['registrazione-utente'] = 'enea/welcome/registrazione_utente';
$route['registrazione-gestore'] = 'enea/welcome/registrazione_gestore';
$route['illuminazione-pubblica'] = 'enea/welcome/illuminazione_pubblica';
$route['edifici'] = 'enea/welcome/edifici';
$route['download'] = 'enea/welcome/download';
$route['lenicalc'] = 'enea/welcome/lenicalc';

$route['pell-application'] = 'enea/pell_application';
$route['scheda-censimento'] = 'enea/scheda_censimento/index';
$route['scheda-censimento/form'] = 'enea/scheda_censimento/form';
$route['scheda-censimento/form/(:num)'] = 'enea/scheda_censimento/form/$1';
$route['scheda-censimento/form/(:num)/(:num)'] = 'enea/scheda_censimento/form/$1/$2';
$route['scheda-censimento/export/(:num)'] = 'enea/scheda_censimento/export/$1';
$route['scheda-censimento/kpi/(:num)/(:any)'] = 'enea/scheda_censimento/kpi/$1/$2';
$route['oxytech'] = 'enea/oxytech/index';

$route['isimm'] = 'enea/nisimm/index';
$route['isimm/content'] = 'enea/nisimm/content';
$route['isimm/recap/(:any)'] = 'enea/nisimm/recap/$1';
$route['isimm/recap/(:any)/(:any)'] = 'enea/nisimm/recap/$1/$2';
$route['isimm/impianti'] = 'enea/nisimm/impianti';
$route['isimm/investimenti'] = 'enea/nisimm/investimenti';
$route['isimm/sez_asis'] = 'enea/nisimm/sez_asis/index';
$route['isimm/sez_tobe'] = 'enea/nisimm/sez_tobe/index';
```

- enea-pell.php: contiene gli URLs degli host di frontend e oxytech e i percorsi dei validatori:

```
$config['enea'] = [
    'backend_host' => 'http://192.168.34.122:8080',
    'oxytech_host' => 'http://192.107.92.27:9090',
];

$config['schematron'] = [
    'validator_path' => 'third_party/ValidazioneSchematron',
    'schematron_path' =>
'third_party/ValidazioneSchematron/sch/CensusTechSheetRequirements.sch',
    'result_path' => 'third_party/ValidazioneSchematron/schemas',
];
```

- template.php: contiene il nome dell'applicazione, il tipo di layout e indica dove recuperare i file css relativi al tema utilizzato;
- xml.php: descrive il percorso delle cartelle di upload e dove sono i file per il caricamento e il controllo della scheda censimento;

```
$config['upload_path'] = './uploads/';
$config['xsd'] = base_url().'assets/data/CensusTechSheet.xsd';
$config['xml_assets'] = './assets/data/xml/';
```

- auth.php: è un file di configurazione importante perché gestisce l'autenticazione degli utenti.
- Controllers: questa cartella contiene tutti i controller definiti dall'utente.
- Core: quando CodeIgniter è avviato, vengono inizializzate molte classi dalle cartelle di sistema in system, ad esempio le configurazioni, le eccezioni, i controllers, i log ecc. È possibile che un utente voglia customizzare una di queste classi già presenti nel framework, in tal caso basta creare in questa cartella un nuovo file che estenda il file da customizzare. In PELL, in questa cartella, sono definiti alcuni controllers custom, ad esempio My_Controller che estende CI_Controller e si occupa del caricamento della pagina, della lingua, dei files di configurazione ove serve e inizializza il template. Un altro file custom importante è il My_Model, che estende il CI_Model e definisce tutte le operazioni sul db di cui necessita l'applicazione.
- Helpers: contiene delle funzioni che possono essere utili allo sviluppatore; possono essere create manualmente e, come permission, language, o anche form, file ecc..., sovrascrivono molti helper già presenti nella cartella application/system/helpers di CodeIgniter; in caso il sistema non trovi l'helper in application/helpers lo cerca in system/helpers.
- Language: è possibile definire ulteriori lingue oltre quella predefinita; come per gli helpers in caso il sistema non trovi il file di lingua in application/language lo cerca in system/language.
- Libraries: in questa cartella vengono inserite le librerie utilizzate dall'applicazione. Come per language e helpers esistono delle librerie di sistema che possono essere sovrascritte o aggiunte; CodeIgniter cerca, ove richiamate, prima le librerie in questa cartella e, se non le trova, le cerca in system/libraries.

In PELL sono state aggiunte delle librerie come ad esempio;

- Dompdf per la stampa in pdf dei report.
- Isimm per il calcolo delle valutazioni economico/finanziarie.
- Schematron per la validazione degli xml caricati.
- Bcrypt per la sicurezza della password.
- XML_IO per i check formali della scheda censimento.
- PHPMailer per l'invio delle email.
- Session per la gestione delle sessioni utente. Questa libreria merita un discorso a parte, in quanto è una libreria di sistema di CodeIgniter. La sessione, infatti, permette di tracciare l'attività degli utenti che navigano il portale, e in PELL è importante perché i dati dell'utente, memorizzati in sessione, sono necessari per alcune funzionalità. La sessione può essere salvata secondo le seguenti modalità:
 - Su file.
 - Su database.
 - Attraverso REDIS.
 - Attraverso MEMCACHED.
 - Modalità custom definita dall'utente attraverso lo sviluppo di librerie apposite.

La sessione di un utente PELL viene salvata sul database, nella tabella "ci_sessions".

- Logs: in questa cartella vengono gestiti i files per il logging degli errori

- **Models:** contiene tutti i file di model del PELL. Alcuni di essi estendono il My_Model e come già indicato definiscono le chiamate verso il database
- **Third_party:** la cartella contiene altri plugins e applicazioni anche custom. Un esempio è la cartella ValidazioneSchematron che contiene tutte le librerie per validare la scheda censimento.
- **Views:** contiene le viste dell'applicazione per la generazione dell'html.

In generale lo sviluppo di una funzionalità prevede:

- la creazione di un controller nella cartella application/controllers;
- la creazione di un controller nella cartella application/controllers/api in caso di necessità di definire delle REST;
- la creazione di una view nella cartella application/views;
- la creazione di un model nella cartella application/models se necessario l'accesso al database;
- la creazione di una funzione all'interno del controller;
- all'interno della funzione:
 - la definizione di variabili per la gestione della pagina web (title, subtitle ecc.);
 - l'invocazione delle dipendenze css e js che agiscono sulla pagina, ed eventuali plugin;
 - il puntamento ai files javascript e css che gestiscono la pagina;
 - il reindirizzamento verso la view associata.

2.2 Struttura dell'applicazione: Home page

Quando viene invocata la URL, ad esempio <https://www.pell.enea.it/>, viene chiamato il file `index.php` che contiene il nome del progetto (ENEA) e reindirizza verso la URL corretta <https://www.pell.enea.it/enea/> che è la home page di PELL.

Viene quindi invocato il controller `Welcome.php` (che estende `My_Controller.php`) che si occupa di inizializzare il template attraverso la funzione `init_template()` in

```
case 'private':
    default:
        if($this->router->fetch_class() == 'auth'){
            $this->layout = $this->layout_folder.'/layout_auth';
            $this->init_template_auth();
        }
        else{
            $this->verify_login();
            $this->layout = $this->layout_folder.'/layout';
            $this->init_template_protected();
        }
        break;
    }
}
```

Figura 4.

In generale, e questo vale per ogni pagina, l'inizializzazione del template prevede:

- caricamento delle librerie;
- caricamento dei plugin;
- caricamento dei percorsi che vengono visualizzati nella menu bar;
 - ✓ welcome_index
 - ✓ welcome_il_progetto
 - ✓ welcome_la_piattaforma
 - ✓ welcome_gli_stakeholders
 - ✓ welcome_pell_lumiere
 - ✓ welcome_videolezioni
 - ✓ welcome_come_aderire
 - ✓ welcome_contatti
 - ✓ welcome_registrazione_utente
 - ✓ welcome_registrazione_gestore
 - ✓ welcome_download
 - ✓ welcome_aziende
- il tipo di template da utilizzare, a seconda del layout definito nel file di configurazione `template.php` che può essere:
 - ✓ public se accessibile da utente anonimo
 - ✓ protected se accessibile da un utente autenticato

L'applicazione presenta due strutture di layout diverse a seconda che la pagina sia visualizzabile da un utente anonimo oppure da un utente autenticato.

Quando viene chiamata la URL, oltre a venire caricati altri plugin, librerie (Figura 6), css e i javascript (Figura 5), viene valutato il parametro che definisce il tipo di layout:

- Se il layout è protected, i dati dell'utente caricati dalla sessione (username, città associate, ruolo ecc...) vengono recuperati per essere utilizzati, ad esempio, per determinare la visibilità sulle schede censimento e gli ambiti di operatività.

- se il layout è pubblico, come ad esempio nella home, questo non avviene.

```

public function init_template () {
    /* Loads */
    $this->load->library('parser');
    $this->load->config('template');
    /* Template var set */
    $this->theme = $this->config->item('theme');
    $this->layout_type = $this->config->item('layout_type');
    $this->layout_folder = 'layout/'.$this->config->item('layout');
    /* Data to pass in layout */
    $this->data['base_assets'] = base_assets();
    $this->data['base_url'] = base_url();
    $this->data['base_title'] = $this->config->item('base_title');
    $this->data['current_route'] = $this->router->class.'_'.$this->router->method;
    $this->data['current_controller'] = $this->router->class;
    $this->data['current_method'] = $this->router->method;
    /* START ENEA TEMPLATE VARS */
    $welcome_routes = array(
        'welcome_index',
        'welcome_il_progetto',
        'welcome_la_piattaforma',
        'welcome_gli_stakeholders',
        'welcome_pell_lumiere',
        'welcome_videolezioni',
        'welcome_come_aderire',
        'welcome_contatti',
        'welcome_registrazione_utente',
        'welcome_registrazione_gestore',
        'welcome_download'
    );
    if (in_array($this->data['current_route'], $welcome_routes)){
        $horizontal_menu_class = '';
        $banner = '';
    }
    elseif ($this->data['current_route'] === 'welcome_edifici' || $this->data['current_route'] ===
'welcome_lenicalc'){
        $banner = 'custom_banner-ed';
        $horizontal_menu_class = 'ed-horizontal-menu';
    }
    else{
        $banner = 'custom_banner-ip';
        $horizontal_menu_class = 'ip-horizontal-menu';
    }
    $this->data['banner'] = $banner;
    $this->data['horizontal_menu_class'] = $horizontal_menu_class;
    /* END ENEA TEMPLATE VARS */
    /* Define Layout to use */
    switch($this->layout_type){
        case 'public':
            $this->layout = $this->layout_folder.'/layout_public';
            $this->init_template_public();
            break;
        case 'protected':
            if(!can($this->router->class.'_'.$this->router->method)){
                redirect('/');
            }

            if(!$this->session->has_userdata('logged_in')){
                $this->layout = $this->layout_folder.'/layout_public';
                $this->init_template_public();
            }
            else{
                $this->layout = $this->layout_folder.'/layout';
                $this->init_template_protected();
            }
            break;
    }
}

```

```

case 'private':
    default:
        if($this->router->fetch_class() == 'auth'){
            $this->layout = $this->layout_folder.'/layout_auth';
            $this->init_template_auth();
        }
        else{
            $this->verify_login();
            $this->layout = $this->layout_folder.'/layout';
            $this->init_template_protected();
        }
        break;
    }
}

```

Figura 4 – My_Controller.php, funzione init_template()

```

public function init_core () {
    //CSS
    $this->data['css_core'] = [
        ['url' => ('//fonts.googleapis.com/css?family=Open+Sans')],
        ['url' => ('//fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400italic,600')],
        ['url' => base_url('assets/css/styles.css')],
        ['url' => base_url('assets/css/styles-responsive.css')],
        ['url' => base_url('assets/css/plugins.css')],
        ['url' => base_url('assets/css/themes/theme-'.$this->theme.'.css'), 'attr' =>
        'type="text/css" id="skin_color"'],
        ['url' => base_url('assets/css/print.css'), 'attr' => 'type="text/css" media="print"']
    ];

    //Javascripts
    $this->data['js_core'] = [
        ['url' => base_url('assets/js/firstlanding.js')],
        ['url' => base_url('assets/js/index.js')],
        ['url' => base_url('assets/js/main.js')],
        ['url' => base_url('assets/js/common/search_autocomplete.js')]
    ];
}

```

Figura 5 -Caricamento Js e css

```

public function init_plugins () {
    //CSS
    $this->data['css_plugins'] = [
        ['url' =>
        ('http://fonts.googleapis.com/css?family=Raleway:400,300,500,600,700,200,100,800'), 'attr' =>
        'type="text/css"'],
        ['url' => base_url('assets/plugins/bootstrap/css/bootstrap.min.css')],
        ['url' => base_url('assets/plugins/font-awesome/css/font-awesome.min.css')],
        ['url' => base_url('assets/plugins/iCheck/skins/all.css')],
        ['url' => base_url('assets/plugins/perfect-scrollbar/src/perfect-scrollbar.css')],
        ['url' => base_url('assets/plugins/animate.css/animate.min.css')]
    ];
}

```

Figura 6 - caricamento plugins

La Figura 7 mostra la funzione `init_template_public()` che inizializza un template visibile a tutti, mentre la Figura 8 mostra la funzione `init_template_protected()` che inizializza un template privato.

Come si vede le funzioni inizializzano più zone del template, ma la funzione `init_template_protected()` passa al layout i dati relativi all'utente, dati che sono in sessione nel momento in cui l'utente è autenticato.

Essi sono:

- username
- profilo di visibilità
- ruolo
- città associate ove presenti

```
public function init_template_public () {

    $this->data['base_assets'] = base_assets();
    $this->data['base_url'] = base_url();

    $this->load->library('parser');
    $this->data['base_title'] = config_item('base_title');
    $this->data['form_action'] = base_url();

    $this->data['username'] = ($this->session->has_userdata('logged_in')) ? $this->session-
>userdata['username'] : '';
    //Init Assets
    $this->init_plugins();
    $this->init_plugins_auth();
    $this->init_subviews();
    //Init Page Assets
    $this->init_page_css_js();
    //Init Core Assets
    $this->init_core();
    //Init javascript class
    $this->init_javascript_class();

    $this->data['year'] = date('Y');
    $this->data['login'] = $this->parser->parse('auth/login', $this->data, TRUE);

    $this->data['js_page'] = [
        ['url' => base_url('assets/js/main.js')],
        ['url' => base_url('assets/plugins/jquery-validation/dist/jquery.validate.min.js')],
        ['url' => base_url('assets/js/login.js')],
    ];

    //Init layout components
    $this->init_sliding_bar();
    $this->init_top_bar();
    $this->init_left_menu();
    $this->init_header();
    $this->init_horizontal_menu();
    $this->init_subview_menu();
    $this->init_right_menu();
    $this->init_subview();
    $this->init_footer();

}
```

Figura 7 - inizializzazione template pubblico

```

public function init_template_protected(){
    /* Data to pass in layout */
    $this->data['username'] = $this->session->has_userdata('logged_in') ? $this->session-
>userdata['username'] : '';
    $this->data['profile_pic'] = $this->session->has_userdata('logged_in') ? $this-
>session->userdata['profile_pic'] : '';
    $this->data['user_role'] = $this->session->has_userdata('logged_in') ? $this-
>session->userdata['username'] : '';

    if ( $this->session->has_userdata('municipalities') ) // per-project userdata
    {
        $this->data['municipalities_list'] = $this->session->has_userdata('logged_in') ?
implode(' ', $this->session->userdata('municipalities')) : '';
    }

    /* Layout assets init */
    /* Init Core Assets (template related) */
    $this->init_core();
    /* Init Plugin Assets */
    $this->init_plugins();
    $this->init_subviews();
    /* Init Page Assets */
    $this->init_page_css_js();
    //Init javascript's Init()
    $this->init_javascript_class();
    /* Init Parsing of Layout Components */
    $this->init_sliding_bar();
    $this->init_top_bar();
    $this->init_header();
    $this->init_horizontal_menu();
    $this->init_left_menu();
    $this->init_subview_menu();
    $this->init_right_menu();
    $this->init_subview();
    $this->init_footer();
}

```

Figura 8 - inizializzazione template protetto

In entrambi i casi vengono inizializzate direttamente le zone della pagina da visualizzare attraverso alcune delle funzioni indicate in Figura 9.

È importante specificare che il template inizializza tutte le zone definite, tuttavia non tutte saranno valorizzate e perché dipende dai dati che vengono visualizzati nella view e dalla view stessa che le richiama se servono.

Ad esempio la Home Page utilizza solo l'horizontal_menu, il content e il footer, mentre la pagina "Illuminazione pubblica" utilizza anche il right_menu per il login e la visualizzazione delle funzionalità del sistema.

```

/* Init Core Assets (template related) */
$this->init_core();
/* Init Plugin Assets */
$this->init_plugins();
$this->init_subviews();
/* Init Page Assets */
$this->init_page_css_js();
//Init javascript's Init()
$this->init_javascript_class();
/* Init Parsing of Layout Components */
$this->init_sliding_bar();
$this->init_top_bar();
$this->init_header();
$this->init_horizontal_menu();
$this->init_left_menu();
$this->init_subview_menu();
$this->init_right_menu();
$this->init_subview();
$this->init_footer();
}

```

Figura 9 - caricamento zone viste

In ciascuna zona del layout vengono visualizzati i dati recuperati dal model attraverso le query; il controller passa i dati caricati dal model alla view, in modo che ogni zona sia valorizzata con quanto presente sul db o direttamente lato codice.

Ovviamente non è detto che il model sia sempre invocato dal controller; infatti esso viene chiamato solo quando vi è una necessità di recuperare i dati dal database.

Nel caso di pagine statiche o no per le quali non sono presenti funzioni di accesso al database, non viene richiamato alcun file dal model.

Un esempio è la home page (Figura 10), che è una pagina statica che non prevede accessi al database, pertanto non è necessaria la configurazione di chiamate al database.

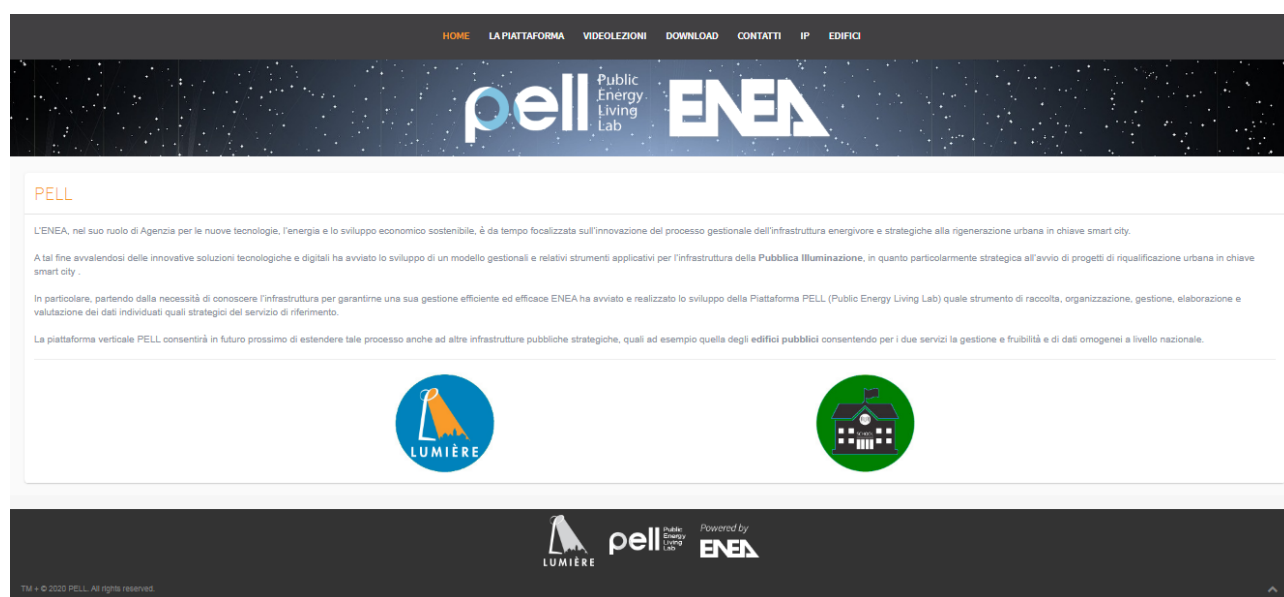


Figura 10 - home

```
<body class="layout-public sidebar-close horizontal-menu-fixed"
    data-site-url="{<?php echo site_url(""); ?>}"
    data-base-url="{<?php echo base_url(""); ?>}"
<div class="main-wrapper">
    <!-- start: HORIZONTAL MENU -->
    {horizontal_menu}
    <!-- end: HORIZONTAL MENU -->
    <!-- start: MAIN CONTAINER -->
    <div class="main-container inner">
        <!-- start: PAGE -->
        <div class="main-content">
            <!-- start: PANEL CONFIGURATION MODAL FORM -->
            <div class="modal fade" id="panel-config" tabindex="-1" role="dialog" aria-hidden="true">
                <div class="modal-dialog">
                    <div class="modal-content">
                        <div class="modal-header">
                            <button type="button" class="close" data-dismiss="modal" aria-hidden="true">
                                &times;
                            </button>
                            <h4 class="modal-title">Panel Configuration</h4>
                        </div>
                        <div class="modal-body">
                            Here will be a configuration form
                        </div>
                        <div class="modal-footer">
                            <button type="button" class="btn btn-default" data-dismiss="modal">
                                Close
                            </button>
                            <button type="button" class="btn btn-primary">
                                Save changes
                            </button>
                        </div>
                    </div>
                </div>
            <!-- /.modal-content -->
        </div>
        <!-- /.modal-dialog -->
    </div>
    <!-- /.modal -->
    <!-- end: SPANEL CONFIGURATION MODAL FORM -->
    <div id="custom_banner" class="row_fluid {<?php echo $banner?>}">
        <div id="logos">
            <div id="logo_pell"></div>
            <div id="logo_enea"></div>
        </div>
    </div>

    <?php if($current_route == 'welcome_illuminazione_pubblica'){?>
    <div class="container">
        <div class="row">
            <div class="col-md-9">
                {content}
            </div>
            <div class="col-md-3">
                <div class="panel panel-white">
                    <div class="panel-heading border-light">
                        <h4 class="panel-
title">Login</h4>
                    </div>
                    <div class="panel-body">
                        {login}
                    </div>
                </div>
            </div>
        </div>
    </div>
```

```
<?php } else {?>
<div class="container">
    <div class="row">
        <div class="col-md-12">
            {content}
        </div>
    </div>
</div>
<?php }?>

<div class="subviews">
    <div class="subviews-container"></div>
</div>
</div>
<!-- end: PAGE -->
```

Figura 11 - layout_public.php

Nella Figura 11 è mostrato il body della vista relativa al layout pubblico (application/views/layout/enea/layout_public.php).

Come si può vedere, questo layout è utilizzato sia dalla home page, che dalla pagina “illuminazione pubblica” (\$current_route = ‘welcome_illuminazione_pubblica’), infatti il codice prevede che se il percorso è quello della pagina “illuminazione pubblica”, deve essere visualizzato anche il form di login (login) oltre che il contenuto della pagina (“content”).

Il layout presenta varie sezioni, per ciascuna delle quali il progetto mostra files separati, in modo da costruire ogni volta la pagina come serve in base alla richiesta del server.

- **L’horizontal menu**, inizializzato nel template attraverso la funzione init_horizontal_menu(), è la regione in cui vengono visualizzati i link alle pagine statiche del menu bar, precedentemente inizializzate.

La vista relativa (application/views/layout/enea/horizontal_menu.php), in Figura 13 presenta tutti i link url alle pagine statiche che sono presenti in application/views /enea/static.

In Figura 12 le pagine statiche all’interno della cartella application/views /enea/static

Nome	Tipo
 come_aderire	File PHP
 contatti	File PHP
 download	File PHP
 edifici	File PHP
 gli_stakeholders	File PHP
 il_progetto	File PHP
 illuminazione_pubblica	File PHP
 la_piattaforma	File PHP
 lenicalc	File PHP
 registrazione_gestore	File PHP
 registrazione_utente	File PHP
 videolezioni	File PHP

Figura 12 – pagine statiche

```
<div id="horizontal-menu" class="navbar navbar-inverse hidden-xs inner <?php echo $horizontal_menu_class?>">
    <div class="container text-center">
        <div class="navbar-collapse">
            <ul class="nav navbar-nav">
                <li <?php if($current_route === 'welcome_index') echo 'class="active"'?>>
                    <a href="<?php echo site_url('/')?>">
                        Home
                    </a>
                </li>
                <li <?php if(site_url('/la-piattaforma') === current_url()) echo
'<div class="active">
    <a href="<?php echo site_url('/la-piattaforma')?>">
        La piattaforma
    </a>
</div>
    <li <?php if(site_url('/videolezioni') === current_url()) echo 'class="active"'
?>>
        <a href="<?php echo site_url('/videolezioni')?>">
            Videolezioni
        </a>
    </li>
    <li>
        <a href="<?php echo site_url('/download')?>">
            Download
        </a>
    </li>
    <li <?php if(site_url('/contatti') === current_url()) echo 'class="active"'?>>
        <a href="<?php echo site_url('/contatti')?>">
            Contatti
        </a>
    </li>
    <li <?php if($current_route === 'welcome_illuminazione_pubblica') echo
'<div class="active">
    <a href="<?php echo site_url('/illuminazione-pubblica')?>">
        IP
    </a>
</div>
    <li <?php if($current_route === 'welcome_edifici') echo 'class="active"'?>>
        <a href="<?php echo site_url('/edifici')?>">
            Edifici
        </a>
    </li>
</ul>
</div>
</div>
</div>
```

Figura 13 – horizontal_menu

Che si traduce nella visualizzazione della barra come descritta in Figura 14:



Figura 14 – visualizzazione dell'horizontal_menu

Come già detto, per questa pagina non è stato richiamato alcun file dal model.

- **Content:** questa zona presenta il contenuto della home (Figura 15) (o della pagina “Illuminazione pubblica”), comprensiva dei link
 - Alla sezione Illuminazione pubblica
 - Alla sezione Edifici (Figura 16)

```

<!-- start: PAGE CONTENT -->
<div class="row">
  <div class="col-md-12">
    <div class="panel panel-white">
      <div class="panel-heading border-light">
        <h4 class="panel-title">PELL</h4>
      </div>
      <div class="panel-body">
        <p>L'ENEA, nel suo ruolo di Agenzia per le nuove tecnologie, l'energia e lo
sviluppo economico sostenibile, è da tempo focalizzata sull'innovazione del processo gestionale
dell'infrastruttura energivore e strategiche alla rigenerazione urbana in chiave smart city.<br><br>
A tal fine avvalendosi delle innovative soluzioni tecnologiche e digitali ha avviato lo sviluppo di un modello
gestionali e relativi strumenti applicativi per l'infrastruttura della <b>Pubblica Illuminazione</b>, in quanto
particolarmente strategica all'avvio di progetti di riqualificazione urbana in chiave smart city .<br><br>
In particolare, partendo dalla necessità di conoscere l'infrastruttura per garantirne una sua gestione
efficiente ed efficace ENEA ha avviato e realizzato lo sviluppo della Piattaforma PELL (Public Energy Living
Lab) quale strumento di raccolta, organizzazione, gestione, elaborazione e valutazione dei dati individuati
quali strategici del servizio di riferimento.<br><br>
La piattaforma verticale PELL consentirà in futuro prossimo di estendere tale processo anche ad altre
infrastrutture pubbliche strategiche, quali ad esempio quella degli <b>edifici pubblici</b> consentendo per i
due servizi la gestione e fruibilità e di dati omogenei a livello nazionale.</p>
<hr>

        <div class="row">
          <div class="col-md-4 col-md-offset-2 col-xs-6 text-center">
            <a href="<?php echo site_url('/illuminazione-pubblica')?>">
              
            </a>
          </div>
          <div class="col-md-4 col-xs-6 text-center">
            <a href="<?php echo site_url('/edifici')?>">
              
            </a>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
<!-- end: PAGE CONTENT-->

```

Figura 15 – content della homepage



Figura 16 – link a PELL IP e PELL EP

- **Footer:** il footer viene inizializzato nel template attraverso la funzione `init_footer()`, che valorizza il file relativo in `application/views/layout/enea/footer.php` (Figura 17); il footer è comune a tutte le pagine, come l' `horizontal_menu`, e visualizza i link al progetto Lumiere, Pell, Enea e i diritti di autore (Figura 18).

```
<footer class="inner">
  <div class="footer-inner">
    <div class="container text-center">
      <a href="http://lumiere.casaccia.enea.it/cms/"></a>
      <a href="<?php echo base_url()>"></a>
      <a href="http://www.enea.it"></a>
    </div>
    <div>
      <div class="pull-left">
        TM + © {year} PELL. All rights reserved.
      </div>
      <div class="pull-right">
        <span class="go-top"><i class="fa fa-chevron-up"></i></span>
      </div>
      <div class="clearfix"></div>
    </div>
  </div>
</footer>
```

Figura 17 – footer.php



Figura 18 –visualizzaione del footer

2.3 Struttura dell'applicazione: pagina Illuminazione Pubblica

Cliccando sul logo Lumiere dalla home di PELL, si accede alla pagina “Illuminazione pubblica”, visualizzabile in Figura 19



Figura 19 – pagina Illuminazione pubblica

Come la home page, anche questa pagina prevede un layout pubblico; come mostrato in Figura 11, il layout_public prevede anche la visualizzazione del form di login per gli utenti autenticati, oltre alle sezioni

- Horizontal_menu
- Content
- Footer

```
<body class="layout-public sidebar-close horizontal-menu-fixed"
    data-site-url="php echo site_url(""); ?&gt;"
    data-base-url="<?php echo base_url(""); ?&gt;"&gt;
&lt;div class="main-wrapper"&gt;
    &lt;!-- start: HEADER --&gt;
    {header}
    &lt;!-- end: HEADER --&gt;
    &lt;!-- start: HORIZONTAL MENU --&gt;
    {horizontal_menu}
    &lt;!-- end: HORIZONTAL MENU --&gt;
    &lt;!-- start: MAIN CONTAINER --&gt;
    &lt;div class="main-container inner"&gt;
        &lt;!-- start: PAGE --&gt;
        &lt;div class="main-content"&gt;
            &lt;!-- start: PANEL CONFIGURATION MODAL FORM --&gt;
            &lt;div class="modal fade" id="panel-config" tabindex="-1" role="dialog" aria-
hidden="true"&gt;
                &lt;div class="modal-dialog"&gt;
                    &lt;div class="modal-content"&gt;
                        &lt;div class="modal-header"&gt;
                            &lt;button type="button" class="close" data-
dismiss="modal" aria-hidden="true"&gt;
                                &amp;times;
                            &lt;/button&gt;
                            &lt;h4 class="modal-title"&gt;Panel Configuration&lt;/h4&gt;
                        &lt;/div&gt;
                        &lt;div class="modal-body"&gt;
                            Here will be a configuration form
                        &lt;/div&gt;
                        &lt;div class="modal-footer"&gt;
                            &lt;button type="button" class="btn btn-default" data-
dismiss="modal"&gt;
                                Close
                            &lt;/button&gt;
                            &lt;button type="button" class="btn btn-primary"&gt;
                                Save changes
                            &lt;/button&gt;
                        &lt;/div&gt;
                    &lt;/div&gt;
                &lt;/div&gt;
            &lt;!-- /.modal-content --&gt;
        &lt;/div&gt;
    &lt;!-- /.modal-dialog --&gt;
&lt;/div&gt;
&lt;!-- /.modal --&gt;
&lt;!-- end: SPANEL CONFIGURATION MODAL FORM --&gt;
&lt;?php
$welcome_routes = array(
    'welcome_index',
    'welcome_il_progetto',
    'welcome_la_piattaforma',
    'welcome_gli_stakeholders',
    'welcome_pell_lumiere',
    'welcome_videolezioni',
    'welcome_come_aderire',
    'welcome_contatti',
    'welcome_registrazione_utente',
    'welcome_registrazione_gestore',
    'welcome_download'
);</pre

```

```

if(in_array($current_route, $welcome_routes))
    $banner = '';
elseif ($current_route === 'welcome_edifici' || $current_route === 'welcome_lenicalc')
    $banner = 'custom_banner-ed';
else
    $banner = 'custom_banner-ip';
?>
<div id="custom_banner" class="row_fluid <?php echo $banner?>">
    <div id="logos">
        <div id="logo_pell"></div>
        <div id="logo_enea"></div>
    </div>
</div>

<div class="container">
    <div class="row">
        <div class="col-md-9">
            {content}
        </div>
        <div class="col-md-3">
            <div class="panel panel-white">
                <div class="panel-heading border-light">
                    <h4 id="pell_user_info" class="panel-title">Info
utente</h4>
                </div>
                <div class="panel-body">
                    <p>Benvenuto, <b>{username}</b></p>
                    <ul>
                        <li><b>Tipo utente: </b>
{user_role}</li>
                        <li><b>Comune: </b> <?php echo
isset($municipalities_list) ? $municipalities_list : 'Nessun comune associato' ?></li>
                    </ul>
                </div>
            </div>
            <div class="panel panel-white">
                <div class="panel-heading border-light">
                    <h4 id="pell_user_menu" class="panel-title">Menu
utente</h4>
                </div>
                <div class="panel-body no-padding">
                    {right_menu}
                </div>
            </div>
        </div>
    </div>

    <div class="subviews">
        <div class="subviews-container"></div>
    </div>
</div>
<!-- end: PAGE -->
</div>

```

Figura 20 –layout_protected

Il form di login viene visualizzato nel caso di layout pubblico, ma quando l'utente si autentica al portale, la pagina "illuminazione pubblica" viene visualizzata su layout protetto (

```

if(in_array($current_route, $welcome_routes))
    $banner = '';
elseif ($current_route === 'welcome_edifici' || $current_route === 'welcome_lenicalc')
    $banner = 'custom_banner-ed';
else
    $banner = 'custom_banner-ip';
?>
<div id="custom_banner" class="row_fluid <?php echo $banner?>">
    <div id="logos">
        <div id="logo_pell"></div>
        <div id="logo_enea"></div>
    </div>
    <div class="container">
        <div class="row">
            <div class="col-md-9">
                {content}
            </div>
            <div class="col-md-3">
                <div class="panel panel-white">
                    <div class="panel-heading border-light">
                        <h4 id="pell_user_info" class="panel-title">Info
utente</h4>
                    </div>
                    <div class="panel-body">
                        <p>Benvenuto, <b>{username}</b></p>
                        <ul>
                            <li><b>Tipo utente: </b>
{user_role}</li>
                            <li><b>Comune: </b> <?php echo
isset($municipalities_list) ? $municipalities_list : 'Nessun comune associato' ?></li>
                        </ul>
                    </div>
                </div>
                <div class="panel panel-white">
                    <div class="panel-heading border-light">
                        <h4 id="pell_user_menu" class="panel-title">Menu
utente</h4>
                    </div>
                    <div class="panel-body no-padding">
                        {right_menu}
                    </div>
                </div>
            </div>
        </div>
        <div class="subviews">
            <div class="subviews-container"></div>
        </div>
    </div>
<!-- end: PAGE -->
</div>

```

Figura 20), che non prevede più il form di login.

Si aggiungono infatti due sezioni:

- Pell_user_info (che mostra un messaggio di benvenuto all'utente in caso di login effettuato).
- Pell User menu in cui viene visualizzato il layout relativo al right_menu, inizializzato nel layout dalla funzione init_right_menu() di init_template().

Il right_menu Visualizza il menu utente laterale con i link seguenti se l'utente è autenticato:

- ✓ Pell application,
- ✓ Scheda censimento,

- ✓ Prestazioni illuminotecniche,
- ✓ Valutazioni economico-finanziarie,
- ✓ Amministrazione.

2.3.1 Utente anonimo –form login

Come già detto, ed evidenziato nel layout_public.php, se la pagina è “Illuminazione pubblica”, viene visualizzato il form di login (Figura 21).

Figura 21 – form login

La view relativa al form di login è in application/views/auth/login.php e mostrata in Figura 22.

```
<div class="box-login">
  <form class="form-login" action="{form_action}" method="post">
    <div class="errorHandler alert alert-danger no-display">
      <i class="fa fa-remove-sign"></i> You have some form errors. Please check below.
    </div>
    <div class="successHandler alert alert-success no-display">
      <i class="fa fa-thumbs-up"></i> Authentication correct.
    </div>
    <fieldset>
      <div class="form-group">
        <span class="input-icon">
          <input type="text" class="form-control" name="username" placeholder="{_username}">
          <i class="fa fa-user"></i> </span>
        </div>
      <div class="form-group form-actions">
        <span class="input-icon">
          <input type="password" class="form-control password" name="password" placeholder="Password">
          <i class="fa fa-lock"></i>
        </span>
      </div>
      <div class="form-actions">
        <label for="remember" class="checkbox-inline">
          <input type="checkbox" class="grey remember" id="remember" name="remember">
          {_keep_me_signed_id}
        </label>
        <button type="submit" class="btn btn-green pull-right">
          {_login_you} <i class="fa fa-arrow-circle-right"></i>
        </button>
      </div>
    </fieldset>
  </form>
</div>
```

Figura 22 – login.php

Quando vengono inserite le credenziali dell’operatore, viene instradata la richiesta al controller application/controllers/api/Authentication.php, che si occupa dell’autenticazione.

Il controller Authentication.php estende il REST_controller, che fa parte di una libreria CodeIgniter RESTful API e permette di creare api custom per l’applicazione.

Authentication.php comprende varie funzioni per la post del form e il recupero degli utenti; per prima cosa vengono caricati plugin e librerie per la gestione del form e la libreria BCrypt per la gestione della password. Viene poi richiamato il file di configurazione auth.php nella cartella config in cui vengono definite le modalità di login e di gestione della password.

Quando viene effettuata la post viene invocata la funzione login_post (Figura 23) che effettua la post del form e restituisce una response al browser che si occuperà di gestirla.

Dal momento che deve essere verificato l'accesso dell'utente, è necessario effettuare una query sul database, pertanto viene caricato il model user_model.php nella cartella application/model/common, che effettua una ricerca sulla tabella "User" tramite username (Figura 24).

Se l'utente non è presente viene restituito un messaggio di errore, altrimenti viene verificata la password attraverso la decriptazione (funzione bcrypt->verify(post(password), passwordDaDB)) che in caso di esito positivo inserisce l'utente in sessione.

```
public function login_post(){

    //Loads
    $this->load->model('common/user_model');

    //Bad request if post data are missing
    if(!$this->post('username') || $this->post('username') == '' || !$this->post('password') || $this->post('password') == '')
    {
        $this->response(NULL, REST_Controller::HTTP_BAD_REQUEST); // BAD_REQUEST (400) being the HTTP response code

        //Search the user in db
        $result = $this->user_model->get_by_username($this->post('username'));
        //No user found
        if(!$result){
            $this->response(['status' => FALSE, 'message' => 'Credenziali errate'], REST_Controller::HTTP_NOT_FOUND); // NOT_FOUND (404) being the HTTP response code
        }
        else{
            //Password verify
            if ($this->bcrypt->verify($this->post('password'), $result->password)) {

                // Store user data in session
                if($this->user_model->set_user_session($result, $this->post('remember') == 'on')){

                    $this->response(['status' => TRUE, 'message' => 'Authentication accomplished'], REST_Controller::HTTP_OK, TRUE); // OK (200) being the HTTP response code
                }
                else{
                    $this->response(NULL, REST_Controller::HTTP_INTERNAL_SERVER_ERROR); // HTTP_INTERNAL_SERVER_ERROR (500)
                }
            }
            else{
                $this->response(['status' => FALSE, 'message' => 'Credenziali errate'], REST_Controller::HTTP_UNAUTHORIZED); // UNAUTHORIZED (401) being the HTTP response code
            }
        }
    }
}
```

Figura 23 – authentication.php, function login_post()

```
public function get_by_username($username) {

    return $this->db->get_where($this->table, array('username' => $username))->row();

}
```

Figura 24 – user-model.php, function get

Poiché per il PELL viene utilizzata una memorizzazione della sessione sul database, è necessario invocare user_model.php per settare la sessione, attraverso la funzione set_user_session() (Figura 25).

```
public function set_user_session($user, $remember, $data = array()) {

    // $roles = $this->get_user_roles($user->id);
    $permissions = array();
    $result = $this->get_user_permissions($user->id);
    foreach ($result as $perm) {
        $permissions[$perm->id] = $perm->name;
    }

    $user_data = array(
        'id' => $user->id,
        'username' => $user->username,
        'first_name' => $user->first_name,
        'last_name' => $user->last_name,
        'email' => $user->email,
        'phone' => $user->phone,
        'gender' => $user->gender,
        'language' => $user->language,
        'logged_in' => TRUE,
        'remember' => $remember,
        // 'roles' => $roles,
        'permissions' => $permissions,
        'group' => $this->get_user_group($user->id),
        'profile' => $this->get_user_profile($user->id),
        'profile_pic' => ($user->picture != '') ? base_url($this->config-
>item('user_pic_dir').$user->picture) : base_url('assets/images/avatar/u1.png')
    );

    if(count($data) > 0){
        foreach($data as $key => $row){
            $user_data[$key] = $row;
        }
    }

    $this->load->library('session');
    $this->session->set_userdata($user_data);
    $this->update_last_login($user->id);
    return TRUE;

}
```

Figura 25 – user_model.php, funztion set_User_Session()

La chiamata POST dal browser viene effettuata tramite Ajax attraverso il javascript in assets/js/login.js che invoca il controller e procede con la validazione e l'autenticazione (Figura 26).

```

var runLoginValidator = function() {
    var form = $('.form-login');
    var errorHandler = $('.errorHandler', form);
    var successHandler = $('.successHandler', form);

    form.validate({
        rules: {
            username: {
                minlength: 2,
                required: true
            },
            password: {
                minlength: 6,
                required: true
            }
        },
        submitHandler: function(form) {
            errorHandler.hide();

            var username = $(form).find('input[name="username"]').val();
            var password = $(form).find('input[name="password"]').val();
            var remember = $(form).find('input[name="remember"]').val();

            var data = { username: username,
                          password: password,
                          remember: remember
            };

            $.ajax({
                url: siteurl+"api/authentication/login/",
                type: "POST",
                data: data,
                cache: false,
                success: function(data) {
                    errorHandler.hide();
                    successHandler.show();
                    form.submit();
                },
                error: function(xhr){
                    successHandler.hide();
                    errorHandler.html(xhr.responseJSON.message).show();
                    //alert(xhr.responseJSON.message);
                }
            });

            return false;
        },
        invalidHandler: function(event, validator) {///display error alert on form submit
            errorHandler.show();
        }
    });
};

```

Figura 26 - login.js

2.3.2 Utente autenticato – menu laterale

In seguito all'autenticazione dell'operatore, viene visualizzato il menu laterale.

A differenza dell'horizontal_menu, in cui le voci di menu sono pagine statiche accessibili direttamente tramite url definito nel template, le voci del menu laterale sono caricate da database.

Esse sono storicizzate nella tabella del database c_modules e vengono richiamate dalla funzione init_left_menu() del Welcome.php che a sua volta chiama le funzioni del Menu_modules_model, in application/models/common/, e carica da database le voci di menu.

La view richiamata dal controller, application/views/layout/enea/right_menu.php è descritta in Figura 27Figura 29;

In essa viene effettuato un ciclo sulla variabile \$menu, che viene valorizzata attraverso recupero dei dati dal db.

Il controller chiama la funzione init_left_menu() (Figura 27) che invoca il menu_modules_model.php ed ,in particolare, la funzione menu_from_permission(), (Figura 28) che effettua la query sulla tabella indicata "c_modules" e restituisce una lista di voci di menu.

```
public function init_left_menu () {

    $this->load->model('common/menu_modules_model');

    $menu_raw = $this->menu_modules_model->menu_from_permissions();
    $menu = [];
    foreach ($menu_raw as $row)
    {
        if (intval($row->iscategory) == 0 && $row->father == '')
            $menu[$row->name] = ['info' => $row];
        elseif (intval($row->iscategory) == 0)
            $menu[$row->father]['children'][$row->url] = $row;
        else
            $menu[$row->name] = ['info' => $row, 'children' => []];
    }

    $this->data['menu'] = $menu;
    $this->data['left_menu'] = $this->parser->parse($this->layout_folder.'/left_menu',
array_merge($this->data, $this->left_menu_data), TRUE);
}
```

Figura 27 – funzione init_left_menu()

```
class Menu_modules_model extends MY_Model {

    public $table = 'c_modules';

    /**
     * construct
     */
    function __construct() {
        /* Call the Model constructor */
        parent::__construct();
    }

    function menu_from_permissions(){

        // Non admin users can see all the menu voices
        $this->db->where('isactive = 1');
        if (!is_admin()) {
            //Menu voices available to everyone
            $this->db->where('permission IS NULL');
            $permissions = $this->session->userdata('permissions');
            if($permissions)
                foreach($permissions as $key=>$permission){
                    if(strpos($permission, 'administration') !== false) {
                        unset($permissions[$key]);
                    }
                }

            //Includes menu voices with defined permission, if user has at least one permission
            if(is_array($permissions) && count($permissions) > 0)
                $this->db->or_where_in('permission', $permissions);
        }
        $this->db->order_by("position", "asc");

        return $this->db->get($this->table)->result();
    }

}
```

Figura 28 - menu_modules_model.php

Per aggiungere una nuova voce di menu, è necessario effettuare una insert sulla tabella c_modules indicando:

- Nome e label
- url
- icon
- position
- permission
- is_active

```
<ul class="main-navigation-menu">

    <?php foreach ($menu as $key => $item) {
        if(!array_key_exists('children', $item)){?>
            <li <?php if(uri_string() == $item['info']->url) echo 'class="active"'?>>
                <a href="<?php echo site_url($item['info']->url)?>">
                    <i class="fa fa-fw fa-<?php echo $item['info']->icon?>"></i>
                    <span class="title"><?php echo ucfirst($this->lang-
>line($item['info']->label)?></span>
                        </a>
                    </li>
                <?php }
            else{?>
                <li <?php if(array_key_exists(uri_string(), $item['children'])) echo
'<class="active"'?>>
                    <a href="javascript:void(0)">
                        <i class="fa fa-fw fa-<?php echo $item['info']->icon?>"></i>
                        <span class="title"><?php echo ucfirst($this->lang-
>line($item['info']->label)?></span>
                            </a>
                            <ul class="sub-menu">
                                <?php foreach($item['children'] as $row){?>
                                    <li <?php if(uri_string() == $row->url) echo
'<class="active"'?>>
                                        <a href="<?php echo site_url($row->url)?>">
                                            <i class="fa fa-fw fa-<?php echo $row-
>icon?>"></i>
                                            <span class="title"><?php echo
ucfirst($this->lang->line($row->label)?></span>
                                                </a>
                                                </li>
                                            <?php }?>
                                        </ul>
                                    </li>
                                <?php }?>
                            </ul>
                        <?php }?>
                    </li>
                </ul>
            </ul>
        </ul>
    </ul>
```

Figura 29 –view right_menu.php

A questo punto sarà visibile il nuovo link all'utente autenticato sul portale.

2.4 Struttura dell'applicazione: Amministrazione

Effettuata l'autenticazione, il menu laterale presenta dei link alle varie sezioni dell'applicazione.

Ogni operatore possiede dei permessi specifici in base:

- Al ruolo
- Ai comuni ad esso associati

La Figura 30 mostra la tabella utente con

- un elenco di utenti,
- la tipologia di utenza (Gruppo) e
- comuni di appartenenza.

Ogni gruppo presenta dei permessi specifici, in particolare:

- Admin: può accedere e gestire la scheda censimento, la sezione Pell Application, la sezione oxytech e il modulo save per le valutazioni economico/finanziarie, gestire account, utenti, permessi e gruppi
- Gestore: può accedere e gestire la scheda censimento, la sezione Pell Application, la sezione oxytech e il modulo save per le valutazioni economico/finanziarie.
- Oxytech: può accedere e gestire solo la sezione oxytech
- Sindaco: può accedere e gestire la scheda censimento, la sezione Pell Application, la sezione oxytech e il modulo save per le valutazioni economico/finanziarie ma con visibilità limitata.
- Utente: può visualizzare e gestire la scheda censimento con una visibilità limitata.

TABELLA UTENTI							+ Crea Utente	↻
Show	10	entries	Search:					
Id ^	Nome utente	Nome	Cognome	Gruppo	Email	Comune	Azioni	
1	administrator	Administrator	TestENEA	Admin		Napoli, Firenze, Roma, Milano, Bologna, Palermo, Genova		
50	test_sindaco	Sindaco	TestENEA	Sindaco		Bologna		
53	test_gestore	Gestore	TestENEA	Gestore		Roma, Bologna, Abano Terme		
54	test_oxytech	Oxytech	TestENEA	Oxytech				
64	test_utente	Utente	TestENEA	Utente				
66	citygreenlight	Dario		Gestore		La Spezia, Borgomaro		
67	cimel	Giovanni		Utente				
68	belligm	Emanuele		Sindaco		Bellaria-Igea Marina		

Showing 1 to 8 of 8 entries

Previous **1** Next

Figura 30- utenti

In ogni caso i permessi sono ampiamente configurabili nella sezione apposita; è infatti possibile personalizzare i gruppi e crearne di nuovi attraverso il tasto “Crea gruppo”, come visibile in Figura 31. **Errore. L'origine riferimento non è stata trovata..**

TABELLA GRUPPI

+ Crea Gruppo

Show

10

entries

Search:

Id	Nome	Descrizione	Profilo di visibilità	Azioni
48	Admin	admin	all	
46	Gestore		all	 
47	Oxytech		all	 
45	Sindaco			 
49	Utente	utente		 

Showing 1 to 5 of 5 entries

Previous

1

Next

Figura 31 - gruppi

È poi possibile configurare i permessi in base alle necessità, come mostrato in Figura 32Figura 32, dove sono definiti, come esempio, i permessi di gestione della scheda censimento, copy, delete, upload ecc.

RUOLI

scheda_censimento

Can use scheda censimento

scheda_censimento_form

scheda_censimento_index

scheda_censimento_copy_get

scheda_censimento_create_polygon_post

scheda_censimento_delete_get

scheda_censimento_delete_node_get

scheda_censimento_form_get

scheda_censimento_form_post

scheda_censimento_history_get

scheda_censimento_index_get

scheda_censimento_index_post

scheda_censimento_kpi_refresh_get

scheda_censimento_kpi_table_get

scheda_censimento_nodes_get

scheda_censimento_polygon_get

scheda_censimento_submit_get

scheda_censimento_table_get

scheda_censimento_update_polygon_post

scheda_censimento_upload_post

scheda_censimento_user_location_get

administration_index

scheda_censimento_export

scheda_censimento_kpi

scheda_censimento_prova

Figura 32 -permessi

La schermata di amministrazione permette di gestire utenti, profili, gruppi e permessi:

ACCOUNT

-  **Utenti**
Gestione anagrafica e password degli utenti.
-  **Gruppi**
Gestione anagrafica dei gruppi di utenti
-  **Ruoli**
Gestione anagrafica dei ruoli e dei permessi.
-  **Permessi**
Gestione dei permessi relativi ai ruoli

Figura 33 – sezione amministrazione

Il controller relativo alla pagina di amministrazione è application/controllers/Administrations.php, e la vista corrispondente è application/views/administration/administration.php (Figura 34).

Il controller si preoccupa di inizializzare il layout di tipo protected, e di caricare le varie sezioni, descritte nei paragrafi seguenti.

```
<div class="row">
  <div class="col-sm-12 col-xs-12">
    <?php if(array_key_exists('accounts', $sections)):?>
      <div class="panel panel-white">
        <div class="panel-heading border-light">
          <h4 class="panel-title"><?php echo $this->lang->line("accounts"); ?></h4>
        </div>
        <div class="panel-body menu-voci">
          <?php foreach($sections['accounts'] as $ctrl => $icon):?>
            <a href="<?php echo site_url("administration/".$ctrl); ?>">
              <div class="col-sm-12">
                <p>
                  <i class="fa fa-<?php echo $icon;?>"></i><?php
echo $this->lang->line($ctrl); ?><br />
                  <span class="descrizione"><?php echo $this-
>lang->line($ctrl."_description"); ?></span>
                </p>
              </div>
            </a>
          <?php endforeach;?>
        </div>
      </div>
    <?php endif;?>
  </div>
</div>

<div id="modal_site_info" class="modal">
  <div class="modal-dialog">
    <div class="modal-content">
      <!-- header -->
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
      <!-- title -->
      <!-- <h4 class="modal-title"><?php echo $this->lang->line("general_info")
?></h4> -->
    </div>
  </div>
</div>
```

```

<!-- body -->
        <div class="modal-body">
            <div class="row">
                <div class="col-md-3 col-xs-4">
                    " />
                </div>
                <div class="col-md-9 col-xs-8">
                    <p class="modal_info_title"><?php echo $this->lang->
>line("dashboard") ?></p>
                    <p class="modal_info_desc"><?php echo $this->lang->
>line("powered_by") ?> <a target="_blank" href="http://www.harpaItalia.it">Harpa Italia srl</a></p>
                    <p class="modal_info_desc"><?php echo $this->lang->
>line("version") ?>: <?php echo $version; ?></p>
                    <p class="modal_info_desc"><a href="javascript:;"
onclick="$('#changelog_container').toggle();"><?php echo $this->lang->line("changelog") ?> <i style="vertical-
align: middle" class="md md-keyboard-arrow-down"></i></a></p>
                </div>
                <div id="changelog_container" class="col-xs-12 panel-scroll"
style="display:none;">
                    <?php foreach($change_log as $log) {
                        $des_html = "";
                        $des = explode("\n", $log["description"]);
                        for($j = 0; $j < count($des); $j++) {
                            $des_html .= sprintf("<li>%s</li>\n", $des[$j]);
                        }
                        echo '<div class="row changelog_version_container">
                            <div class="col-xs-3">
                                <p class="changelog_version">'. $this->
>lang->line("version").' '. $log["version"].'</p>
                                </div>
                                <div class="col-xs-9">
                                    <p
class="changelog_description">'. $des_html.'</p>
                                    </div>
                                </div><hr>';
                    }?>
                </div>
            </div>
        <div class="modal-footer">
            <button type="button" class="btn btn-default" data-dismiss="modal"><?php echo
$this->lang->line("close") ?></button>
        </div>
    </div>
</div>

```

Figura 34- vista administration.php

2.4.1 Gruppi

Cliccando sulla voce “Gruppi”, il controller chiama la funzione groups (Figura 35) in Administrations.php; poiché il sistema deve caricare dal database i dati relativi ai gruppi, viene caricato

- il role_model.php in models/common/ che effettua le chiamate al database per gestire i ruoli,
- i profili di visibilità attraverso il model profile_model.php in models/common/ che effettua le chiamate al database per gestire i profili di visibilità (cosa vede l’operatore)

```
public function groups()
{
    //Roles
    $this->load->model('common/role_model');
    $roles = $this->role_model->get_all();
    $this->data['roles'] = $roles;

    //Profile List
    $this->load->model('common/profile_model');
    $profileOptions = $this->profile_model->get_list_profile();
    $this->data['profile_options'] = $profileOptions;

    $this->data['page_title'] = $this->lang->line('groups');
    $this->data['page_subtitle'] = $this->lang->line('administration');
    $this->data['title'] = $this->lang->line('groups') . ' - ' . $this->lang->
    >line('administration');

    $this->data['subviews'] = array(
        array('subview' => $this->parser->parse('administration/groups/form', $this->data,
TRUE))
    );

    $this->add_asset('css_page', 'assets/plugins/select2/css/select2.min.css');
    $this->add_asset('css_page', 'assets/plugins/DataTables/media/css/DT_bootstrap.css');

    $this->add_asset('js_plugins', 'assets/plugins/bootbox/bootbox.js');
    $this->add_asset('js_plugins', 'assets/plugins/select2/js/select2.min.js');
    $this->add_asset('js_plugins', 'assets/plugins/DataTables/media/js/jquery.dataTables.js');
    $this->add_asset('js_page', 'assets/js/administration/groups.js');

    $options = array('hasAction' => can('group_write'));
    $optionsJ = json_encode($options);
    $table_init = 'Groups.table(' . $optionsJ . ')';
    $this->data['js_class_init'] = array(
        array('func' => $table_init)
    );

    $this->render('administration/groups/content');
}
```

Figura 35 - administrations.php, function groups

La funzione Groups.Table, in assets/js/, della quale si mostrano le funzioni principali nelle figure successive, effettua le seguenti operazioni:

- inizializza le colonne della tabella (Figura 36)

```
var columns = [
    { "data": "id" },
    { "data": "name" },
    { "data": "description" },
    { "data": "profile_name" },
];

var targetcolumnDefs = false;

if (options.hasAction == true) {
    var action = { "data": "action" };
    columns.push(action);
    targetcolumnDefs = -1;
}
```

Figura 36 - inizializzazione colonne

- Crea la tabella invocando la funzione `group_cross_profile` (Figura 1Figura 38) nel controller `Groups.php` (Figura 37) in `applications/controllers/api/`

```

$('#table_1')
    .on('error.dt', function ( e, settings, techNote, message ) {
    })
    .DataTable( {
        "columnDefs": [
            {
                "render": function ( data, type, row ) {

                    if (options.hasAction == true) {
                        var buttonEdit      = '<a href="javascript:;" data-
id="'+row.id+'" class="btn_edit btn btn-success btn-xs"><i class="fa fa-edit"></i></a>';
                        var buttonRemove    = '<a href="javascript:;" data-
id="'+row.id+'" class="btn_remove btn btn-danger btn-xs"><i class="fa fa-times fa fa-white"></i></a>';
                        var buttons = buttonEdit+' '+buttonRemove;
                        return buttons;
                    }
                },
                "targets": targetcolumnDefs
            },
        ],
        "ajax": Main.handleBaseUrl()+"/api/group/group_cross_profile",
        "columns": columns,
        "drawCallback": function( settings ) {
            $('#.btn_edit').click(function(){
                var id = $(this).attr('data-id');

                openForm(id);
            });
            $('#.btn_remove').click(function(){
                var id = $(this).attr('data-id');
                confirmRemove(id);
            });
        }
    });

```

Figura 37 - Groups.php

```

public function group_cross_profile_get () {
    // carico i dati
    $result = $this->group_model->get_group_cross_profile();
    $data = ['data' => $result];
    $this->response($data, REST_Controller::HTTP_OK);
}

```

Figura 38 - funzione `group_cross_profile`

che carica i dati da `applications/models/Group_model.php` effettuando una query, in Figura 39, sulla vista `"v_r_group_profile"`.

```

public function group_cross_profile_get () {
    // carico i dati
    $result = $this->group_model->get_group_cross_profile();
    $data = ['data' => $result];
    $this->response($data, REST_Controller::HTTP_OK);
}

```

Figura 39 – restituzione del risultato della query

La vista `v_r_group_profile` è definita come:

```
SELECT g.id AS id,g.name AS name, g.description AS description, p.name AS profile_name
FROM ((group g
      LEFT JOIN r_group_profile gp ON ((g.id = gp.group_id)))
      LEFT JOIN profile p ON ((p.id = gp.profile_id)))
```

E associa il gruppo al suo profilo di visibilità.

Il controller effettua il rendering del content che viene inviato al layout e visualizzato nella sezione content della view.

Attraverso le apposite funzioni javascript è possibile;

- creare nuovi gruppi associando i ruoli e modificare i gruppi esistenti attraverso la funzione `js openForm` (in Figura 40), che attraverso la chiamata al controller `Groups.php` effettua le operazioni di salvataggio chiamando il model;
- eliminare i gruppi.

```
var openForm = function(id) {
    var form = $('#form form');
    $.subview({
        content: "#form",
        startFrom: "right",
        onShow: function() {
            //EDIT
            if(id){
                $.ajax({
                    type: "GET",
                    url:
                    dataType : 'json',
                    success: function(response){
                        response.profile_id =
                        form.find('.form-title').text('Modifica
                        gruppo '+response.name);
                        $.each(response, function( key, value )
                        {
                            form.find('#'+key).val(value);
                        });
                        $.each(response.role, function( key,
                        value ){
                            $('#role_'+value).iCheck('check');
                        });
                        function(jqXHR,text_status,error_thrown){
                            complete: function(){
                                //CREATE
                                else{
                                    form.find('input[type="text"], textarea').val('');
                                    form.find('.form-title').text('Crea nuovo gruppo');
                                    handleForm();
                                }
                            }
                        });
                    }
                });
            }
        }
    });
};
```

```
onClose: function() {  
  function(result) {  
    bootbox.confirm("Are you sure you want to close subview?",  
      if(result) {  
        $.hideSubview();  
      };  
    });  
  }  
});  
};
```

Figura 40 – funzione open form

Le sezioni relative a ruoli e permessi sono oggetto di verifica, analisi e riorganizzazione.

2.5 Struttura dell'applicazione: la scheda censimento

Nel contesto dei processi gestionali che si occupano di gestire e mantenere gli impianti di illuminazione pubblica, l'esigenza che il sistema deve soddisfare è quella di censire in modo accurato e dettagliato un impianto di illuminazione per estrarre informazioni che siano utili al monitoraggio ed alla conoscenza del processo gestionale delle infrastrutture e dei servizi di pubblica illuminazione.

Al fine di monitorare e valutare l'efficienza di funzionamento di un impianto si rende prima necessario conoscere la sua infrastruttura completa attraverso la compilazione corretta e dettagliata di una "Scheda Censimento" che presenti tutti i suoi dati.

La Scheda Censimento definisce l'intera struttura dell'impianto di illuminazione pubblica, gestita direttamente o tramite concessione dagli enti pubblici locali; essa tiene conto:

- del contesto geografico/territoriale in cui esso è collocato (attraverso la definizione di zone omogenee comprendenti vie e piazze pubbliche, luoghi pubblici in genere e aree di pertinenza);
- del contesto tecnico e tecnologico (dati relativi ai POD, ai quadri elettrici, agli apparecchi collegati ed anche alle sorgenti luminose utilizzate).

I dati che vengono storicizzati nell'applicazione descrivono completamente il pod nella sua locazione geografica (Comune, zone stradali) ma anche nella sua gestione (pod, dati del quadro elettrico, apparecchi, sorgenti luminose); questi dettagli sono molto importanti perché permettono di poter studiare la riqualificazione di un impianto alla luce delle sue caratteristiche tecniche e geografiche.

I dati elettrici di un POD rappresentano un punto chiave nella gestione degli impianti; infatti la manutenzione di un quadro elettrico, oppure delle strutture di sostegno di un punto luce, od ancora lo stesso punto luce, sono dei costi non indifferenti per una amministrazione pubblica, pertanto, ai fini di una riqualificazione energetica completa, oppure anche solo di un risparmio di risorse, è importante conoscerne l'entità.

La compilazione della scheda censimento è quindi un punto molto importante della fase statica, in quanto la correttezza e la completezza dei dati sono la base per qualsiasi valutazione o riqualificazione di ogni impianto.

La scheda censimento, e dunque la definizione dell'impianto, ha una struttura ad albero; questo è fondamentale per definire tutte le caratteristiche di un impianto, organizzarle anche visivamente (semplificando una eventuale modifica o aggiornamento dei dati), ed infine renderle disponibili ad algoritmi di calcolo e/o di valutazione delle prestazioni illuminotecniche o finanziarie.

Ai fini di garantire una visibilità immediata della struttura, data la complessità del form che prevede varie schermate, nell'applicazione è sempre visibile in un box la struttura di un impianto (Figura 41**Errore. L'origine riferimento non è stata trovata.**), che può essere espansa o ridotta dall'operatore, facilitando l'accesso dello stesso ad ogni sezione della scheda censimento.

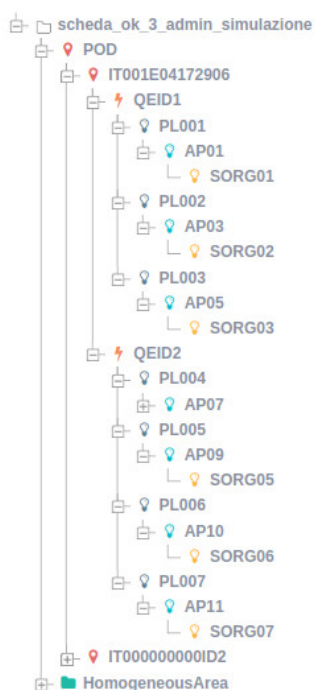


Figura 41 - struttura ad albero della scheda censimento

Cliccando sul singolo dettaglio, è possibile visualizzare direttamente la scheda relativa alla sezione selezionata, facilitando l'operatore anche nell'accesso ai dati che devono essere ricercati.

La lista delle schede censimento (Figura 42) fa riferimento al controller in applications/controllers/enea/Scheda_censimento.php che estende il My_Controller.php.

ELENCO SCHEDE CENSIMENTO

Id	Descrizione	codice istat comune	nome comune	N. POD	N. quadri	N. punti luce	Ultima modifica	Sottomissione	Azioni
7	Test_caricamento	12058091	Roma	1	1	1	02/08/2019 11:25:44	02/08/2019 11:26:20	C A D E
6	Prova Linee Guida 01	12058091	Cesano di Roma	2	3	2	26/06/2019 16:24:09		C A D E
2	scheda_12058003_201906121239	120580912	Cesano di Roma	2	3	10	26/06/2019 11:50:39		C A D E

Show 10 entries

Previous
1
Next

CARICA XML

Attenzione

- Sono supportati i formati .xml e .zip, si consiglia l'utilizzo del secondo per file superiori ai 10MB.
- Il processo di upload può richiedere dei minuti, non chiudere la pagina finché non è completo il caricamento.

Descrizione

Figura 42 – pagina elenco schede censimento e caricamento xml

La funzione `index()` (Figura 43) carica i javascript e i css per la pagina, e carica anche tutti i comuni su cui l'utente è abilitato.

Infatti, ogni utente ha accesso esclusivamente alle schede di sua competenza. Questa ricerca viene effettuata tramite ricerca all'interno della sessione.

```
public function index()
{
    $this->data['breadcrumbs'] = array();
    $this->data['page_title'] = $this->lang->line('scheda_censimento');
    $this->data['page_subtitle'] = '';
    $this->add_asset('css_page', 'assets/css/administration-menu.css');
    $this->add_asset('css_page', 'assets/plugins/toastr/toastr.min.css');
    $this->add_asset('css_page', 'assets/plugins/sweetalert/lib/sweet-
alert.css');
    $this->add_asset('css_page', 'assets/plugins/select2/css/select2.min.css');
    $this->add_asset('css_page',
'assets/plugins/DataTables/media/css/DT_bootstrap.css');

    $this->add_asset('js_plugins', 'assets/plugins/toastr/toastr.js');
    $this->add_asset('js_plugins', 'assets/plugins/sweetalert/lib/sweet-
alert.min.js');
    $this->add_asset('js_plugins', 'assets/plugins/bootbox/bootbox.js');
    $this->add_asset('js_plugins', 'assets/plugins/select2/js/select2.min.js');
    $this->add_asset('js_plugins',
'assets/plugins/DataTables/media/js/jquery.dataTables.js');
    $this->add_asset('js_plugins', 'assets/plugins/DataTables/extensions/date-
time-html-v2.js');

    $user_locations = $this->session->location;
    if(is_array($user_locations) && count($user_locations) > 0){
        $user_region = reset($user_locations);
        $district = reset($user_region['districts']);
        $this->data['default_label_value'] =
'scheda_'. $district['code']. '_'.date('YmdHi');
    }
    else{
        $this->data['default_label_value'] = 'scheda_'. '_'.date('YmdHis');
    }

    $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/schede-
manager.js');
    $this->add_init('SchedeManager.init()');
    $this->add_init('SchedeManager.dataTable()');

    $this->render('enea/scheda_censimento/content');
}
```

Figura 43 – funzione `index` in `Scheda_censimento.php`

Una volta caricate le città su cui è abilitato l'operatore, il file `Schede-manager.js` in `assets/js/enea/scheda_censimento/` viene caricato.

Il javascript permette alla view di visualizzare la lista di schede censimento, attraverso una chiamata ajax alla funzione `list` (Figura 44) del controller `Scheda.php`, in `/api/enea/schedaCensimento/`

```

public function list_get()
{
    $result = $this->CensusTechSheet_model->get_list();
    $data = array('data' => $result);
    $this->response($data, REST_Controller::HTTP_OK); // OK (200)
}

```

Figura 44 – scheda.php funzione list

La funzione list_get() invoca la funzione get_list() del model CensusTechSheet_model.php in application/models/enea/scheda_censimento/, che effettua il controllo di visibilità delle schede (Figura 45).

In particolare:

- recupera da database i codici istat relativi ai comuni associati all'utente;
- dalle tabelle recupera tutte le schede;
- per ciascuna scheda, se il codice istat è tra i codici a cui l'utente afferisce, estrae i dati da inserire nella tabella.

```

public function get_list()
{
    $user_istat_codes = $this->get_user_istat_code();
    $ret = array();
    $user_id = $this->session->userdata('id');
    $schede = $this->db->select('sc_CensusTechSheet.*')
        ->select('sc_PLSystemGeneralData.id AS sc_PLSystemGeneralData_id,
sc_PLSystemGeneralData.ISTATCode AS an_ip_ist')
        ->from('sc_CensusTechSheet')
        ->join('sc_PLSystemGeneralData', 'sc_PLSystemGeneralData.sc_CensusTechSheet_id =
sc_CensusTechSheet.id')
        ->where('sc_CensusTechSheet.user_id', $user_id)
        ->get()->result();
    foreach ($schede as $sc) {
        $id = $sc->id;
        $label = $sc->label;
        $an_ip_ist = $sc->an_ip_ist;
        if (is_admin() || in_array($an_ip_ist, $user_istat_codes)) {
            $city = $this->db->get_where('sc_City', array('sc_PLSystemGeneralData_id' =>
$sc->sc_PLSystemGeneralData_id))->row();
            $an_ip_nom = $city->City;
            $n_pod = $this->db->where('sc_CensusTechSheet_id', $id)->get('sc_POD')-
>num_rows();
            $n_qe = $this->db->where('sc_CensusTechSheet_id', $id)->get('sc_ElectricPanel')-
>num_rows();
            $n_pl = $this->db->where('sc_CensusTechSheet_id', $id)->get('sc_LightSpot')-
>num_rows();

            $ret[] = array(
                'id' => $id,
                'label' => $label,
                'an_ip_ist' => $an_ip_ist,
                'an_ip_nom' => $an_ip_nom,
                'n_pod' => $n_pod,
                'n_qe' => $n_qe,
                'n_pl' => $n_pl,
                'updated_at' => $sc->updated_at ? date("d/m/Y H:i:s", strtotime($sc-
>updated_at)) : ($sc->created_at ? date("d/m/Y H:i:s", strtotime($sc->created_at)) : ''),
                'submitted' => $sc->submitted ? date("d/m/Y H:i:s", strtotime($sc-
>submitted)) : '',
                'action' => '',
            );
        }
    }
    return $ret;
}

```

Figura 45 –funzione getList

I dati visualizzati sono:

- id
- label
- codice istat
- comune
- numero di pod
- numero di quadri elettrici
- data aggiornamento
- data di submit

Id	Descrizione	codice istat comune	nome comune	N. POD	N. quadri	N. punti luce	Ultima modifica	Sottomissione	Azioni
7	Test_caricamento	12058091	Roma	1	1	1	02/08/2019 11:25:44	02/08/2019 11:26:20	

Figura 46 – dati della scheda censimento visualizzati

ed è possibile visualizzarli in (Figura 46).

Per ciascuna scheda sono presenti dei pulsanti di azione:

- **modifica:** invoca la funzione `form()` del controller(`applications/controllers/enea/Scheda_censimento.php`), passando l'id della scheda. La funzione carica tutti i javascript e inizializza la scheda censimento.

```
public function form($id = 0, $ts = 0)
{
    $this->data['breadcrumbs'] = array();
    $this->data['page_title'] = 'Scheda Censimento';
    $this->data['page_subtitle'] = '';

    if ($this->config->item('static_form')) {
        $this->add_asset('css_plugins', 'assets/plugins/jstree/dist/themes/default/style.css');
        $this->add_asset('js_plugins', 'assets/plugins/jstree/dist/jstree.min.js');
        $this->add_asset('css_plugins', 'assets/plugins/datepicker/css/datepicker.css');
        $this->add_asset('js_plugins', 'assets/plugins/bootstrap-datepicker/js/bootstrap-datepicker.js');
        $this->add_asset('js_page', 'assets/plugins/bootstrap-datepicker/js/locales/bootstrap-datepicker.it.js');
        $this->add_asset('js_plugins', 'assets/plugins/jquery.maskedinput/src/jquery.maskedinput.js');
        // Page scripts
        $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/spin-manager.js');
        $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/scheda-static-manager.js');
        $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/CensusTechSheet-manager.js');
        $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/POD-manager.js');
        $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/ElectricPanel-manager.js');
        $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/LightSpot-manager.js');
        $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/LightSpotDevice-manager.js');
        $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/LightSource-manager.js');
        $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/HomogeneousArea-manager.js');
        $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/HomogeneousAreaExt-manager.js');

        $this->data['cs_censusTechSheet_id'] = $id;
        $this->add_init("SchedaManager.init()");
        $this->add_init("CensusTechSheetManager.init()");
    }
}
```

```

else {
    $this->add_asset('css_plugins', 'assets/plugins/jstree/dist/themes/default/style.css');
    $this->add_asset('css_plugins', 'assets/plugins/sweetalert2/sweetalert2.css');
    $this->add_asset('css_plugins', 'assets/css/enea/scheda_censimento/form.css');
    $this->add_asset('js_plugins', 'assets/plugins/jstree/dist/jstree.js');
    $this->add_asset('css_plugins', 'assets/plugins/datepicker/css/datepicker.css');
    $this->add_asset('css_plugins', 'assets/plugins/leaflet2/leaflet.css');
    $this->add_asset('js_plugins', 'assets/plugins/bootstrap-datepicker/js/bootstrap-
datepicker.js');
    $this->add_asset('js_page', 'assets/plugins/bootstrap-datepicker/js/locales/bootstrap-
datepicker.it.js');
    $this->add_asset('js_plugins',
'assets/plugins/jquery.maskedinput/src/jquery.maskedinput.js');
    $this->add_asset('js_plugins', 'assets/plugins/sweetalert2/sweetalert2.all.min.js');
    $this->add_asset('js_plugins', 'assets/plugins/leaflet2/leaflet.js');

    // Page scripts
    $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/spin-manager.js');
    $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/scheda-manager.js');
    $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/CensusTechSheet-
manager.js');
    $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/POD-manager.js');
    $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/ElectricPanel-
manager.js');
    $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/LightSpot-manager.js');
    $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/LightSpotDevice-
manager.js');
    $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/LightSource-manager.js');
    $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/HomogeneousArea-
manager.js');
    $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/HomogeneousAreaExt-
manager.js');

    $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/CensusTechSheet-form.js');
    $this->add_asset('js_page', 'assets/js/enea/utilities/ajaxloader.js');
    $this->add_asset('js_page', 'assets/js/enea/utilities/error.js');
    $this->add_asset('js_page', 'assets/js/enea/utilities/modal.js');
    $this->add_asset('js_page', 'assets/js/enea/utilities/normalizer.js');

    $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/CensusTechSheet-form.js');

    $this->data['cs_censusTechSheet_id'] = $id;

    $parametricForm = $this->makeFormParametric("fieldList.csv");
    $this->data['parametricForm'] = $parametricForm;

    $this->add_init("SchedaManager.init()");
    $this->add_init("SchedaCensimento.initForm()");
}
$this->render('enea/scheda_censimento/form/layout');
}

```

Figura 47 – funzione form per il caricamento della scheda censimento

Nella funzione viene chiamato il metodo makeFormParametric (Figura 48), a cui viene passato un file csv.

```

public function makeFormParametric($fieldListName) {

    $this->load->helper("url");
    $fields = $this->process_csv($fieldListName);
    $fieldsList = $this->makeFields($fields);
    $formsFile = file_get_contents(base_url()."assets/data/scheda_censimento/formSchema.json");
    $formsStruct = json_decode($formsFile, true);

    $html = '';

```

```

for ($i=0; $i < count($formsStruct) ; $i++) {
    $form = $formsStruct[$i];
    $html .= '<div id="' . $form['idForm'] . '_model' class="forms_model">';
    $html .= '<div class="panel panel-white">';
    // head
    $html .= '<div class="panel-heading border-light">';
    $html .= '<div class="row">';
    $html .= '<div class="col-md-7">';
    $html .= '<h4 class="panel-title">' . $form['labelForm'] . ' <span class="current_id_box"><span
class="current_id"></span></span></h4>';
    $html .= '</div>';
    $html .= '<div class="col-md-5">';

    $addButtonsByForm= array(
        'anagraficaForm' => array('podForm', 'zonaOmogeneaForm'),
        'podForm' => array('quadroElettricoForm'),
        'quadroElettricoForm' => array('puntoLuceInstallazioneForm'),
        'puntoLuceInstallazioneForm' => array('puntoLuceApparecchioForm'),
        'puntoLuceApparecchioForm' => array('puntoLuceSorgenteLuminosaForm'),
        'puntoLuceSorgenteLuminosaForm' => array(),
        'zonaOmogeneaForm' => array('zonaOmogeneaExtForm'),
        'zonaOmogeneaExtForm' => array()
    );
    $linkButtonsIcons = array(
        'podForm' => array('fa-map-marker', 'danger'),
        'quadroElettricoForm' => array('fa-bolt', 'warning'),
        'puntoLuceInstallazioneForm' => array('fa-lightbulb-o', 'primary'),
        'puntoLuceApparecchioForm' => array('fa-lightbulb-o', 'info'),
        'puntoLuceSorgenteLuminosaForm' => array('fa-lightbulb-o', 'warning'),
        'zonaOmogeneaForm' => array('fa-globe', 'dark-azure'),
        'zonaOmogeneaExtForm' => array('fa-flag', 'dark-orange')
    );
    $linkButtonsLabels = array(
        'podForm' => 'POD',
        'quadroElettricoForm' => 'Quadro Elettrico',
        'puntoLuceInstallazioneForm' => 'Punto Luce',
        'puntoLuceApparecchioForm' => 'Apparecchio',
        'puntoLuceSorgenteLuminosaForm' => 'Sorgente Luminosa',
        'zonaOmogeneaForm' => 'Zona Omogenea',
        'zonaOmogeneaExtForm' => 'Estensione Zona Omogenea'
    );
    $listButtons = $addButtonsByForm[$form['idForm']];
    $numButtons = count($listButtons);

    if ($form['idForm'] != 'anagraficaForm') {
        $html .= '<button id="btn_clone_form_' . $form['idForm'] . '" data-toggle="tooltip" data-
placement="top" title="Clona ' . $linkButtonsLabels[$form['idForm']] . '" data-target="' . $form['idForm'] . '"';
        $html .= 'class="clone_form btn btn-sm btn-' . $linkButtonsIcons[$form['idForm']][1] . '" type="button"
>';
        $html .= '<i class="fa fa-files-o"></i> <i class="fa
' . $linkButtonsIcons[$form['idForm']][0] . '"></i> </button>';
    }

    for ($b=0; $b<$numButtons; $b++) {
        $iconButton = $linkButtonsIcons[$listButtons[$b]];
        $icon = $iconButton[0];
        $color = $iconButton[1];

        $html .= '<button id="btn_new_form_' . $form['idForm'] . '" data-toggle="tooltip" data-placement="top"
title="Aggiungi ' . $linkButtonsLabels[$listButtons[$b]] . '" data-target="' . $listButtons[$b] . '" class="new_form btn
btn-sm btn-' . $color . '" type="button" >';

        $html .= '<i class="fa fa-plus"></i> <i class="fa ' . $icon . '"></i> </button>';
    }
    $html .= '<button data-toggle="tooltip" data-placement="top" title="Salva" data-target="' . $form['idForm'] . '" data-
real-send="1" class="save_form btn btn-sm btn-success" type="button">';
    $html .= '<i class="fa fa-save"></i></button>';

    $html .= '<button data-toggle="tooltip" data-placement="top" title="Check Scheda" data-
target="' . $form['idForm'] . '" data-real-send="0" class="validate_form btn btn-sm btn-warning" type="button" >';
    $html .= '<i class="fa fa-check"></i></button>';
}

```

```

        if ($form['idForm'] != 'anagraficaForm') {
            $html .= '<button id="'. $form['idForm']. '_delete' data-toggle="tooltip" data-placement="top"
            title="Elimina" data-target="'. $form['idForm']. '" class="delete_form btn btn-sm btn-red" type="button">';
            $html .= '<i class="fa fa-trash"></i></button>';
        }

        $html .= '<button data-toggle="tooltip" data-placement="top" title="Info Generali" data-
        target="'. $form['idForm']. '" class="info_modal btn btn-sm btn-azure" type="button">';
        $html .= '<i class="fa fa-info"></i></button>';

        $html .= '</div>';
        $html .= '</div>';

        $html .= '<div class="row cloning-row">';
        $html .= '<div class="col-md-12">Form in stato di clonazione</div>';
        $html .= '</div>';

        $html .= '</div>';

// body no TAB
if (count($form['tabs']) == 1) {
    $tab = $form['tabs'][0];
    $html .= '<div class="panel-body">';
    $html .= '<form id="'. $tab['idTab']. '" class="formValidable">';
    $html .= $this->populateForm($tab['rows'],$fieldsList);
    $html .= '</form>';
    $html .= '</div>';
}

// body TAB
else {
    $html .= '<div class="panel-body no-padding">';
    $html .= '<div class="tabbable no-padding no-margin">';

// tab index
$html .= '<ul class="nav nav-tabs nav-justified">';
for ($j=0; $j < count($form['tabs']); $j++) {
    $tab = $form['tabs'][$j];
    $isTabActive = '';
    if ($j == 0) {$isTabActive = 'active';}
    $html .= '<li class="'. $isTabActive. '">';
    $html .= '<a id="panel_'. $tab['idTab']. '_link' data-toggle="tab'
href="#panel_'. $tab['idTab']. '">';
    $html .= $tab['labelTab'];
    $html .= '</a>';
    $html .= '</li>';
}
$html .= '</ul>';

// tab content
$html .= '<div class="tab-content">';

for ($j=0; $j < count($form['tabs']); $j++) {
    $isTabActive = '';
    $tab = $form['tabs'][$j];
    if ($j == 0) {$isTabActive = 'fade in active';}
    $html .= '<div id="panel_'. $tab['idTab']. '" class="tab-pane '. $isTabActive. '">';
    $html .= '<form id="'. $tab['idTab']. '" class="formValidable">';
    $html .= $this->populateForm($tab['rows'],$fieldsList);
    $html .= '</form>';
    $html .= '</div>';
}
$html .= '</div>';

$html .= '</div>';
$html .= '</div>';
}

```

Figura 48 – funzione makeformParametric

Questa funzione, come visibile in (Figura 48), è molto complessa ed è stata sviluppata per gestire in modo dinamico il form relativo alla scheda censimento.

Infatti, la scheda censimento è il cuore di tutta l'applicazione

- perché permette di storicizzare tutti i dati degli impianti;
- perché sulla scheda censimento si basano tutte le funzionalità del PELL.

La scheda però è in continua evoluzione, infatti è possibile che dati vengano aggiunti, modificati o eliminati in modo agevole senza compromettere lo storico degli impianti già inseriti e senza impattare in modo invadente tutta l'applicazione.

Per questa esigenza il form è stato sviluppato in modo parametrico, in modo che fosse possibile aggiungere dei campi in modo agevole e semplice senza troppi impatti sullo storico e sull'applicazione.

Il metodo utilizza per la creazione del form due files:

- un file fieldList.csv presente in assets/data/scheda censimento/ che viene passato dal controller al metodo
- un file formSchema.json presente in assets/data/scheda censimento/ che viene caricato dal metodo

Il concetto di base è gestire il form della scheda censimento come una tabella in cui siano presenti delle righe e delle colonne; come si vede dalla Figura 49 per ogni sezione sono presenti una o più "Tab":

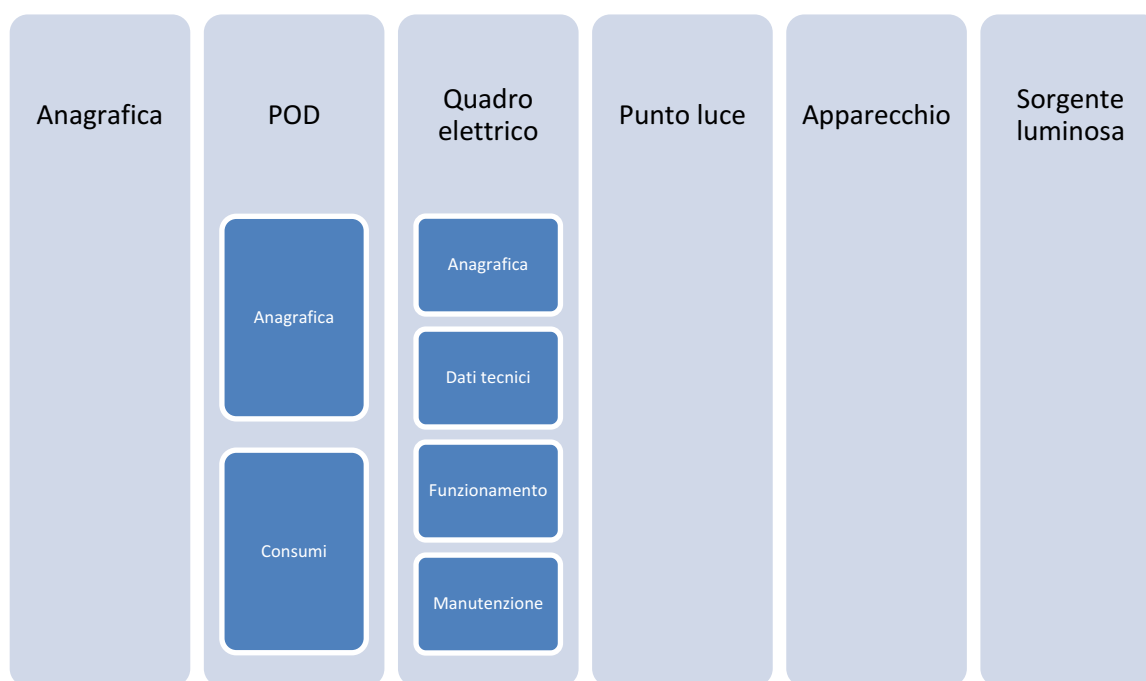


Figura 49 – struttura delle tabs della scheda censimento

Ogni tab prevede dei campi specifici; se vediamo ad esempio la Figura 50 **Errore. L'origine riferimento non è stata trovata.**, cioè il form relativo all'anagrafica dell'impianto che si sta inserendo, vediamo che presenta:

- una label;
- nome dell'impianto, tipologia e tipo scheda;
- Comune, codice ISTAT e la Regione;
- la sua estensione e il numero di abitanti;
- Il nome del responsabile comunale che si occupa dell'anagrafica degli impianti di illuminazione pubblica e il gestore;

- riepilogo quantitativo dei punti luce presenti nell'impianto, totali e di proprietà e flag se l'impianto è a forfait o no;
- la label "Metadati Anagrafica";
- inizio e fine validità del dato;

Ogni punto indicato corrisponde ad una riga della tabella, i singoli campi sono le colonne; pertanto la scheda anagrafica presenta

- 17 righe, ogni riga presenta 1-3 colonne.

ANAGRAFICA ILLUMINAZIONE PUBBLICA

+?
+?
+
✓
i

Descrizione * i scheda_ok_3_admin_simulazione		Tipologia * i simulazione	Scheda corrente * i ☒ No ☐ Si
Nome comune * i Cesano di Roma, Cesano di Rome	Codice istat comune * i 12058091	Regione * i Lazio	
N. di abitanti * i # 10001	Superficie * i km2 183,5		
Responsabile comunale * i Mario Rossi		Nome gestore * i Pippo	
Gestione impianto a forfait * i ☒ No ☐ Si	N. punti luce totali * i # 10	N. punti luce di proprietà * # 8	

Metadati Anagrafica

Inizio validità del dato 03/05/2019	Fine validità del dato 02/06/2019
Fonte del dato * rilievo diretto	Scala * i scala 1:10000

Polygon Anagrafica

Identificatore Superficie i POL2	Sistema di riferimento * i + http://www.opengis.net/def/crs/EPSSG/0/4936
Lista coordinate * i + 45.256 -110.45 46.46 -109.48 43.84 -109.86 45.256 -110.45	Dimensione coordinate i 2=xy

Area

Tipo di estensione * i altro	Nome/codice * i Area Test	Scala * i scala 1:10000
---	--	--

Polygon Area

Identificatore Superficie i POL1	Sistema di riferimento * i + http://www.opengis.net/def/crs/EPSSG/0/4936
Lista coordinate * i + 45.256 -110.45 46.46 -109.48 43.84 -109.86 45.256 -110.45	Dimensione coordinate i 2=xy

Figura 50 - anagrafica

Se vediamo tutte le tab della scheda censimento, possiamo dire che:

La scheda POD, costituita da due tabs presenta:

- Anagrafica: 12 righe e 1-3 colonne;
- Consumi: 10 righe e 1-3 colonne;

POD

Anagrafica

Consumi

Codice pod *

IT001E04172906

Toponimo stradale pod *

Via Roma

Data del rilievo *

23/03/2019

Stato di conservazione del manufatto contenente il pod

buono/adeguato

Promiscuità *

No

Quadri elettrici

N. quadri elettrici *

2

N. quadri da sostituire

1

N. quadri da ricondizionare

1

Posizione

Identificatore Superficie

PT01

Latitudine Longitudine

-30.7 134.1

Sistema di riferimento *

http://www.opengis.net/def/crs/EPSTG/0/4936

Dimensione coordinate

2=xy

Metadati di istanza

Iniziativa Metadati Madre

Inizio validità del dato

03/05/2019

Fine validità del dato

02/06/2019

Fonte del dato *

rilevo diretto

Scala *

scala 1:10000

POD

Anagrafica

Consumi

Anno di riferimento consumi

2018

Denominazione del distributore *

Ena spa

Consumo annuale (anno precedente) *

kWh/anno 302

Consumo annuale (anno attuale) *

kWh/anno 200

Costo annuale (anno precedente) per la sola fornitura di energia elettrica per pubblica illuminazione *

4000

Costo annuale (anno attuale) per la sola fornitura di energia elettrica per pubblica illuminazione *

2000

Potenza installata

kw 0,8

Potenza contrattuale impegnata *

kw 1,2

Cosφ? tipico *

0,9

Metadati di istanza

Iniziativa Metadati Madre

Inizio validità del dato

03/05/2019

Fine validità del dato

02/06/2019

Fonte del dato *

rilevo diretto

Scala *

scala 1:10000

Figura 51 – tabs relative al pod

La scheda quadro elettrico costituita da 4 tabs presenta:

- Anagrafica: 17 righe e 1-3 colonne;
- Dati tecnici: 11 righe e 1-3 colonne;
- Funzionamento: 15 righe e 1-3 colonne;
- Manutenzione: 11 righe e 1-3 colonne;

La scheda punto luce presenta:

- Anagrafica: 13 righe e 1-3 colonne;

La scheda apparecchio presenta:

- Anagrafica: 11 righe e 1-3 colonne;

La scheda sorgente luminosa presenta:

- Anagrafica: 9 righe e 1-3 colonne;

QUADRO ELETTRICO

Anagrafica	Dati Tecnici	Funzionamento	Manutenzione
Codice quadro elettrico *			
QEID1			
Data del rilievo quadro elettrico *		Quadro misto *	
22/03/2019		☒ No ☒ Si	
Posizione			
Identificatore Superficie		Sistema di riferimento *	
PT05		http://www.opengis.net/def/crs/EPSG/0/4936	
Latitudine Longitudine *		Dimensione coordinate	
-30.7 134.5		2=xy	
Anno di costruzione	Tipologia accessibilità *	Chiusura a chiave dell'involucro	
1970	interna	☐ No ☒ Si	
Potenza installata *			
kW 0,2			
Stato			
di conservazione quadro *	di conservazione armadio	di funzionamento *	impianto elettrico da riqualificare *
da sostituire	accettabile	non in esercizio	sufficiente
Punti luce			
N. totale di punti luce *	N. di punti luce di proprietà *	N. di punti luce non di proprietà *	
# 3	# 2	# 1	
Carichi esogeni elettrici *		Sostegni da riqualificare *	
elettrico temporaneo		insufficiente	
N. linee aeree *	% di linee aeree sul totale delle linee *	Numero sostegni *	
# 1	50	3	
Metadati Anagrafica			
Inizio validità del dato		Fine validità del dato	
03/05/2019		02/06/2019	
Fonte del dato *	Scala *		

Figura 52 – tabs relative al quadro elettrico -anagrafica

QUADRO ELETTRICO

Anagrafica **Dati Tecnici** **Funzionamento** **Manutenzione**

N. di fasi * **i**
corrente continua

Tipo di protezione generale * **i**
entrambi gli interruttori

N. sottoquadri
2

Indice ipei* (cam 2018) * **i**
Non applicabile: valore previsto dalla specifica ma non applicabile all'istanza (ad es. non e' applicabile la categ

Presenza di trasformatore di tensione in cabina elettrica **i**
☒ No ☐ Si

Tipo di impianto
in derivazione

Presenza di spd **i**
☒ No ☐ Si

Metadati Dati Tecnici **i** **inietta Metadati Madre**

Inizio validità del dato
03/05/2019

Fonte del dato *
ritratto diretto

N. circuiti in uscita dal quadro elettrico *
2

Tensione nominale del quadro elettrico * **i**
V 230

Indice ipei (cam 2013) * **i**
Non applicabile: valore previsto dalla specifica ma non applicabile all'istanza (ad es. non e' applicabile la categ

Tipo di linea * **i**
aerea

Potenza del trasformatore **i**
kVA

Presenza impianto in media tensione **i**
☒ No ☐ Si

Presenza di impianto di terra **i**
☒ No ☐ Si

Fine validità del dato
02/06/2019

Scala * **i**
scala 1:10000

Figura 53– tabs relative al quadro elettrico – dati tecnici

QUADRO ELETTRICO

Anagrafica **Dati Tecnici** **Funzionamento** **Manutenzione**

Tipo di accensione * **i**
Telecontrollo

Parzializzazione

Parzializzazione accensione tutta notte - mezza notte * **i**
☒ No ☐ Si

Flusso luminoso

Riduzione del flusso luminoso **i**
☐ No ☒ Si

Valore medio di riduzione del flusso luminoso **i**
% 50

% di riduzione della potenza **i**
%

Telegestione

Sistema di telegestione (gestione real-time) * **i**
☐ No ☒ Si

Sistema di telediagnosi remota * **i**
☐ No ☒ Si

Id meter **i**

Classe del meter
Nessuna selezione

Metadati Funzionamento e Gestione **i** **inietta Metadati Madre**

Inizio validità del dato
03/05/2019

Fonte del dato *
ritratto diretto

Ore di accensione dell'impianto (ore/anno) * **i**
ore/anno 4000

Durata accensione parzializzata (ore/anno) **i**
ore/anno

Durata di riduzione del flusso luminoso **i**
ore/anno 2500

Tipo regolazione del flusso luminoso * **i**
Punto a punto

% di riduzione media della potenza **i**
% 50

Telecontrollo su quadro elettrico * **i**
adattivo

Fine validità del dato
02/06/2019

Scala * **i**
scala 1:10000

Figura 54 – tabs relative al quadro elettrico - funzionamento

QUADRO ELETTRICO

Anagrafica

Dati Tecnici

Funzionamento

Manutenzione

Anno di riferimento manutenzione *i*
1990

Denominazione affidatario * *i*
Pippo 4

Approvvigionamento materiali *i*
☐ No ☐ Si

Voce di spesa *i*

Attività *i*

Tipologia di contratto * *i*
servizio manutenzione semplice

Remunerazione *i*
A SAL

Impianti di proprietà

Anno precedente *i*
€/anno 15000

Anno attuale *i*
€/anno 8000

Impianti non di proprietà

Anno precedente *i*
€/anno 5000

Anno attuale *i*
€/anno 2000

Importi manutenzione straordinaria annua (€/Anno iva esclusa)

Anno precedente *i*
€/anno 4000

Anno attuale *i*
€/anno 3000

Metadati Manutenzione *i* [Inietta Metadati Madre](#)

Inizio validità del dato
03/05/2019

Fonte del dato *
rilievo diretto

Fine validità del dato
02/06/2019

Scala * *i*
scala 1:10000

Figura 55 - tabs relative al quadro elettrico -manutenzione

PUNTO LUCE

Le modifiche ai campi contrassegnati dal simbolo *i* saranno replicate a tutti gli elementi dello stesso livello

Id punto luce * *i*
PL001

Id zona omogenea * *i*
via roma

Tipologia di Installazione * *i*
su braccio

Altezza apparecchio * *i*
m 6

Inclinazione *i*
degree

Distanza sostegno inizio della carreggiata *i*
m 3

Lunghezza braccio (m) *i*
m 4

Posizione

Identificatore Superficie *i*
PTLS1

Sistema di riferimento * *i* [Inietta Metadati Madre](#)
http://www.opengis.net/def/crs/EPSS/0/4936

Latitudine Longitudine * *i* [Inietta Metadati Madre](#)
-30.7 134.2

Dimensione coordinate *i*
2=xy

Sostegno

Età sostegno
>= 10 anni

Materiale sostegno *
cemento

Altri elementi sul sostegno * *i*
hotspot Wi-Fi

Installazione altri elementi sul palo
Non integrati

Stato del sostegno *
da sostituire

Metadati Installazione *i* [Inietta Metadati Madre](#)

Inizio validità del dato
03/05/2019

Fonte del dato *
rilievo diretto

Fine validità del dato
02/06/2019

Scala * *i*
scala 1:10000

Figura 56 – punto luce

APPARECCHIO

Le modifiche ai campi contrassegnati dal simbolo * saranno replicate a tutti gli elementi dello stesso livello

Tipologia di apparecchio * ≡ i

lanterna ▼

Id apparecchio * i

AP01

Età alla data del rilievo i

>= 5<10 anni ▼

Marca

C.&G. CARANDINI SA (E) ▼

Modello i

Proprietà del punto luce * ≡

Terzi ▼

Tipo di chiusura dell'apparecchio

Nessuna selezione ▼

N. lampade o moduli per singolo apparecchio * ≡

1

Potenza ai morsetti * ≡ i

W 50

Stato apparecchio alla data del rilievo *

da sostituire ▼

Indice ipca (cam 2013) * i

Non applicabile: valore previsto dalla specifica ma non applicabile all'istanza (ad es. non e' applicabile la categ) ▼

Indice ipca* (cam 2018) *

Non applicabile: valore previsto dalla specifica ma non applicabile all'istanza (ad es. non e' applicabile la categ) ▼

Emissione diretta verso l'alto dell'apparecchio ≡

45

Unità di misura ≡ i

gradi

Apparecchio cut off ≡ i

☐ No ☒ Si

Flusso caratteristico apparecchio ≡ i

lm 5000

Metadati Apparecchio 📄 Inietta Metadati Madre

Inizio validità del dato

03/05/2019

Fine validità del dato

02/06/2019

Fonte del dato *

rilievo diretto ▼

Scala * i

scala 1:10000 ▼

Figura 57 - apparecchio

SORGENTE LUMINOSA

Le modifiche ai campi contrassegnati dal simbolo * saranno replicate a tutti gli elementi dello stesso livello

Id sorgente luminosa *

SORG01

Tipologia sorgente luminosa * ≡ i

LED ▼

Potenza caratteristica della sorgente luminosa (lampada o modulo provvisto di attacco - intercambiabile) * ≡ i

W 50

Flusso luminoso caratteristico della sorgente luminosa (lampada o modulo provvisto di attacco ?intercambiabile) * ≡

lm 5000

Marca sorgente luminosa

Pluto

Modello sorgente luminosa

Topolino

Temperatura correlata di colore cct i

k

Metadati Sorgente Luminosa 📄 Inietta Metadati Madre

Inizio validità del dato

03/05/2019

Fine validità del dato

02/06/2019

Fonte del dato *

rilievo diretto ▼

Scala * i

scala 1:10000 ▼

Figura 58 – sorgente luminosa

La scheda zona omogenea presenta (Figura 59):

- Anagrafica: 27 righe e 1-3 colonne;

La scheda zona omogenea estesa presenta:

- Anagrafica: 6 righe e 1-3 colonne;

ZONA OMOGENEA

Zona Omogenea

Identificativo univoco zona omogenea * *i* Data del rilievo zona omogenea * *i*

Area illuminata

Tipologia di area illuminata * *i* Altra tipologia di area illuminata *i*

Classificazione della strada e categoria illuminotecnica di progetto * *i*

Manto stradale

Tipologia * *i* Altra tipologia *i*

Reticolo

Lunghezza del reticolo di riferimento Larghezza del reticolo di riferimento

Superficie area illuminata

Carreggiata

Tipo * *i* N. di corsie prima carreggiata *i* N. di corsie seconda carreggiata *i*

Marciapiede

Presenza di marciapiede *i* Larghezza marciapiede *i* Larghezza altro marciapiede *i*

Illuminamento di progetto *i* Luminanza di progetto *i*

Figura 59 – zona omogenea

La struttura così presentata, descrive completamente il form in tutte le sue componenti, ed è descritta nel file formSchema.json in assets/data/scheda censimento/, di cui si riporta, come esempio, solo le prime righe, relative alla scheda anagrafica (Figura 60):

```

[{"JSON": [{"idForm": "anagraficaForm", "labelForm": "Anagrafica illuminazione pubblica", "mainTable": "sc_PLSystemGeneralData", "tabs": [{"idPanel": "an_model", "idTab": "anagrafica", "labelTab": "", "rows": [{"type": "default", "cols": [{"fields": [{"id": "sc_anagraficaForm_CensusTechSheet_label"}]}]}, {"type": "default", "cols": [{"fields": [{"id": "sc_anagraficaForm_PLSystemGeneralData_City"}, {"id": "sc_anagraficaForm_PLSystemGeneralData_ISTATCode"}, {"id": "sc_anagraficaForm_PLSystemGeneralData_Region"}]}]}]}]}]}]}

```

Figura 60 – esempio di formSchema.json

Attraverso un ulteriore file, fieldList.csv in assets/data/scheda_censimento/, viene infine valorizzato il form; infatti questo file contiene tutta la descrizione dei singoli campi e del loro posto:

- nell'xsd che descrive la scheda censimento (su cui si basa tutta la validazione dei dati e l'importazione delle schede censimento tramite xml);
- nel database;

order	type	addon	form	xsd	special	field	output	section	table	title	labelSmall	label	code	required	nillable	classes	base	min	mintype	max	maxtype	step	minlength	maxlength	value	um	definition	options	validation
1	text		anagraficaArea	Polygon	gml_id		1	area	sc_Polygo	Identifica	0	META_SUI	202101	0	1								1					Identificatore univoco	name
2	text		anagraficaArea	Polygon	gml_posi		1	area	sc_Polygo	Lista coord	0	META_SUI	202101	1	0								1					Lista di coordinate geografiche	
3	select		anagraficaArea	Polygon	gml_srsDi		1	area	sc_Polygo	Dimensioni	0	META_SUI	202101	0	1				2				1					Dimensione delle coordinate	
4	text		anagraficaArea	Polygon	gml_srsNi		1	area	sc_Polygo	Sistema d	0	META_SUI	202101	1	0								1					Sistema di riferimento in cui sc	

Figura 61 – fieldList.csv

In Figura 61 è riportato l'header del csv e delle righe di esempio; di seguito alcuni campi tra i più importanti, che permettono di capire la logica di costruzione dinamica del form.

- type: indica il tipo di dato sul database, come per esempio txt, select, ecc...;
- form: indica il form che contiene il campo. Es. anagraficaForm;
- il tag dell'xsd in cui è posizionato il dato;
- il nome del campo;
- la sezione;
- la tabella;
- il nome del campo;
- required;
- nillable.

La funzione genera la struttura del form attraverso il file json, e lo popola, generando direttamente l'html con i campi indicati nel csv, andando a recuperare i dati direttamente sulla tabella del database indicata nel campo table.

Questo vuol dire che per aggiungere un campo alla scheda censimento, basta aggiungere il record nel csv e nel json e l'html verrà aggiornato di conseguenza.

Una volta creato dinamicamente il form, viene invocata la funziona javascript SchedaManager.init in assets/js/enea/scheda_censimento/scheda-manager.js che effettua una chiamata ajax verso il controller Scheda.php in application/controller/api/enea/schedaCensimento/ che attraverso la funzione tree_get() recupera la struttura ad albero da visualizzare (Figura 62).

```
public function tree_get()
{
    $sc_CensusTechSheet_id = $this->get('id');
    if (!is_numeric($sc_CensusTechSheet_id)) {
        $this->response(NULL, REST_Controller::HTTP_BAD_REQUEST); // BAD_REQUEST (400)
    }

    if (!$this->CensusTechSheet_model->can($sc_CensusTechSheet_id)) {
        $this->response(NULL, REST_Controller::HTTP_UNAUTHORIZED); // UNAUTHORIZED (401)
    }

    $result = $this->CensusTechSheet_model->get_tree($sc_CensusTechSheet_id);
    $data = array('data' => $result);
    $this->response($data, REST_Controller::HTTP_OK); // OK (200)
}
```

Figura 62 – controller Scheda.php, funzione tree_get()

L'html viene infine passato dal controller (Figura 63) alla views application/views/enea/scheda_censimento/form/layout.php.

```
<div class="panel panel-white">
  <div class="panel-heading border-light">
    <div class="row">
      <div class="col-md-12">
        <h4 class="panel-title">Scheda Censimento</h4>
      </div>
    </div>
  </div>
</div>

<div class="row" id="box-edit-sc">
  <?php
    if ($this->config->item('static_form')) {
      $this->load->view('enea/scheda_censimento/form/layout_static');
    } else {
      $this->load->view('enea/scheda_censimento/form/layout_parametric');
    }
  <?>
</div>
```

Figura 63 – views layout.php

Poiché il form è parametrico, viene caricata la view application/views/enea/scheda_censimento/form/layout_parametric.php di cui si riportano le parti più importanti:

- Sezione per l'amministratore (Figura 64, Figura 65);

```
<div id="schede_utilities">
  <div class="row">
    <div class="col-md-6">
      <label class="" for="show_id_schede">Mostra ID
      Scheda</label>

      <div class="form-group">
        <label class="radio-inline">
          <input value="0" type="radio"
            checked="checked">No
        </label>
        <label class="radio-inline">
          <input value="1" type="radio"
            checked="checked">Si
        </label>
      </div>
    </div>
    <div class="col-md-6">
      <label class="" for="show_id_schede">Edita ID
      Nascosti</label>

      <div class="form-group">
        <label class="radio-inline">
          <input value="0" type="radio"
            checked="checked">No
        </label>
        <label class="radio-inline">
          <input value="1" type="radio"
            checked="checked">Si
        </label>
      </div>
    </div>
  </div>
</div>
```

Figura 64 – layout_parametric.php sezione amministratore

Figura 65 – visualizzazione sezione amministratore

- Form di ricerca nella scheda o struttura (Figura 66, Figura 67, Figura 68);

```
<div class="row">
    <div class="col-md-6">
        <div class="form-group">
            <input type="text" class="form-control input-sm" value=""
id="schede_search" placeholder="Cerca in scheda">
        </div>
    </div>
    <div class="col-md-6">
        <div class="form-group">
            <div class="btn-group">
                <button id="prev_search" class="btn btn-primary btn-
sm">
                    <i class="fa fa-step-backward"></i>
                </button>
                <button id="next_search" class="btn btn-primary btn-
sm">
                    <i class="fa fa-step-forward"></i>
                </button>
                <button id="schede_search_apply" class="btn btn-sm btn-
success" type="button">
                    <i class="fa fa-search"></i>
                </button>
                <button id="schede_search_reset" class="btn btn-sm btn-
red" type="button">
                    <i class="fa fa-trash"></i>
                </button>
            </div>
        </div>
    </div>
</div>
```

Figura 66 – layout_parametric.php form di ricerca

```
<div class="row">
    <div class="col-md-9">
        <div class="form-group">
            <input type="text" class="form-control input-sm" value=""
id="tree_search" placeholder="Cerca in struttura">
        </div>
    </div>
    <div class="col-md-3">
        <div class="form-group">
            <button id="tree_search_reset" class="btn btn-sm btn-info"
type="button">Reset <i
class="fa fa-refresh"></i></button>
        </div>
    </div>
</div>
```

Figura 67 - layout_parametric.php form di ricerca

Figura 68 – visualizzazione layout_parametric.php form di ricerca

- Visualizzazione albero (Figura 69, Figura 70)

```
<div id="tree_view"></div>
```

Figura 69 – visualizzaione albero



Figura 70 – visualizzazione albero

- Il form della scheda censimento, che carica una ulteriore view cioè CensusTechSheetParametric.php che carica il form parametrico con tutti i dati (Figura 71):

```
<div class="col-md-8">
    <?php $this->load->view('enea/scheda_censimento/form/censusTechSheetParametric'); ?>
</div>
```

Figura 71 – vista che carica il form parametrico

Vengono caricati tutti i form relativi alla scheda censimento ma, a seconda della tab che viene visualizzata, il css nasconde le altre tab.

-esporta: funzione di esportazione, invoca la funzione export nel controller Scheda_censimento.php e restituisce la scheda in formato .xml.

```
public function export($id) {
    $this->load->library('XML_IO');
    // $this->xml_io->generateXML($id, true);
    $this->xml_io->export($id, true);
    die;
}
```

Figura 72 – funzione di esportazione scheda

-dettaglio kpi: è la funzione di visualizzazione kpi.

Un'altra funzionalità di particolare importanza è il calcolo dei KPI (Key Performance Indicator) statici in fase di sottomissione scheda.

Per visualizzare i kpi, il controller invoca il KPI.js in assets/js/enea/scheda_censimento/ passando l'id della scheda (Figura 73), in particolare la funzione dataTableKPI che effettua una chiamata ajax alla funzione kpi_table() in /api/enea/scheda-censimento passando il tipo di kpi e l'id della scheda. Viene quindi caricato il model in enea/scheda_censimento_model.php che effettua la query sul database e recupera i kpi e li passa alla view kpi.php.

```
public function kpi($id,$label)
{
    $this->data['breadcrumbs'] = array();
    $this->data['page_title'] = $this->lang->line('scheda_censimento');
    $this->data['page_subtitle'] = '';
    $this->add_asset('css_page', 'assets/css/administration-menu.css');
    $this->add_asset('css_page', 'assets/plugins/select2/css/select2.min.css');
    $this->add_asset('css_page', 'assets/plugins/DataTables/media/css/DT_bootstrap.css');
    $this->add_asset('css_page', 'assets/css/enea/scheda_censimento/kpi.css');

    $this->add_asset('js_plugins', 'assets/plugins/bootbox/bootbox.js');
    $this->add_asset('js_plugins', 'assets/plugins/select2/js/select2.min.js');
    $this->add_asset('js_plugins', 'assets/plugins/DataTables/media/js/jquery.dataTables.js');

    $this->add_asset('js_page', 'assets/js/enea/scheda_censimento/kpi.js');
    $this->add_init("KPI.init('.intval($id).','.$label.')");
    $this->render('enea/scheda_censimento/kpi');
```

Figura 73 - kpi

In Figura 74 la scheda relativa ai kpi come visualizzata sul sistema.

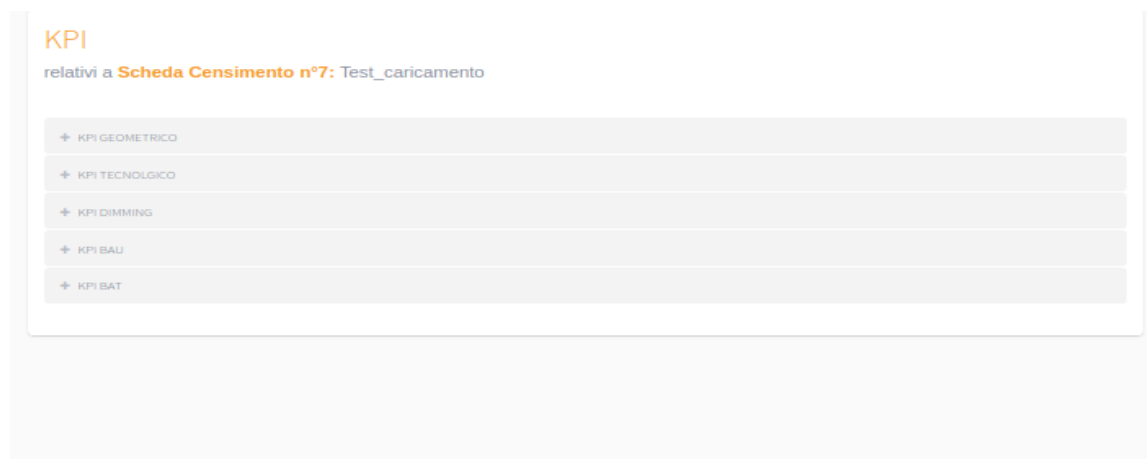


Figura 74 - kpi

- **-conferma:** è il tasto di sottomissione della scheda che la congela e calcola i kpi. Premendo il tasto viene invocata la funzione `submit_get()` di `api/enea/schedaCensimento/` che converte la scheda censimento in xml e poi effettua il check formale attraverso la funzione `verifyXML` della libreria `XML_IO` (Figura 75)

```
public
function submit_get()
{
    $id = $this->get('id');
    $this->load->library('XML_IO');
    // Generate file
    // $output = $this->xml_io->generateXML($id, false);
    $output = $this->xml_io->export($id, false);
    // Verify file
    $verify = $this->xml_io->verifyXML($output);
    // Delete file
    unlink($output);
    // Response
    $result = ($verify === true) ? array('success' => true) : array('error' => $verify);
    $this->set_response($result, REST_Controller::HTTP_OK);
}
```

Figura 75 – funzione verifyXML

Se l'esito della validazione è positivo, vengono calcolati i kpi attraverso la funzione js in `schede-manager`, `SchedeManager.calculateKPI`, passando l'id scheda;

- **-rimuovi:** il tasto invoca la chiamata ajax verso la funzione `delete` in `api/enea/schedaCensimento/scheda/`

La modalità di compilazione di una scheda può avvenire nel seguente modo:

- Manuale: attraverso la compilazione del form online descritto in precedenza
- Automatico: attraverso il caricamento di un file .xml oppure .zip opportunamente formattati. Attraverso questa modalità è altresì possibile importare massivamente più impianti contemporaneamente, riducendo le probabilità di errore rispetto ad una compilazione totalmente manuale e riducendo i tempi di inserimento dei dati.

L'importazione di una scheda censimento avviene dalla stessa pagina in cui è visualizzata la lista delle schede censimento (Figura 76).

ELENCO SCHEDE CENSIMENTO

Id	Descrizione	codice istat comune	nome comune	N. POD	N. quadri	N. punti luce	Ultima modifica	Sottomissione	Azioni
7	Test_caricamento	12058091	Roma	1	1	1	26/01/2020 09:58:12	02/08/2019 11:26:20	C X U D
6	Prova Linee Guida 01	12058091	Cesano di Roma	2	3	2	26/06/2019 16:24:09		C X U D
2	scheda_12058003_201906121239	120580912	Cesano di Roma	2	3	10	26/06/2019 11:50:39		C X U D

Show 10 entries

Previous 1 Next

CARICA XML

Attenzione

- Sono supportati i formati .xml e .zip, si consiglia l'utilizzo del secondo per file superiori ai 10MB.
- Il processo di upload può richiedere dei minuti, non chiudere la pagina finché non è completo il caricamento.

Descrizione

Figura 76 – sezione caricamento xml

Attraverso il tasto “seleziona file”, viene caricato il file xml che poi viene salvato sul database previa validazione.

```

$('#upload_form').on('submit', function (e) {
    e.preventDefault();
    $.ajax({
        type: 'POST',
        url: upload_url,
        data: new FormData(this),
        contentType: false,
        cache: false,
        processData: false,
        dataType: 'json',
        beforeSend: function () {
            $('#upload_form button').attr('disabled', 'disabled');
            $('body').block({
                overlayCSS: {
                    backgroundColor: '#fff'
                },
                message: '<i class="fa fa-spinner fa-spin fa-5x"></i><br>',
                css: {
                    border: 'none',
                    color: '#333',
                    background: 'none',
                }
            });
        },
    });
},

```

```

complete: function (response) {

    $('body').unlock();
    var json_response = response.responseJSON;
    console.log(json_response);
    if (json_response.error) {

        var verify_response = '<div class="alert alert-block alert-danger"><h4 class="alert-heading"><i class="fa fa-times"></i> Errore di validazione</h4><p>sono presenti '
+ json_response.error.length + ' errori</p></div>';

        if (json_response.validation == 'schematron') {

            $(json_response.error).each(function (key,
error) {

                var classError = 'alert-danger';
                if (error.flag == 'warning') {
                    classError = 'alert-warning';
                }
                verify_response += '<div class="alert '
+ classError + ">";
                Verifyazione: </strong>Schematron<br/>';
                </strong>'+error.flag+'<br/>';
                </strong>'+error.test+'<br/>';
                </strong>'+error.location+'<br/>';
                </strong>'+error.description;

                verify_response += '<div class="alert '
+ classError + ">";
                verify_response += '<strong>Metodo '
+ error.test + '</strong>';
                verify_response += '<strong>Livello: '
+ error.location + '</strong>';
                verify_response += '<strong>Test: '
+ error.test + '</strong>';
                verify_response += '<strong>Settore: '
+ error.location + '</strong>';
                verify_response += '<strong>Descrizione: '
+ error.description + '</strong>';
                verify_response += '</div>';
            });
        }
        else {
            $(json_response.error).each(function (key,
error) {

                alert-danger">';
                Verifyazione: </strong>XSD<br/>';
                error.line + ': </strong>' + error.message + '</div>';

                verify_response += '<div class="alert '
+ classError + ">";
                verify_response += '<strong>Metodo '
+ error.test + '</strong>';
                verify_response += '<strong>Linea ' +
error.line + '</strong>';
                verify_response += '</div>';
            });
        }
    }
    $('#upload_error').removeClass('hidden').find('p').html(verify_response);
    $('#upload_form_result #upload_success').hide();
}
else if (json_response.validation == 'schematron') {
    $('#upload_form_result '
+ verify_response).html(verify_response);
    $('#upload_form_result #upload_success').hide();
}
else {
    $('#upload_form_result '
+ verify_response).html(verify_response);
    $('#upload_form_result #upload_success').hide();
}
}
dt.ajax.reload();
$('#upload_form button').removeAttr('disabled');
},
error: function (jqXHR, text_status, error_thrown) {
    console.log(jqXHR, text_status, error_thrown);
    $('body').unlock();
    $('#upload_form_result #upload_error').removeClass('hidden');
    if (typeof jqXHR.responseJSON === 'undefined') {

        var errorTxt = 'Loading XML failed';
    }
}

```

```

    else {
        var errorTxt = jqXHR.responseJSON.error;
    }
    $('#upload_form_result #upload_error').empty().html('<p>Error: '
+ errorTxt+'</p>');
    $('#upload_form_result #upload_success').hide();
});
});

```

Figura 77 – schede-manager.js upload

La compilazione automatica prevede una validazione molto stringente dei dati già in fase di importazione, per evitare l'importazione di un impianto che presenti dati erranei o non idonei alle valutazioni economiche/finanziarie previste dalla piattaforma.

Quando viene caricato il file, il javascript `schede-manager.js` invoca il controller `application/controllers/api/enea/Scheda_censimento.php`, in particolare la funzione `upload_post()` che si occupa del caricamento.

Questa funzione è molto complessa, pertanto si mostreranno solo i passaggi fondamentali al caricamento della scheda censimento.

Per prima cosa vengono caricate le due librerie necessarie:

- XML_IO
- SchematronLib

La prima permette il caricamento dell'xml, effettuando la validazione formale del file secondo lo schema xsd; viene infatti a tale scopo invocato la funzione `xsd_validation` (in `controllers/api/enea/scheda_censimento.php`) che a sua volta invoca la funzione `verifyXml` delle librerie `xml_io(application/libraries/xml_io.php/verifyXML, in Figura 78)` che a sua volta invoca lo `schemaValidate` dell'xsd (una libreria che effettua la validazione sulla base delle specifiche indicate sull'xsd). In caso di errori viene restituito un messaggio di errore validazione (Figura 79)

```

public function verifyXML($file)
{
    libxml_use_internal_errors(true);
    $xml = new DOMDocument();
    $xml->load($file);
    if (!$xml->schemaValidate($this->xsd)) {
        return libxml_get_errors();
    } else
        return true;
}

```

Figura 78 – verifyXML

```

foreach ($files as $file) {
    error_log($file);
    // validazioni
    if ($this->verify_xml == true) {

        // validazione xsd

        error_log('inizio XSD: '.time());
        $validationXsd = $this->xsd_validation($file);

        if ($validationXsd !== true) {
            error_log('D - validazione XSD fallita');
            $validationXsd = array('error' => $validationXsd);
            $this->response($validationXsd,
REST_Controller::HTTP_INTERNAL_SERVER_ERROR);
        }
    }
}

```

```

else {
    error_log('D - validazione XSD riuscita');
}

error_log('fine XSD: '.time());

// validazione schematron
error_log('inizio SCHEMATRON: '.time());
// $xmlToTest_path =
'/var/www/html/pell/application/third_party/ValidazioneSchematron/SchedeCensimento/SchedaCensimentoEsempioValido.xml';

$validationSchematron = $this->SchematronLib->schematron_validation($file);
if ($validationSchematron['success'] !== true) {
    error_log('E - validazione SCHEMATRON fallita');
    $this->response($validationSchematron, REST_Controller::HTTP_INTERNAL_SERVER_ERROR);
}
else {
    error_log('E - validazione SCHEMATRON riuscita');
}
error_log('fine SCHEMATRON: '.time());
}

```

Figura 79 – schema di validazione

La seconda libreria effettua una validazione dei singoli campi secondo la funzione `schematron_validation` della libreria `SchematronLib` in Figura 80.

Essa carica in una cartella temporanea l'xml importato, crea un file xml di output dove salvare il risultato della validazione, ed infine lancia il validatore `udValidator.jar` che è una applicazione java che controlla i campi e restituisce un file xml con i record errati.

Per ciascuno di essi viene settato l'errore con i seguenti campi

- Descrizione
- Condizione errata
- Campo

E viene visualizzato nella view per dar modo all'operatore di correggere gli errori.

Se il caricamento va a buon fine la tabella presenterà un record relativo all'ultima scheda aggiunta.

```
public function schematron_validation($xmlToTest_path) {

    //error_log($xmlToTest_path);
    //error_log('sono in schematron library '.time());

    $xmlToTest_pathA = explode("/", $xmlToTest_path);
    if ($xmlToTest_pathA[1] == 'uploads') {
        $upload_path = realpath(APPPATH.'../'.$xmlToTest_pathA[1].'/'.$xmlToTest_pathA[2]);
    }
    else {
        $upload_path =
realpath(APPPATH.'../'.$xmlToTest_pathA[1].'/'.$xmlToTest_pathA[2].'/'.$xmlToTest_pathA[3].'/'.$xmlToTest_
pathA[4]);
    }

    $result = [
        'validation' => 'schematron',
        'success' => false,
        'error' => []
    ];

    $this->CI->config->load('enea_pell');
    $schematron = $this->CI->config->item('schematron');
    //error_log(json_encode($schematron));
    $this->CI->load->library('XML_IO');

    $xmlId = time();
    $outputFolder = "output_". $xmlId;

    // lancio il validator
    //error_log('lancio validator');
    $command = 'java -jar udValidator.jar -s '.$upload_path.' -sc
'.APPPATH.$schematron['schematron_path'].' -o '.APPPATH.$schematron['result_path'].'/'.$outputFolder;
    //error_log($command);
    exec('cd '.APPPATH.$schematron['validator_path'].'; '.$command,$output);

    // elaboro il risultato
    $resultXmlFile = APPPATH.$schematron['result_path'].'/'.$outputFolder.'/result.xml';

    //error_log($resultXmlFile);

    //$resultXml = read_file($resultXmlFile);
}
```

```

if (file_exists($resultXmlFile)) {
    //error_log('il file esiste');

    $xml = simplexml_load_file($resultXmlFile);
    //error_log(json_encode($xml));
    $xml->registerXPathNamespace('svrl', 'http://example.org/svrl');
    $failedAssert = $xml->xpath('//svrl:failed-assert');
    $text = $xml->xpath('//svrl:text');
    $errors = [];
    $isFailed = false;
    foreach ($failedAssert as $key => $assert) {

        $assertJ = json_encode($assert);
        $assertA = json_decode($assertJ, true);
        $flag = $assertA["@attributes"]["flag"];
        if ($flag == 'fatal') {
            $isFailed = true;
        }
        $error = [
            'flag' => $flag,
            'description' => (string)$text[$key][0],
            'location' => $assertA["@attributes"]["location"],
            'test' => $assertA["@attributes"]["test"]
        ];
        array_push($errors, $error);
    }

    if ($isFailed === true) {
        $result['success'] = false;
    }
    else {
        $result['success'] = true;
    }

    //error_log('cancello file');

    $this->delete_files(APPPATH.$schematron['result_path'].'/'.$outputFolder);
    //error_log('cancellato file');
    $result['error'] = $errors;
} else {
    error_log('XML da testare non trovato al path: '.$upload_path);
}

//error_log(json_encode($result));
return $result;
}

```

Figura 80 – Libreria Schematron per la validazione

2.6 Struttura dell'applicazione: valutazioni economico-finanziarie

Sulla base della definizione delle schede censimento, il sistema PELL provvede a presentare una sezione interamente dedicata alle valutazioni economico-finanziarie degli impianti.

Questa sezione, seppur analizzata in modo approfondito, è ancora in fase di completamento pertanto, vengono fornite solo alcune informazioni di base circa la sua struttura.

La schermata relativa alla sezione è mostrata in Figura 81.

Errore. L'origine riferimento non è stata trovata.

Data	Valutazione	Impianto	Zone omogenee
09/10/2019 12:24	Prima Valutazione Impianto 1	ImpiantoTest01	• via padova
10/10/2019 11:48	yuuu	Impianto 2	• via roma
10/10/2019 11:59	igg	Impianto 2	• via padova • via roma
16/10/2019 11:49	prova	prova_fabio_osea_15_10	• via padova 2

Figura 81 - sezione "valutazioni economico-finanziarie"

La sezione è divisa in 4 parti:

- Tabella con lo storico delle valutazioni;
- Definizione degli investimenti da effettuare: l'investimento può essere configurato nei suoi parametri, cliccando sul tasto "Gestione Investimenti".
- Definizione e visualizzazione degli impianti da valutare: vi si può accedere cliccando sul tasto "Gestione impianti".
- Creazione di una nuova simulazione: può essere effettuata una simulazione attraverso il tasto "Nuova simulazione".

Il controller relativo a tutta questa sezione è Nisimm.php che estende My_Controller.php e si trova nella cartella application/controllers/enea/.

La funzione index() carica il model nRecapModel.php, per il recupero dello storico valutazioni da visualizzare in tabella, e la libreria custom RecapLib.

Viene sempre recuperato l'elenco delle città visibili all'operatore, attraverso il recupero delle informazioni dalla sessione; successivamente la libreria recupera tutte le valutazioni effettuate per le città recuperate, crea la tabella con le valutazioni e passa i dati alla vista enea/isimm/valutazioni/index.php, che visualizza le informazioni sotto forma di tabella.

Le sezioni "Gestione investimenti", "Gestione Impianti" e "Nuova simulazione" sono accessibili tramite link alle pagine, definiti nella view.

2.6.1 Sezione investimenti

La sezione “Gestione investimenti” permette di configurare i dati di investimento ai fini delle valutazioni economico-finanziarie dell’impianto.

In Figura 85 **Errore. L'origine riferimento non è stata trovata.** si può osservare la sezione relativa alla gestione dell’investimento; sono presenti e configurabili vari parametri, alcuni sono personalizzabili in base al tipo di investimento da effettuare, altri sono preconfigurati con dei valori stabiliti dalla situazione economica attuale.

Gestione investimento

Investimento	es. Investimento 1
	1 Testo identificativo dell'investimento
Comune	Seleziona un comune
	1 Comune di riferimento
WACC	0,00 %
	1 Costo medio ponderato del capitale (Weighted Average Cost of Capital), espresso in percentuale
Costo unitario energia	0,19 €
	1 Costo per l'acquisto dell'energia
Durata incentivi	5
	1 Numero di anni nel quali si può usufruire dei benefici economici derivanti da incentivi statali per il risparmio di CO2 emessa, ovvero i cosiddetti "Certificati Bianchi"
KWh per TEP	5.347,49
	1 Fattore di conversione kilowattora-TEP, espresso come numero di kilowattora per singolo TEP
Valore monetario TEP	100,00 €
	1 Valore monetario del singolo TEP, o tonnellata equivalente di petrolio in euro
Costo gestione	0,00 €
	1 Costo annuale di gestione della sezione di impianto in termini di personale e asset tangibili
Durata ammortamento	30
	1 Durata dell'ammortamento misurata in numero di anni
Durata progetto	30
	1 Durata del progetto di finanziamento espressa in anni
Imposte	35,00 %
	1 Tasso percentuale relativo alle imposte a carico del soggetto privato (percentuale)
Quota finanziata	75,00 %
	1 Quota parte dell'investimento attribuita al soggetto privato (percentuale)
Costo finanziamento	0,00 €
	1 Costo annuale per la stipulazione di assicurazioni, fidejussioni, etc. ai fini della stipula di finanziamento
	Salva Annulla

Figura 82 - gestione investimenti

È possibile accedere alla sezione dal link <http://pell.enea.it/enea/isimm/investimenti> ; a tal fine il controller Nisimm.php invoca il javascript nel file /assets/js/enea/isimm/valutazioni/isimm-valutazioni.js, che a sua volta invoca il controller per questa sezione che è application/controllers/api/enea/isimm/Investment.php, che invoca il model nInvestmentModel.php per il recupero degli investimenti già inseriti.

2.6.2 Sezione gestione impianto

Una volta che è stato configurato l’investimento si può passare alla configurazione dell’impianto di cui valutare la qualificazione.

È possibile accedere alla sezione dal link <http://pell.enea.it/enea/isimm/impianti> ; a tal fine il controller Nisimm.php invoca il file javascript \assets\js\enea\isimm\valutazioni\isimm-valutazioni.js, che invoca il controller api/enea/isimm/plantManager, che, a sua volta interroga il model nPlantModel che effettua la query sul database cercando gli impianti per comune.

I dati recuperati sono passati alle views `impianti.php` e `gestioneImpianti.php` in `application/views/enea/isimm/valutazioni/`.

La schermata corrispondente è in Figura 83.



Figura 83 – gestione impianti

Nella sezione “Gestione Impianto”, è possibile definire un nuovo impianto o inserire nuove zone per l’impianto già selezionato (Figura 84).

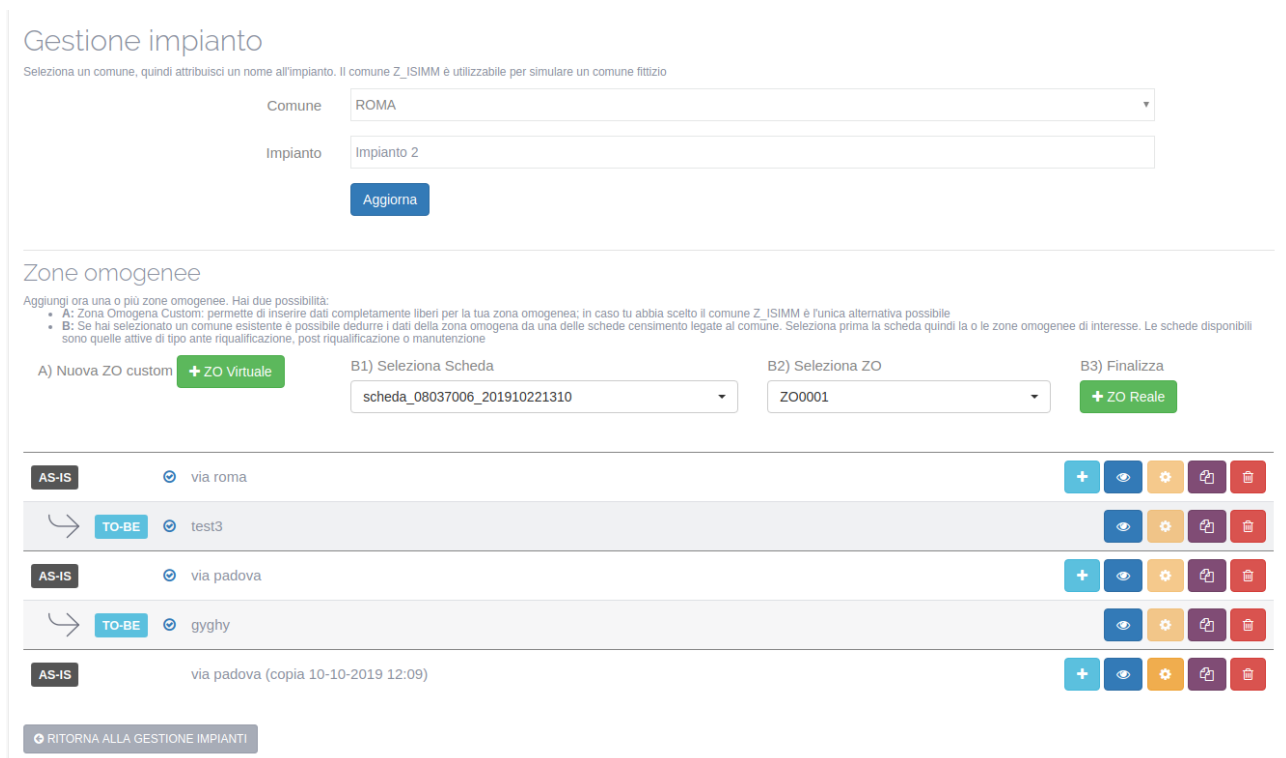


Figura 84 – struttura dell’impianto

2.6.3 Nuova simulazione

Alla sezione si accede tramite il link. È possibile accedere alla sezione dal link <http://pell.enea.it/enea/isimm/content>; a tal fine il controller `Nisimm.php` invoca il metodo `content()` che passa i dati alla view `application/views/enea/isimm/valutazioni/content.php` (Figura 85).

In due passaggi vengono inseriti e valutati i dati relativi alla riqualificazione e, infine, il controller `Nisimm.php` invoca una funzione di riepilogo `getRecap()` che recupera i dati e li passa alla vista `application/views/enea/isimm/recap`.

VALUTAZIONE

1

Step 1

Parametri tecnico-economici

2

Step 2

Parametri economico-finanziari

3

Step 3

Esito ipotesi

Parametri tecnico-economici

Valutazione

es. prima valutazione impianto x

Impianto

impianto 2

ROMA

AS-IS

via roma

TO-BE

test3

AS-IS

via padova

TO-BE

gyghy

AS-IS

via padova (copia 10-10-2019 12:09)

Gestisci impianti

Seleziona tutte le zone omogenee

Step successivo

Figura 85 - simulazione

La valutazione della riqualificazione dell’impianto, da un punto di vista esclusivamente finanziario ed economico, è così completa e a disposizione delle amministrazioni/società per le valutazioni del caso; completa il tutto la possibilità di export in formato PDF della simulazione completa, attraverso apposito tasto.

74

2.7 Struttura dell'applicazione: Prestazioni illuminotecniche

Una sezione interessante, ma ancora in fase di studio e di completamento, è quella relativa alla valutazione delle prestazioni illuminotecniche, ovvero la verifica, per ogni dispositivo di illuminazioni, dell'efficienza di illuminazione in base al contesto stradale dove è posizionato.

Si forniscono di seguito dei cenni circa la struttura della sezione.

L'algoritmo di calcolo degli indicatori illuminotecnici ed energetici relativi alle zone stradali è stato sviluppato da OXYTECH secondo la normativa UNI 13201, e permette anche la valutazione sulla base di modelli appositi da caricare, nonché la rappresentazione grafica dinamica della struttura di illuminazione pubblica.

Al fine del calcolo si rende necessario l'inserimento dei parametri stradali e il riferimento ad un file fotometrico esterno con estensione .ASN, .LDT, .CEN, .CIE, .OXL (formato custom di oxytech).

In **Errore. L'origine riferimento non è stata trovata.**86 è mostrata la schermata che presenta:

- un form di inserimento dei dati;
- un'area con una immagine 3D della struttura di cui si vogliono valutare le prestazioni; essa è anche interattiva, infatti è possibile ruotare l'immagine per vedere il risultato dalle varie angolazioni.

In seguito all'elaborazione, per ogni carreggiata saranno disponibili alcuni dati tra cui:

- illuminamento medio, EMed [lux];
- illuminamento minimo, EMin [lux];
- illuminamento massimo, EMax [lux];
- luminanza media minima mantenuta, LAv [cd/m²];
- uniformità generale, Uo [adim];
- uniformità longitudinale, Ul [adim];
- indice di abbagliamento, Ti [%].

Lato codice il controller di riferimento è Oxytech.php in application/controllers/enea/ che invoca la funzione index() che a sua volta recupera i dati attraverso la chiamata al javascript assets/js/enea/oxitech.js e li passa alla view enea/oxytech/content per la visualizzazione.

Toponimo *

Dispositivo di Illuminazione *

Tipo Strada *

Strada a senso unico

Altra carreggiata *

senza marciapiede

Amplezza carreggiata (in m.)

Amplezza mediana (in m.)

Potenza lampada (in watt)

Carica un modello lampada *

Select file...

File lampada

Rotazione Braccio asse z (RSZ) *

0

Rotazione Lampada asse Z (RZ) *

0

Rotazione Lampada asse X (RX) *

0

Altezza lampada da terra 12 m

Sbraccio lampada da palo 2 m

Calcola

Prestazioni illuminotecniche

Per effettuare il calcolo illuminotecnico sulla base dell'algoritmo Oxytech compilare la form con i dati di interesse. I campi contrassegnati da asterisco sono obbligatori. I formati ammessi per il file da caricare sono .asn, .cen, .cie, .gt, .jes, .ldt, .lit, .oxi, .tmi. In alternativa al caricamento di un nuovo file è possibile utilizzare un modello precedentemente caricato (fino ad un massimo di 10) scegliendolo dalla lista.

RISULTATO CALCOLO OXYTECH

IN QUESTA SEZIONE SONO RIPORTATI I RISULTATI DEL CALCOLO OXYTECH SECONDO I PARAMETRI DI INPUT.

Dati in ingresso

Risultato Analisi

Figura 86 - prestazioni illuminotecniche

2.8 Struttura dell'applicazione: Pell Application

La fase dinamica è attualmente in fase di definizione e sviluppo. Essa rappresenta il cuore applicativo di tutto quanto costruito nella fase statica infatti, a completamento dell'anagrafica di tutti i dispositivi di illuminazione pubblica, verranno storicizzati, valutati, analizzati, monitorati e visualizzati, tutti i dati di consumo energetico giornalieri.

Ciò vuol dire che per ogni dispositivo di illuminazione pubblica, attraverso uno smart meter installato sul quadro elettrico dell'impianto di illuminazione pubblica, i dati relativi ai consumi (energia attiva e reattiva, potenza, cos phi, ecc.) vengono inviati (secondo una granularità da definire, quartoraria, oraria, giornaliera ecc.) verso un sistema che si occupi di storicizzarli per renderli disponibili ad una eventuale elaborazione da un punto di vista prestazionale e statistico, ma anche di visualizzazione grafica dei dati.

Come già specificato, l'adesione al PELL da parte delle amministrazioni comunali prevede l'utilizzo e l'installazione su tutti i quadri elettrici di smart meters che si occuperanno di recuperare i dati che verranno poi inviati al sistema tramite apposita modalità.

2.8.1 PELL Application

I dati che vengono inviati dai meter al server, e storicizzati attraverso apposite piattaforme che verranno descritte in seguito, saranno rielaborati e visualizzati nella sezione "Pell Application", visualizzabile in figura 87, che permette anche una analisi dei dati relativi ai quadri elettrici monitorati.

Attraverso una mappa che utilizza le librerie di Google Maps, è possibile localizzare uno o più PODS (massimo 5), con la possibilità di navigare e zoomare avanti e indietro per visualizzare i dettagli delle varie regioni.

È possibile effettuare la ricerca di un singolo POD, presente nel sistema, inserendo il suo codice nella casella "locate pod", ed esso, se presente, verrà localizzato nella mappa.

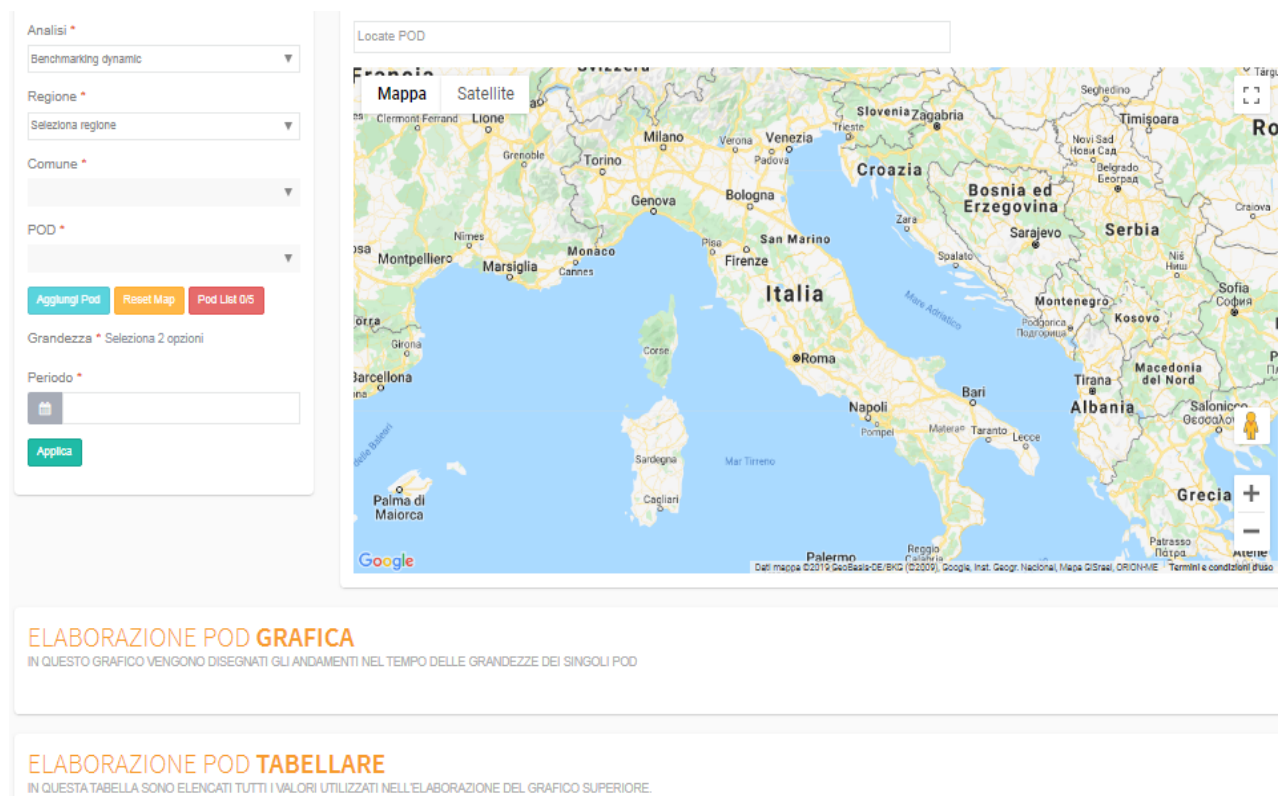


Figura 87 – visualizzazione dati dei pod

Da questa pagina è possibile effettuare analisi di vario genere con i dati di consumo presenti nel database.

Figura 87 - selezione del tipo di analisi

Attraverso la compilazione del form posizionato accanto alla mappa (dettaglio in Fig. 88), è possibile effettuare le seguenti analisi:

- Benchmarking dynamic
- Benchmarking static
- Diagnostica
- Prestazioni.

Il controller di riferimento per questa sezione si trova in `application/controllers/enea` ed è `Pell_application.php`.

Viene utilizzato il file javascript `Pell_application.js` in `assets/js/enea/` per mostrare la mappa attraverso le librerie di Google.

La vista che si occupa di mostrare i dati all'utente è `views/enea/pell_application/content.33`.

2.8.2 Archiviazione dei dati

Una problematica fondamentale del PELL è quella di dover gestire una grande quantità di dati provenienti dagli smart meters installati sui quadri elettrici.

Infatti, i dati relativi ai consumi dei pods devono essere memorizzati e, di conseguenza, elaborati e visualizzati.

Questa operazione deve ovviamente avvenire in modo efficiente e deve essere opportunamente robusta, in grado di gestire quantità di dati:

- molto consistenti (che possono superare l'ordine dei terabyte);

- potenzialmente eterogenei; perché possono contenere informazioni di vario tipo e, in caso di modifiche alla struttura dati, deve essere sempre garantito l'accesso ai dati già disponibili.

Al contempo, deve essere garantita una buona rapidità di accesso ai dati per un monitoraggio che sia efficace e, per eventuali interazioni con sistemi di analisi predittiva per l'analisi dei dati (es. regressione, forecasting, modelli predittivi, ecc.).

Un normale sistema relazionale classico (RDBMS) sicuramente può gestire l'archiviazione di molti dati, tuttavia una scelta di questo genere richiede grandi investimenti economici in termini di storage, ma anche computazionali, perché serve una struttura robusta ed affidabile che non è detto sia poi effettivamente in grado di permettere un accesso real time ai dati memorizzati.

Per questo motivo sono state progettate piattaforme, anche open source, dedicate all'archiviazione dei "Big data", che garantiscono velocità di accesso, affidabilità, sicurezza e scalabilità che non possono essere garantite tramite normali database relazionali.

La maggior parte dei colossi del WEB, come Facebook e Google già utilizzano proprio queste piattaforme per gestire i loro dati, e questo è un indice dell'affidabilità di questi innovativi sistemi.

Una delle piattaforme più utilizzate è stata sviluppata da Apache Software Foundation nel 2008, e prende il nome di Hadoop; proprio Hadoop è stato scelto da ENEA per gestire la fase dinamica ed interagire con il frontend di PELL.

Apache Hadoop è un framework open source che consente l'elaborazione distribuita di grandi set di dati tra cluster di computer utilizzando semplici modelli di programmazione. Vengono utilizzati a tal fine molti macrosistemi (ad esempio HDFS, che è in grado di gestire un numero elevatissimo di file, anche di dimensioni ragguardevoli, dell'ordine dei gigabyte o terabyte), e vengono realizzati dei cluster che possono contenere migliaia di nodi progettati proprio per archiviare molti dati, ottimizzando le operazioni di memorizzazione ed accesso.

I normali filesystem lavorano sull'accesso a molti file di piccole dimensioni, le strutture di questo tipo invece lavorano sull'ottimizzazione dell'accesso a pochi file ma di grandi dimensioni.

Un altro dei componenti fondamentali di Hadoop è MapReduce, un nuovo e innovativo pattern per l'elaborazione distribuita di elevate moli di dati.

In Figura 88**Errore. L'origine riferimento non è stata trovata.** è possibile vedere un semplice schema di molti dei componenti che compongono la struttura di Hadoop.

Le caratteristiche fondamentali di Hadoop possono essere riassunte di seguito:

- Le librerie per l'elaborazione dei dati sono pensate per essere accessibili direttamente sui nodi di calcolo; questo permette di ridurre al minimo i tempi di accesso ai dati perché non vi è necessità di trasferimento degli stessi.
- L'affidabilità della piattaforma è molto elevata, infatti sono le stesse applicazioni che gestiscono i problemi e le anomalie, senza delegare all'hardware tale gestione; pertanto l'eventuale problematica su una macchina viene gestita a priori da applicazioni specializzate.
- La scalabilità; perché, al bisogno, basta aggiungere dei nodi al cluster, che vanno da singoli server a migliaia di macchine, ognuna delle quali offre elaborazione e archiviazione locali.

Se effettuiamo un confronto tra Hadoop ed un classico sistema RDBMS, possiamo renderci conto facilmente del perché esso abbia preso piede per la gestione dei "Big Data", e perché ad oggi sia considerato lo strumento più efficace per gestire i miliardi di dati di consumo di tutti gli impianti di illuminazione pubblica di tutta Italia e, in futuro, anche di tutti gli edifici pubblici (Fig. 89**Errore. L'origine riferimento non è stata trovata.**).

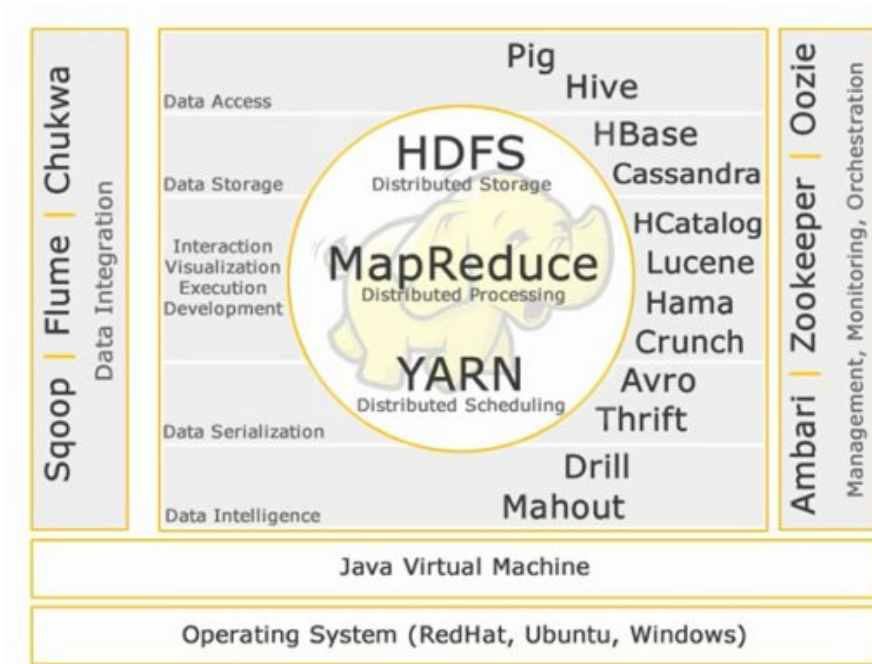
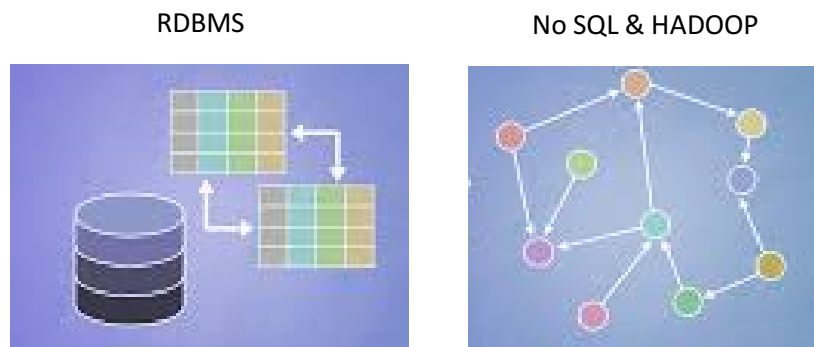


Figura 88 - Hadoop framework



- È necessario uno schema dati già impostato; le tabelle devono essere create prima

Ogni dato deve essere compliant con la struttura interna del database, un dato nel formato errato non

Se devono esser inseriti nuovi dati, è necessario aggiornare manualmente il database aggiungendo nuove colonne, perché difficilmente questa operazione può essere

- Non è necessario avere uno schema dati, gli stessi vengono memorizzati nel file

Il dato viene semplicemente letto ed

L'aggiunta di nuove tipologie di dati non necessita di intervento manuale, i dati possono essere letti o estratti in

Figura 90 - RDBMS vs NOsql

Come si evince da quanto esposto, Hadoop è uno strumento molto complesso, costituito a sua volta da più strumenti che interagiscono tra loro per fare il loro lavoro al meglio e, il suo utilizzo nell'infrastruttura PELL, è garanzia di efficacia, efficienza e sicurezza a tutti i livelli.

2.8.3 Integrazione Hadoop – Frontend

Affidata l'archiviazione dei dati ad un framework appositamente studiato per questo, analizziamo ora come le strutture Hadoop e frontend di PELL possano interagire tra loro.

I dati relativi alle grandezze elettriche degli impianti di illuminazione pubblica, provenienti da smart meters presenti in ogni quadro elettrico, vengono raccolte dai gestori, che si devono allineare a quanto richiesto dalla specifica tecnica, e inviate verso il framework di archiviazione.

Per vari motivi, tra cui la sicurezza del sistema, non è possibile "scrivere" i dati direttamente, ma questi devono essere inviati attraverso una struttura che utilizzi un protocollo abbastanza affidabile e sicuro.

MQTT (Message Queue Telemetry Transport), il protocollo che è stato utilizzato per questa applicazione, è un protocollo pensato per ridurre al minimo l'utilizzo della banda nelle comunicazioni tra dispositivi ed ottimizzare il consumo energetico, infatti, normalmente viene impiegato nei dispositivi che prevedono una alimentazione a batterie e che, per la sua versatilità, può essere utilizzato in molti modi.

Un esempio molto comune di progetto che utilizza il protocollo MQTT è Facebook Messenger.

Quando un client (publisher) vuole comunicare (subscriber) con un altro client, deve pubblicare un messaggio su un certo argomento (denominato topic) sul message broker.

Il message broker ha una duplice funzione:

- Filtra
- Distribuisce

Le comunicazioni tra publisher e subscriber.

Ogni volta che un messaggio viene pubblicato in un topic, il message broker si occupa di distribuirlo a tutti i client iscritti.

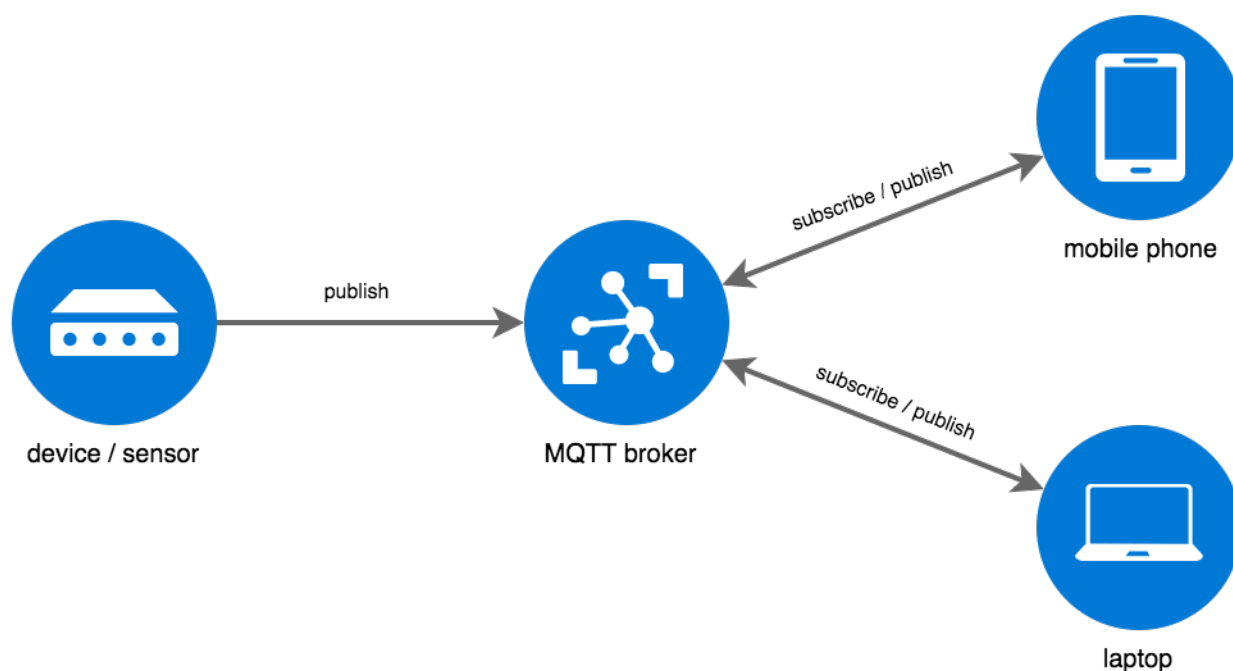


Figura 91 - Esempio di flusso su protocollo MQTT

In Fig. 91**Errore. L'origine riferimento non è stata trovata.** viene mostrato un generico flusso di scambio tra un device (che nel nostro caso è uno smart meter) che pubblica i dati e li rende disponibile al broker. Questo li rende poi disponibili ai vari device, nel nostro caso i server che si occuperanno dell'archiviazione dei dati.

Un broker MQTT molto versatile, semplice da usare e molto comune è MOSQUITTO, un software open source parte di Eclipse Foundation, un progetto iot.eclipse.org e sponsorizzato da cedalo.com.

È un software leggero e disponibile per un uso su tutti i device, dalle semplici board a basso consumo, ai computer, ai server.

Avere il database gestito tramite Hadoop separato anche fisicamente dal PELL (sono su due macchine distinte) oltre che logicamente, è molto importante per motivi di sicurezza (PELL non accede direttamente al database e viceversa) ma anche perché potenzialmente più applicazioni possono integrarsi con il database rendendolo fruibile per svariati progetti. L'utilizzo di una interfaccia, quindi di un broker MQTT, rende il sistema sicuro, e l'utilizzo di macchine con bilanciatori di carico rende il sistema anche efficiente.

In questo modo il broker del PELL comunicherà con il broker MQTT richiedendo i dati da recuperare su Hadoop e il broker MQTT interrogherà il database per fornire i dati di cui PELL necessita. Il tutto in modo indipendente da eventuali richieste di dati da altri server e in modo sufficientemente rapido.

Nel contempo i dati relativi al monitoraggio dei consumi, gestiti dagli smart meters, vengono inviati dai gestori sempre attraverso il broker MQTT che si occupa di passarli al Database dove vengono memorizzati, indipendentemente dal PELL che si occuperà solo di richiedere i dati al Broker.

Questa fase è attualmente in sviluppo e rappresenta un punto molto centrale di tutto il sistema, perché come è importante nella fase statica l'acquisizione e il censimento dei dati degli impianti, è molto importante che si venga a conoscenza di come questi impianti consumino risorse energetiche. Solo attraverso la consapevolezza di tali informazioni è possibile pensare realmente ad una struttura urbana che sia realmente "smart".

3 Conclusioni

Nell'ambito del Progetto PELL, come prima attività, ci siamo occupati della presa in carico della gestione di tutto il frontend del sistema, che include l'aggiornamento/modifica della scheda censimento, del calcolo relativo alla valutazione economica finanziaria e di tutta la parte gestionale/amministrativa legata alla creazione di utenti, gruppi, ruoli e permessi.

La scheda censimento, in particolare, è spesso oggetto di aggiornamento per la sua continua evoluzione per venire incontro alle necessità degli utenti del PELL e, di conseguenza, sono oggetto di aggiornamento continuo, tutte le parti del sistema che si appoggiano alla scheda censimento, in particolare il modulo Save che si occupa di effettuare le valutazioni economico-finanziarie di un impianto da riqualificare.

Ai fini della riduzione dell'incertezza di sistema, il testing del portale ha avuto sia lo scopo di verificare la presenza di eventuali bug, ma principalmente di evidenziare le sezioni che l'utente medio interpretava con difficoltà apportando tutte le migliorie che aumentassero la fruibilità delle pagine e del loro utilizzo.

Questa parte, che corrisponde alla fase statica del progetto, è volta alla raccolta di una grande quantità di parametri che definiscono un impianto di illuminazione pubblica. Questi parametri sono importanti per valutare prestazioni illuminotecniche ed eventuali investimenti.

Inoltre l'informatizzazione della struttura dell'impianto permette di avere disponibili i dati che possono essere elaborati in vari modi, dalla semplice visualizzazione su browser al monitoraggio dell'efficienza dell'impianto, ponendo le basi per uno studio statistico o previsionale del suo funzionamento.

La fase dinamica, che comprende l'acquisizione dei dati di consumo e la relativa visualizzazione sul PELL, è in fase di sviluppo per quanto riguarda la visualizzazione sul sistema, cioè si sta lavorando sull'interfaccia che permette la visualizzazione dei dati monitorati (sezione "pell application") per migliorarne la fruibilità da parte degli operatori, mentre la parte di acquisizione dei consumi è in fase di studio e valutazione.

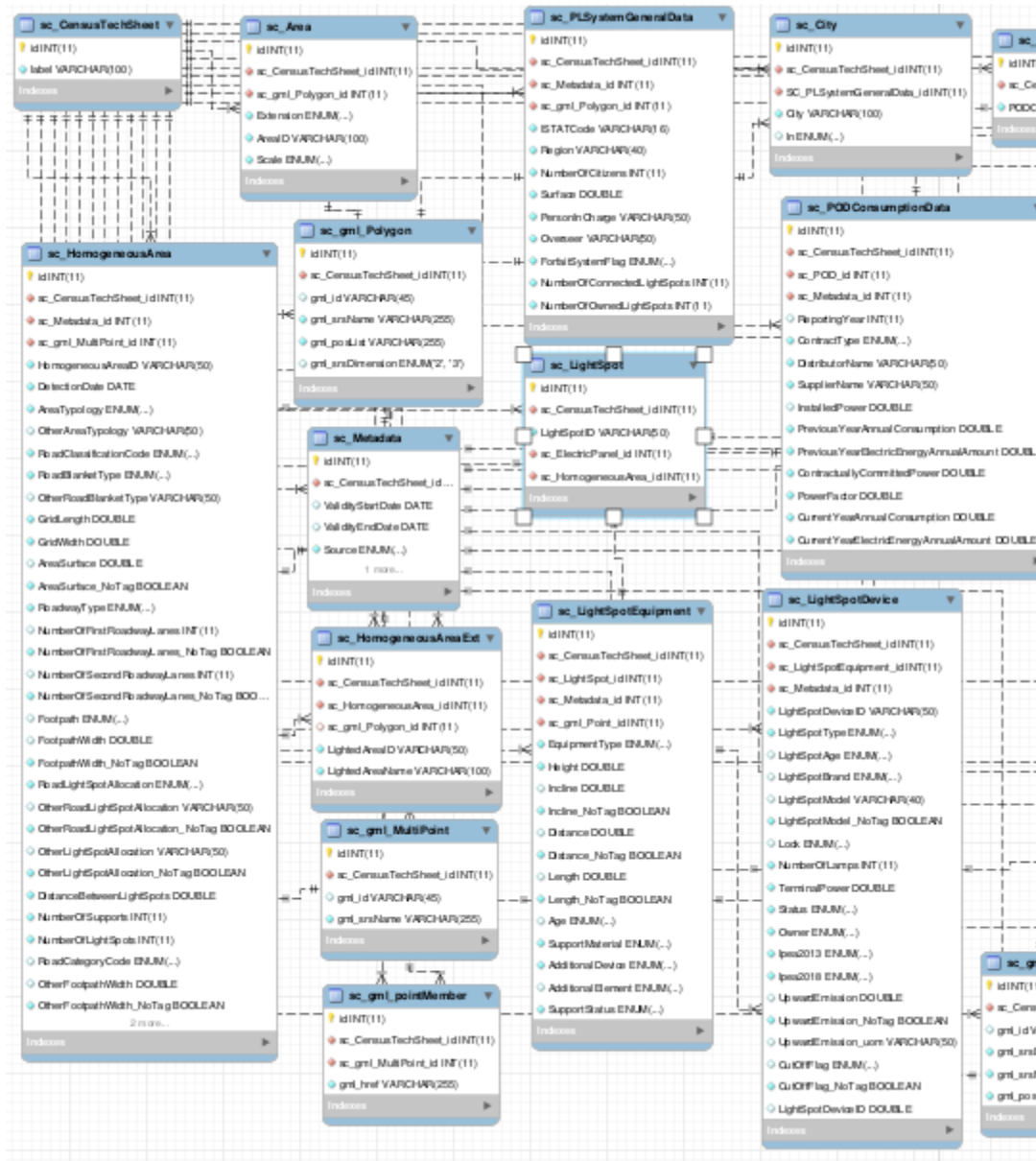
Oltre all'aggiornamento del sistema e ad eventuali miglioramenti, è importante anche la valutazione di eventuali anomalie di sistema che possono verificarsi sull'interfaccia e/o sui calcoli effettuati. Spesso infatti, sebbene vengano fatti dei test specifici per ogni funzionalità, non bastano a scongiurare eventuali anomalie, in quanto è solo attraverso l'utilizzo continuo che possono essere rilevate delle problematiche che impediscono il corretto utilizzo dell'applicazione.

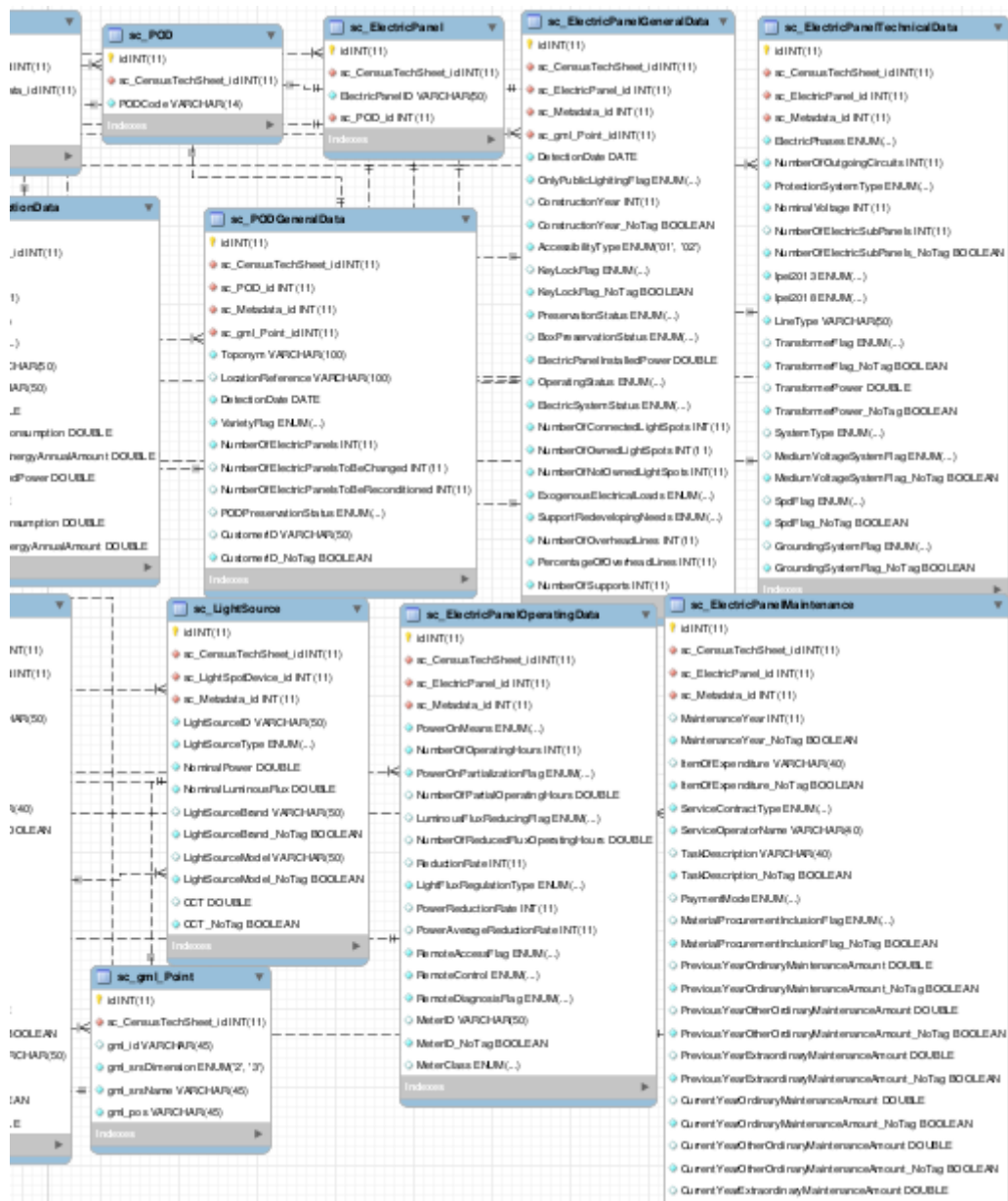
A questo scopo l'applicazione è stata oggetto di profonda critica secondo l'approccio classico che i "quality engineer" del software adottano per individuare i bug nei software sotto indagine. In questo caso, più che l'individuazione di bug, si è realizzata una sorta di "reverse engineering" per conoscere, da un punto di vista tecnico, sia l'applicazione stessa che il suo database per cercare di far emergere situazioni dell'applicazione che potevano essere migliorate sia a livello di use case applicativi, che a livello di codice vero e proprio.

Per tutta la fase dinamica, che è in fase di elaborazione, abbiamo svolto attività di supporto alla progettazione coadiuvando ENEA ed in generale il Gruppo di Lavoro, nella scelta della piattaforma, dei pacchetti applicativi e/o dei linguaggi di programmazione da usare per lo sviluppo di questa fase.

4 Appendice A

Di seguito il diagramma ER relativo alla scheda censimento.





5 Riferimenti bibliografici

I riferimenti bibliografici devono essere richiamati nel testo con numeri progressivi tra parentesi quadre e riportati a fine testo con il seguente formato:

1. ENEA “*Progetto Lumiere*”. Disponibile on line al sito: <http://www.progettolumiere.enea.it/>
2. ENEA “*PELL - Lumière & Public Energy Living Lab (PELL) per una gestione efficiente della Pubblica Illuminazione*” Disponibile al sito: http://www.enea.it/it/comunicare-la-ricerca/events/pell_18mag16/ENEA-Roma.
3. Specifiche di contenuto di riferimento PELL - illuminazione pubblica
https://geodati.gov.it/geoportale/images/Specifica-PELL-IP_ver-1.0_20180723.pdf, 23-07-2018

6 Abbreviazioni ed acronimi

AU: Acquirente Unico

BAT: Best Available Technology

BAU: Business As Usual

CAM: Criteri Ambientali Minimi

CONSIP: Concessionaria Servizi Informativi Pubblici

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

ENEL: Ente nazionale per l'energia elettrica

KPI: Key Performace indicator

kW: kilowatt

LED: Light Emitting Diode

PELL: Progetto Public Energy Living Lab

POD: Point Of Delivery

WEB: web browser.

7 APPENDICE: breve curriculum scientifico del gruppo di lavoro impegnato nell'attività

Laboratorio di Misure Elettriche ed Elettroniche dell'Università degli Studi "Roma Tre"

Responsabile: Dott. Ing. Ph.D. RTI Fabio Leccese

Collaboratori: Dott. Enrico Petritoli (Assegnista di Ricerca), Dott.sa Mariagrazia Leccisi (Borsista)

Il laboratorio fa parte del Gruppo Nazionale delle Misure Elettriche ed Elettroniche (GMEE) i cui scopi principali sono lo studio delle misure o "metrologia", l'analisi di qualità fisiche e la realizzazione di campioni di misura con particolare attenzione allo studio dell'incertezza di misura.

In questo quadro generale, il nostro laboratorio segue da anni diverse linee di ricerca tra le quali la qualità dell'energia (power quality – dal 2004), l'analisi informativa dei segnali (dal 2002), i controlli di apparati locali e remoti ed in particolare di sistemi di risparmio energetico applicati ad illuminazione e riscaldamento (dal 2008), la sensoristica distribuita incluse le Wireless Sensor Network (dal 2008) e le analisi affidabilistiche di sistemi complessi (dal 2013) trovano ampia utilità e complementarità con le attività svolte in ENEA dal gruppo del Dott. Stefano Pizzuti.

Ciascuna linea presenta peculiarità proprie che coinvolgono non solo il campo specifico delle misure, ma anche settori ad esso correlati quali l'elettronica, l'elettrotecnica, le telecomunicazioni, l'informatica e l'automazione. Il Laboratorio progetta e sviluppa sistemi di misura avvalendosi dei software più moderni come Orcad o Protel e programmando microcontrollori di varie famiglie come Microchip o Siemens, processori ARM, avendo confidenza anche con la progettazione di FPGA. I linguaggi di programmazione più usati sono il C, la piattaforma .NET e vari linguaggi "WEB oriented".

Il nostro team, avvalendosi anche di professionalità esterne, spazia anche in altri settori come la progettazione e realizzazione di impianti di energie alternative, l'illuminotecnica, la realizzazione e gestione di database, la progettazione e realizzazione di droni per attività subacquee e terrestri.

Nel campo della Didattica abbiamo nel tempo sviluppato dei percorsi all'interno dei nostri Dipartimenti rivolti al mondo dell'ambiente e dell'energia con materie come Qualità Ambientale, Qualità dell'Energia, Elementi di Misure per l'Analisi Ambientale, Alimentazione da Fonti Rinnovabili e Strumentazione Avanzata di Misura che, nel tempo, si sono state apprezzate da un numero crescente di studenti.

Il nostro lavoro ha permesso la creazione di link importanti come quelli con gruppi di ricerca dell'ENEA, con l'Istituto di Microsistemi e Microelettronica del CNR di Roma e con il Dipartimento di Mineralogia de La Sapienza.

A testimonianza dell'esperienza maturata, si riporta una breve selezione delle pubblicazioni più recenti (dal 2012) fatte dal Laboratorio e riguardanti i temi di ricerca citati, evidenziando in giallo i lavori sviluppati insieme ad ENEA. Più a valle, l'elenco dei premi vinti e l'elenco dei progetti fatti:

1) F. Leccese: "Remote-Control System of High Efficiency and Intelligent Street Lighting Using a ZigBee Network of Devices and Sensors," Power Delivery, IEEE Transactions on Volume: 28, Issue: 1, 2013, Page(s): 21 – 28, DOI: 10.1109/TPWRD.2012.2212215, ISSN: 0885-8977, IF: 1.353.

2) M. Caciotta, F. Leccese, S. Giannetti, S. Di Pasquale: "Geographical monitoring of Electrical Energy Quality determination: the problems of the sensors," Journal of Communication and Computer, Vol. 10, No. 12, 31 December 2013, pp. 1566-1572, David Publishing Company, ISSN 1930-1553. DOI: 10.17265/1934-8975. http://www.davidpublishing.org/journals_info.asp?jld=1919.

3) F. Marino, A. Capozzoli, M. Grossoni, F. Lauro, F. Leccese, F. Moretti, S. Pizzuti, S. Panzieri: "Indoor Lighting Fault Detection and Diagnosis Using a Data Fusion Approach," Proc. of "Energy Quest 2014," Vol.1, pp. 83-94, 23 -25 April, Ekaterinburg, Russia, WIT Press, ISBN: 978-1-84564-938-8, DOI: 10.2495/EQ140101. eid=2-s2.0-84897835365.

- 4) F. Leccese, M. Cagnetti, A. Calogero, D. Trinca, S. Di Pasquale, S. Giarnetti, L. Cozzella: "A New Acquisition and Imaging System for Environmental Measurements: an Experience on the Italian Cultural Heritage," *Sensors (Basel)* Special Issue on "Sensors for Cultural Heritage Diagnostics," 2014 May 23; 14(5):9290-312. doi: 10.3390/s140509290, IF: 2.245. <http://www.mdpi.com/1424-8220/14/5/9290>. WOS:000337112200088, eid=2-s2.0-84901350456.
- 5) M. Caciotta, S. Di Pasquale, S. Giarnetti, F. Leccese, D. Trinca: "A New Multi-Platform Data Acquisition System for Power Quality Metrological Certification," *Journal of Energy and Power Engineering* ISSN 1934-8975, USA, Vol. 8, No. 7, July 2014, David Publishing Company. <http://www.davidpublishing.org/show.html?17255>
- 6) F. Leccese, M. Cagnetti, D. Trinca: "A Smart City Application: A Fully Controlled Street Lighting System Is Based on Raspberry-Pi Card, a ZigBee Sensor Network and WiMAX Sensors - Special Issue on "Sensors and Smart Cities," 2014, 14(12), 24408-24424; DOI: 10.3390/s141224408, IF: 2.245. <http://www.mdpi.com/1424-8220/14/12/24408>. WOS:000346794300111, eid=2-s2.0-84919372229.
- 7) S. Giarnetti, F. Leccese, M. Caciotta: "Non Recursive Multiharmonic Least-Square Fitting for Frequency Estimation for Grid Frequency Estimation," *Measurement*, Volume 66, 2015, Pages 229-237, ISSN 0263-2241. DOI: 10.1016/j.measurement.2015.02.021; DOI: <http://dx.doi.org/10.1016/j.measurement.2015.02.021> IF: 1.484. WOS:000350825500026, eid=2-s2.0-84924058427.
- 8) Ed. De Francesco, Ett. De Francesco, R. De Francesco, F. Leccese, M. Cagnetti: "A Proposal to update LSA Databases for an Operational Availability based on Autonomic Logistic," *Proc. of Metrology for Aerospace*, 2nd IEEE International Workshop on, Benevento, Italy, June 3-5, 2015. Pag. 38-43. ISBN: 978-1-4799-7568-6, DOI: 10.1109/MetroAeroSpace.2015.7180623. WOS:000380405000008, eid=2-s2.0-84941351478.
- 9) M. Camponeschi, A. Fonti, F. Leccese, G. Comodi, M. Grossoni, S. Pizzuti: "Winter Thermal Multi-Objective Optimization: a Simulation Case Study," *International Journal of Engineering, Science and Innovative Technology-IJESIT*, Volume 4, Issue 4, Pages 1-7, July 2015, ISSN: 2319-5967.
- 10) M. Peroni, F. Dolce, J. Kingston, C. Palla, A. Fanfani, F. Leccese: "Reliability study for LEO satellites to assist the selection of End Of Life disposal methods," 2016 IEEE Metrology for Aerospace (MetroAeroSpace), Florence, Italy, 22-23 June 2016, pp. 141-145. doi: 10.1109/MetroAeroSpace.2016.7573201. eid=2-s2.0-84991833420. WOS:000389769800026.
- 11) R. Paggi, G. L. Mariotti, A. Paggi, A. Calogero, F. Leccese: "Prognostics via Physics-Based Probabilistic Simulation Approaches," 2016 IEEE Metrology for Aerospace (MetroAeroSpace), Florence, Italy, 22-23 June 2016, pp. 130-135. doi: 10.1109/MetroAeroSpace.2016.7573199. eid=2-s2.0-84991785890. WOS:000389769800024.
- 12) E. Petritoli, F. Leccese: "Reliability and SEE Mitigation in Memories for Space Applications," 2016 IEEE Metrology for Aerospace (MetroAeroSpace), Florence, Italy, 22-23 June 2016, pp. 136-140. doi: 10.1109/MetroAeroSpace.2016.7573200. eid=2-s2.0-84991740144. WOS:000389769800025.
- 13) V. Pasquali, R. Gualtieri, G. D'Alessandro, M. Granberg, D. Hazlerigg, M. Cagnetti, F. Leccese: "Monitoring and Analyzing of Circadian and Ultradian Locomotor Activity Based on Raspberry-Pi," *Electronics* 2016, 5(3), 58. EISSN 2079-9292. doi:10.3390/electronics5030058. WOS:000385490400007, eid=2-s2.0-84990062647.
- 14) F. Leccese, M. Cagnetti, S. Di Pasquale, S. Giarnetti, M. Caciotta: "A new Power Quality Instrument based on Raspberry-Pi," *Electronics* 2016, 5(4), 64. EISSN 2079-9292. doi: 10.3390/electronics5040064. eid=2-s2.0-84990050993.
- 15) S. Giarnetti, F. Leccese, M. Caciotta: "Non recursive Nonlinear Least Squares for periodic signal fitting," *Measurement*, Volume 103, 2017, Pages 208-216, ISSN 0263-2241; DOI: <http://dx.doi.org/10.1016/j.measurement.2017.02.023>. eid=2-s2.0-85014142799.
- 16) F. Marino, F. Leccese, S. Pizzuti: "Adaptive Street Lighting Predictive Control," *Energy Procedia*, Volume 111, Pages 790-799, March 2017, ISSN: 1876-6102; DOI: 10.1016/j.egypro.2017.03.241. eid=2-s2.0-85017266946.
- 17) F. Leccese, M. Cagnetti, S. Sciuto, A. Scorza, K. Torokhtii, E. Silva: "Analysis, design, realization and test of a sensor network for aerospace applications" (2017) I2MTC 2017 - 2017 IEEE International Instrumentation and Measurement Technology Conference, Proceedings, art. no. 7969946, . DOI: 10.1109/I2MTC.2017.7969946. eid=2-s2.0-85026811790. WOS:000431839600256.
- 18) V. Pasquali, G. D'Alessandro, R. Gualtieri, F. Leccese: "A new data logger based on Raspberry-Pi for Arctic Notostraca locomotion investigations" (2017) *Measurement: Journal of the International Measurement Confederation*, 110, pp. 249-256. DOI: 10.1016/j.measurement.2017.07.004. eid=2-s2.0-85022218699.
- 19) F. Leccese, M. Cagnetti, S. Tuti, P. Gabriele, E. De Francesco, R. Đurović-Pejčev, A. Pecora: "Modified LEACH for Necropolis Scenario" IMEKO International Conference on Metrology for Archaeology and Cultural Heritage, Lecce, Italy, October 23-25, 2017. ISBN: 978-92-990084-0-9. eid=2-s2.0-85048990580.
- 20) E. Petritoli, F. Leccese, M. Botticelli, S. Pizzuti, F. Pieroni: "'Smart Street' Pilot Site: A RAMS Analysis for a Scale-Up configuration" (2018) 2018 Workshop on Metrology for Industry 4.0 and IoT, MetroInd 4.0 and IoT 2018 - Proceedings, art. no. 8428338, pp. 129-133. DOI: 10.1109/METROI4.2018.8428338. eid=2-s2.0-85052530185.
- 21) E. Petritoli, F. Leccese, L. Ciani: "Reliability and Maintenance Analysis of Unmanned Aerial Vehicles," *Sensors* 2018, 18(9), 3171; <https://doi.org/10.3390/s18093171>. eid=2-s2.0-85053913129.
- 22) E. Petritoli, F. Leccese, S. Pizzuti, F. Pieroni: "Smart Lighting as basic building block of Smart City: an energy performance comparative case study," (2019) *Measurement: Journal of the International Measurement Confederation*. *Measurement*, Volume 136, Pages 466-477, ISSN 0263-2241, <https://doi.org/10.1016/j.measurement.2018.12.095>. <http://www.sciencedirect.com/science/article/pii/S0263224118312405>. eid=2-s2.0-85059534529.
- 23) E. Petritoli, F. Leccese, M. Botticelli, S. Pizzuti, F. Pieroni: "A RAMS analysis for a precision scale-up configuration of "Smart Street" pilot site: an Industry 4.0 Case Study," (2019) *Acta IMEKO*, 8 (2), pp. 3-11. DOI: 10.21014/acta_imeko.v8i2.614. eid=2-s2.0-85070330371.
- 24) Ett. De Francesco, R. De Francesco, F. Leccese: "Use of the S3000L for the Optimization of Projects in order to Reduce the Risk of Obsolescence of Complex Systems," 2019 6th IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace), June 19-21, 2019, Turin, Italy, pp. 233-237. DOI: 10.1109/MetroAeroSpace.2019.8869586. eid=2-s2.0-85074409926.
- 25) F. Leccese, M. Leccisi, M. Cagnetti: "Cluster Layout for an Optical Wireless Sensor Network for Aerospace Applications," 2019 6th IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace), June 19-21, 2019, Turin, Italy, pp. 556-561. DOI: 10.1109/MetroAeroSpace.2019.8869643. eid=2-s2.0-85074421424.

Si segnalano i premi nazionali/internazionali vinti dal Laboratorio

- 1) 2016 – PUBLONS - The sentinels of Science Awards 2016 The top 10 percent of reviewers- Certified Sentinel of Science award recipient: As one of the top 10 per cent of researchers contributing to the peer review of the field of Chemistry
- 2) 2018 – Il Forum Nazionale Delle Misure – Sezione GMEE – Padova, 17-19 Settembre 2018: Miglior Poster per l'articolo: "Measurements of Q factor in microwave resonators: relevance of the calibration" a cura di K. Torokhtii, A. Alimenti, N. Pompeo, F. Leccese, F. Orsini, A. Scorza, S.A. Sciuto, E. Silva.
- 3) 2018 – IEEE International Workshop on Metrology for the Sea, October 08-10, Bari, Italy: Miglior Demo per il drone di nuova concezione con movimentazione a pendolo vincolato a cura di Eduardo De Francesco e Fabio Leccese.
- 4) 2019 - WEB OF SCIENCE – PUBLONS -TOP PEER REVIEWER 2019 -For placing in the top 1% of reviewers in Cross-Field on Publons global reviewer database.

Elenco di partecipazioni a progetti scientifici (in giallo quelli in comun con ENEA)

Progetti Internazionali:

- 1) "PROGETTO DI GRANDE RILEVANZA ITALIA - SERBIA 2016-2018 sul tema di Agriculture and Food Technologies dal titolo **SMART MONITORING OF PESTICIDES IN FARMING AREAS**" Finanziato dal Ministero degli Affari Esteri e della Cooperazione Internazionale. **Ruolo: Responsabile Scientifico.** Durata 3 anni.

Progetti Nazionali:

- 2) Bando PROGRAMMI DI RICERCA SCIENTIFICA DI RILEVANTE INTERESSE NAZIONALE RICHIESTA DI COFINANZIAMENTO **PRIN 2010-2011** dal titolo: **"Interazione fra minerali e biosfera: conseguenze per l'ambiente e la salute umana"**- sottosezione **"Emissioni antropogeniche di CO2: immobilizzazione per carbonatazione e discriminazione isotopica della componente fossile e non fossile"**. PRIN 2010-2011, Area 04, Durata 36 mesi, Protocollo 2010 MKHT9B_007
- 3) Progetto **Co-Research POR FESR LAZIO 2007-2013 – Titolo SIMPLIFEX** Progetti di R&S in collaborazione presentati dalle PMI del Lazio con Numero di protocollo assegnato: FILAS-CR-2011-1076 dal 09/01/2012 al 08/01/2014. **Ruolo: Responsabile Scientifico di Sede.** Durata 2 anni.
- 4) Progetto di ricerca: **"Sviluppo e implementazione di algoritmi per applicazioni di Smart Lighting"** per conto di ENEA – Roma, 2014. **Ruolo: Responsabile Scientifico.** Durata 1 anno.
- 5) Progetto di ricerca: **"Sviluppo e implementazione di indicatori di prestazione e diagnostica energetica per impianti di illuminazione pubblica"** per conto di ENEA – Roma, 2014. **Ruolo: Responsabile Scientifico.** Durata 5 mesi.

I seguenti progetti sono stati sviluppati all'interno del piano Piano Triennale della Ricerca nell'ambito del Sistema Elettrico Nazionale 2015-2017 finanziato dal Ministero dello Sviluppo Economico (MiSE) e gestito da ENEA all'interno dell'Accordo di Programma MiSE-ENEA 2015-2017.

- 6) **Progettazione e sviluppo prototipale di strumenti per la gestione del PELL**, per conto di ENEA – Roma, 2015. **Ruolo: Responsabile Scientifico di Sede.** Durata 5 mesi.
- 7) **Analisi di affidabilità e analisi dei guasti e delle criticità (FMECA) del sistema smart street**, per conto di ENEA – Roma, 2016. **Ruolo: Responsabile Scientifico di Sede.** Durata 5 mesi.
- 8) **Studio affidabilistico dei componenti di una linea di illuminazione "smart" stradale pubblica operativa in contesto urbano: vantaggi e criticità**, per conto di ENEA – Roma, 2017. **Ruolo: Responsabile Scientifico di Sede.** Durata 5 mesi.
- 9) **Studio affidabilistico preliminare dei componenti fondamentali del sistema di termoregolazione dell'edificio F-40 ENEA (Casaccia): vantaggi e criticità**, per conto di ENEA – Roma, 2018. **Ruolo di Responsabile Scientifico di Sede.** Durata 3 mesi.

Progetti Conto Terzi:

- 10) Progetto di ricerca: **"Studio delle criticità delle PowerLine su Navi da guerra"** per conto della Se.Te.L. group di Roma, 2012. **Ruolo: Responsabile Scientifico.** Durata 1 mese.
- 11) Progetto di ricerca: **"Valutazione del Processo di Rivitalizzazione degli Accumulatori al Piombo- Acido e del Relativo Liquido Additivo"** per conto della Battery Equalizer Italia s.r.l. di Fiumicino, 2012, **Ruolo: Responsabile Scientifico.** Durata 3 mesi.
- 12) Progetto di ricerca: **"Evoluzioni del supporto logistico delle power line di unità navali"** per conto della Se.Te.L. group di Roma, 2013. **Ruolo: Responsabile Scientifico.** Durata 1 mese.