



Ricerca di Sistema elettrico

Progettazione, implementazione e test delle funzionalità del portale WEB “Ambiente Costruito”: un portale dinamico per soluzioni costruttive off-site

S. Serrano, F. Battaglia, M. Scarpa, G. Campobello

PROGETTAZIONE, IMPLEMENTAZIONE E TEST DELLE FUNZIONALITÀ DEL PORTALE WEB “AMBIENTE COSTRUITO”: UN PORTALE DINAMICO PER SOLUZIONI COSTRUTTIVE OFF-SITE

S. Serrano, F. Battaglia, M. Scarpa, G. Campobello
(Dipartimento di Ingegneria – Università degli Studi di Messina)

Aprile 2021

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dello Sviluppo Economico (oggi Ministero della Transizione Ecologica) - ENEA

Piano Triennale di Realizzazione 2019-2021 - II annualità

Obiettivo: N. 1 - Tecnologie

Progetto: 1.6 - Efficienza energetica dei prodotti e dei processi industriali

Work package: 2 - Miglioramento dell'efficienza energetica dei processi e di gestione dell'ambiente costruito

Linea di attività: LA 2.10 - Realizzazione di un portale web e creazione di una matrice dinamica delle opzioni

Responsabile del Progetto: Miriam Benedetti, ENEA

Responsabile del Work package: Maria-Anna Segreto, ENEA

Il presente documento descrive le attività di ricerca svolte all'interno dell'Accordo di collaborazione *“Sviluppo di una piattaforma ICT per soluzioni Off-Site Construction finalizzate alla riqualificazione energetica degli edifici”*

Responsabile scientifico ENEA: Maria-Anna Segreto

Responsabile scientifico Dipartimento di Ingegneria Università di Messina: Salvatore Serrano

Indice

SOMMARIO	5
INTRODUZIONE.....	6
PROGETTAZIONE DEL LAYOUT DELL'INTERFACCIA GRAFICA DEL PORTALE	8
INTRODUZIONE	8
LAYOUT DEL PORTALE	8
LAYOUT DELLA COLONNA DI NAVIGAZIONE	9
LAYOUT DEL CORPO CENTRALE	13
ARCHITETTURA SOFTWARE DEL PORTALE	16
INTRODUZIONE	16
IL DATABASE MySQL	17
<i>Le tabelle per la memorizzazione dell'Atlante</i>	18
<i>Le tabelle per la memorizzazione delle soluzioni costruttive e delle soluzioni "costruttive tipo"</i>	20
<i>Le tabelle di login e di log</i>	24
<i>La tabella dm_schema</i>	25
IL CODICE PHP	25
<i>Il file mysql.php</i>	25
<i>Il file myquery.php</i>	25
<i>Il file mygetdata.php</i>	25
<i>Il file mygetdatafromquery.php</i>	26
<i>Il file mygettonedata.php</i>	26
<i>Il file mygettableschemata.php</i>	27
<i>Il file myinsertdata.php</i>	27
<i>Il file mydeletedata.php</i>	27
<i>Il file myupdateX.php</i>	28
<i>Il file myIsUser.php</i>	28
IL CODICE HTML/JAVASCRIPT	28
<i>Il file index.html</i>	28
<i>Le funzioni del file do.js</i>	33
POPOLAMENTO DELL'ATLANTE CON I DATI FORNITI DA POLIMI PRODOTTI NELLA LA2.12.....	41
L'ACQUISIZIONE DELLE SCHEDE PRODOTTI	41
LA GENERAZIONE DELLE QUERY SQL DI CREAZIONE TABELLE E INSERIMENTO DATI	45
DESCRIZIONE DEL PORTALE.....	52
INTRODUZIONE	52
LA SEZIONE ATLANTE	52
<i>La sezione "Strutture" dell'Atlante</i>	52
<i>La sezione "Best Practices" dell'Atlante</i>	56
<i>La sezione "Aziende" dell'Atlante</i>	57
LA SEZIONE SOLUZIONI COSTRUTTIVE	58
LA SEZIONE MATRICE DINAMICA	65
PROCEDURA DI INSTALLAZIONE	68
INSTALLARE APACHE	68
INSTALLARE PHP	68
INSTALLARE MYSQL	69
CREARE E IMPORTARE IL DATABASE AMBIENTE COSTRUITO	70
INSTALLARE PHPMYADMIN	71
INSTALLAZIONE DEI FILE DEL PORTALE E CREAZIONE DELL'HOST VIRTUALE	72
RIFERIMENTI BIBLIOGRAFICI	74

Sommario

Le attività svolte nell'ambito dell'accordo di collaborazione LA2.10 nel periodo lug'20-apr'21 illustrate in questo rapporto tecnico hanno riguardato la realizzazione di un portale WEB denominato "Ambiente Costruito" che ospita un atlante di soluzioni elaborato da POLIMI nella LA2.12 e una matrice dinamica delle opzioni. Il rapporto tecnico descrive la progettazione e una descrizione delle funzionalità del portale, il codice di programmazione implementato e le procedure di installazione.

1 Introduzione

Il presente report tecnico descrive le attività svolte nell'ambito di un accordo di collaborazione tra ENEA e il Dipartimento di Ingegneria dell'Università degli Studi di Messina per il Piano Triennale di Realizzazione 2019-2021, per quanto attiene l'Obiettivo "Tecnologie"; nello specifico il tema dell'accordo di collaborazione (linea di attività LA2.10) ha riguardato la creazione di un portale web per la raccolta dei risultati del Work Package "WP2 – Miglioramento dell'efficienza energetica dei processi e di gestione dell'ambiente costruito" e lo sviluppo di strumenti per l'elaborazione e la visualizzazione dei dati.

Il portale WEB realizzato, attualmente disponibile presso un server dell'Università degli Studi di Messina all'indirizzo <http://eneadi.unime.it/portale> ma che, appena disponibile, sarà allocato presso un server ENEA all'indirizzo <http://www.ambientecostruito.it>, intende essere uno strumento fondamentale di ausilio per il progettista di soluzioni per il miglioramento dell'efficienza energetica degli edifici.

Più precisamente, le attività oggetto della linea di attività LA2.10 hanno riguardato:

- la progettazione e implementazione di un database relazionale per l'immagazzinamento e la gestione di tutti i dati al fine di consentirne l'inserimento, la consultazione e l'aggiornamento ed in particolare:
 - un atlante delle soluzioni (elaborato da POLIMI nella LA2.12) contenente diverse tipologie e categorie di prodotti con le relative schede tecniche, un elenco di soluzioni costruttive denominate "Best Practices" con le relative schede tecniche, una rubrica delle aziende produttrici con le relative schede tecniche;
 - le diverse soluzioni costruttive e specifiche combinazioni delle stesse denominate "soluzioni costruttive tipo";
 - le strutture dati necessarie alla realizzazione di una "matrice dinamica delle scelte" in grado di individuare le "soluzioni costruttive tipo" che soddisfano specifici requisiti di tipologia costruttiva, di tipologia di impianti e di categoria climatica e sismica;
 - le informazioni relative alle interrogazioni fatte dagli utenti alla "matrice dinamica delle scelte" al fine di poter registrare e archiviare i dati di ricerca inseriti dagli utenti per poter estrarre informazioni statistiche future;
 - le informazioni degli utenti (username e password) che possono accedere al portale in modalità di amministratore (abilitati cioè a fare l'inserimento di nuovi dati e la modifica o cancellazione dei dati già inseriti);
- la progettazione e la realizzazione del layout, del codice HTML, del codice JavaScript, del codice PHP e delle query SQL per l'implementazione dell'interfaccia grafica dell'applicazione, per l'interrogazione, la modifica e la cancellazione dei dati contenuti nel database e per l'implementazione della matrice dinamica delle scelte;
- le operazioni di elaborazione ed inserimento di tutti i dati contenuti nell'atlante delle soluzioni elaborato da POLIMI nella LA2.12;
- il test e il debug dell'applicazione.

Il portale è stato realizzato utilizzando la tecnologia messa a disposizione da un server di tipo LAMP (Linux, Apache, MySQL, PHP/Perl/Python) ed in particolare:

- server Linux con distribuzione Ubuntu versione 20.04.2 LTS;
- MySQL versione 8.0.22;
- Apache versione 2.4.41;
- PHP version 7.4.3.

L'interazione con il browser client dell'utente per la visualizzazione dell'interfaccia grafica è stata realizzata utilizzando il linguaggio JavaScript ed in particolare le librerie W3.JS.

Le funzionalità suddette sono descritte nelle successive sezioni del presente report. In particolare:

- nella Sez. 2 sarà descritta la progettazione del layout dell'interfaccia grafica del portale;
- nella Sez. 3 sarà descritta l'architettura software del portale ed in particolare nell'ordine la struttura del database MySQL, il codice PHP e il codice JavaScript;
- nella Sez. 4 saranno descritte le metodologie utilizzate per estrarre i dati presenti nei PDF rilasciati da POLIMI nella LA2.12 e popolare la sezione denominata "Atlante" del database;
- nella Sez. 5 sarà descritto l'intero portale con le diverse funzionalità disponibili agli utenti semplici e agli amministratori;
- infine, nella Sez. 6 sarà descritta la procedura di installazione.

2 Progettazione del layout dell'interfaccia grafica del portale

2.1 Introduzione

La prima attività effettuata ha riguardato la progettazione dell'interfaccia grafica del portale in modo da soddisfare i requisiti di facile e intuitiva individuazione del suo contenuto da parte degli utenti. Il layout di un portale è un aspetto di notevole importanza. Infatti, per creare un portale che sia uno strumento efficace di comunicazione è necessario disporre nel miglior modo possibile tutti gli elementi presenti in ogni pagina del sito. Una pagina di un sito, infatti, non deve apparire come un banale contenitore riempito da informazioni, ma deve essere uno strumento ben ingegnerizzato che serve a comunicare con l'utente che la consulta. Secondo Krug, autore del libro sulla "web usability" dal titolo "Don't make me think" [1], chi visualizza un portale deve capire subito dove si trova e che cosa può fare, senza doverci pensare. Nella mente del nostro visitatore non devono sorgere domande su interfaccia, struttura del sito, caratteristiche dei componenti della pagina.

Nello specifico, per la progettazione del layout del portale ci siamo sforzati di:

- costruire un layout semplice e immediatamente comprensibile al visitatore;
- mantenere un equilibrio tra spazi pieni e vuoti della pagina;
- mantenere un layout omogeneo per tutto il portale in maniera tale che tutte le pagine appariranno come parte dello stesso portale;
- garantire una fruibilità ottimale per dispositivi diversi di visualizzazione (smartphone, tablet, laptop, PC).

2.2 Layout del portale

Il layout realizzato prevede la presenza di un unico corpo centrale (main body) che svolge le funzioni di contenitore per tutti i contenuti informativi. Attraverso un'unica colonna di navigazione presente (quando lo schermo del dispositivo è di dimensioni sufficienti) alla sinistra del corpo centrale, il visitatore potrà selezionare i diversi contenuti del portale. Qualora lo schermo del dispositivo dovesse essere di dimensioni ridotte (come, ad esempio, il display di uno smartphone), la colonna di navigazione di sinistra scompare e risulta accessibile attraverso un pulsante di menu posto in alto a destra dell'intera pagina.

Il main body è suddiviso in due diverse sezioni: la prima, in alto, conterrà l'intestazione per permettere al visitatore di identificare costantemente per tutto il periodo di navigazione il nome del portale che sta utilizzando; la seconda, centrale, conterrà il contenuto informativo vero e proprio dei diversi elementi selezionati attraverso la colonna di navigazione. La Figura 1 visualizza il layout progettato che sarà fruibile quando si naviga il portale utilizzando dispositivi con schermi di grandi dimensioni mentre la Figura 2 visualizza il layout progettato che sarà fruibile quando si naviga il portale utilizzando un dispositivo con schermo di piccole dimensioni (come, ad esempio, uno smartphone).

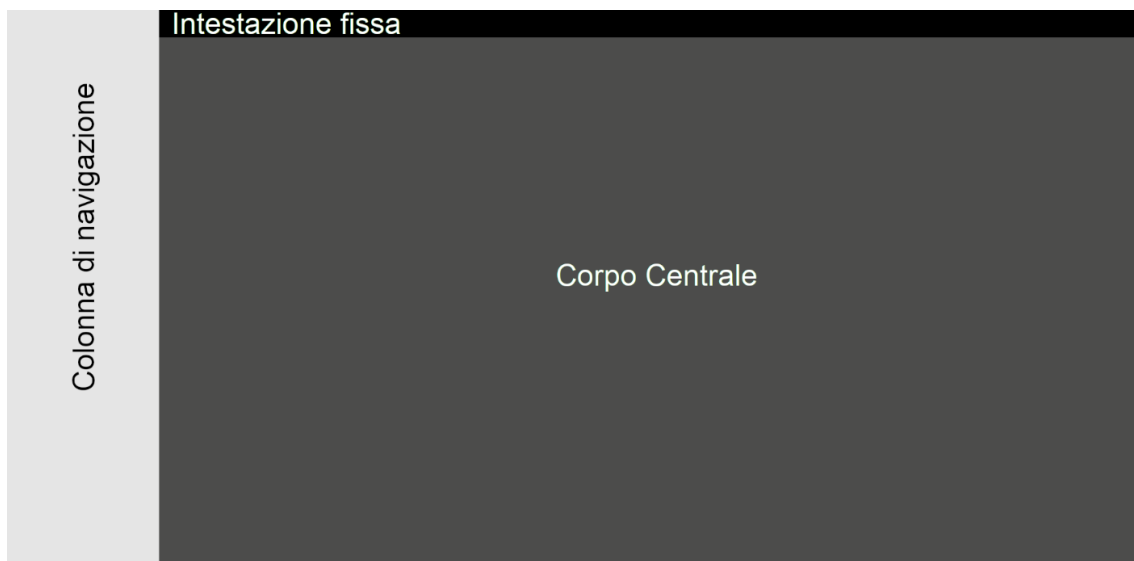


Figura 1: layout del portale su schermi di grandi dimensioni

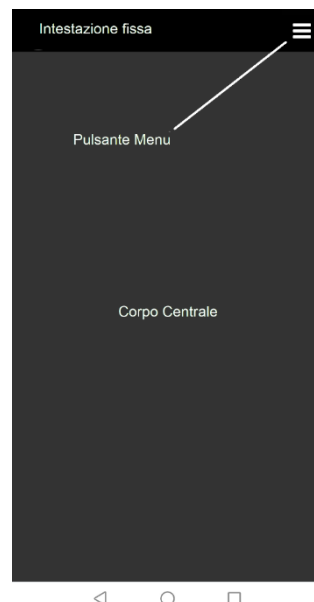


Figura 2: layout del portale su schermi di piccole dimensioni

Si osservi che nel caso in cui si utilizzasse un dispositivo con schermo di grandi dimensioni ma si decidesse di diminuire le dimensioni della finestra del browser al di sotto del minimo consentito, il layout commuta automaticamente a quello progettato per schermi di piccole dimensioni, al fine di non ridurre eccessivamente lo spazio destinato al corpo centrale.

2.3 Layout della colonna di navigazione

La colonna di navigazione, mostrata in Figura 3, conterrà, nella parte alta, un'icona che rappresenta l'immagine del logo del progetto e, successivamente in sequenza, gli elementi che consentono di visualizzare nel corpo centrale:

- la pagina di login come amministratore del sistema;
- la home page contenente le informazioni generali sul portale;

- le pagine contenenti le informazioni presenti nell'atlante delle soluzioni: in questo caso sarà possibile scegliere attraverso l'apertura di tre diversi elementi se visualizzare le "strutture costruttive", le soluzioni costruttive denominate "Best Practices" o la rubrica delle aziende produttrici (Figura 4); in particolare le "strutture costruttive" presenti nell'atlante sono suddivise per tipologia e categoria, entrambe selezionabili per mezzo di appositi elementi di selezione dedicati (drop-down list) che saranno visualizzati agendo sull'apposita voce di menu "Strutture" (Figura 5);
- le pagine contenenti le informazioni relative alle diverse soluzioni costruttive, selezionabili per mezzo dell'elemento di selezione dedicato (drop-down list) che sarà visualizzato agendo sull'elemento denominato "Soluzioni Costruttive" (Figura 6);
- la pagina che permette di effettuare la selezione delle "soluzioni costruttive tipo" in base alle condizioni costruttive definite dall'utente e alle informazioni contenute nella matrice dinamica delle soluzioni;
- la pagina dei contatti.

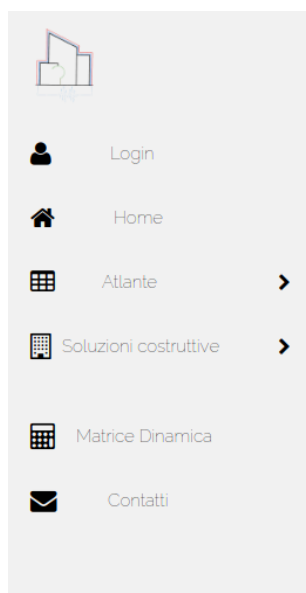


Figura 3: colonna di navigazione

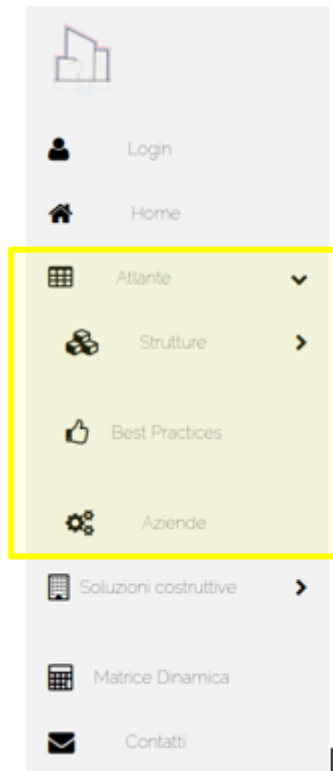


Figura 4: colonna di navigazione con, evidenziata, l'apertura dell'Atlante

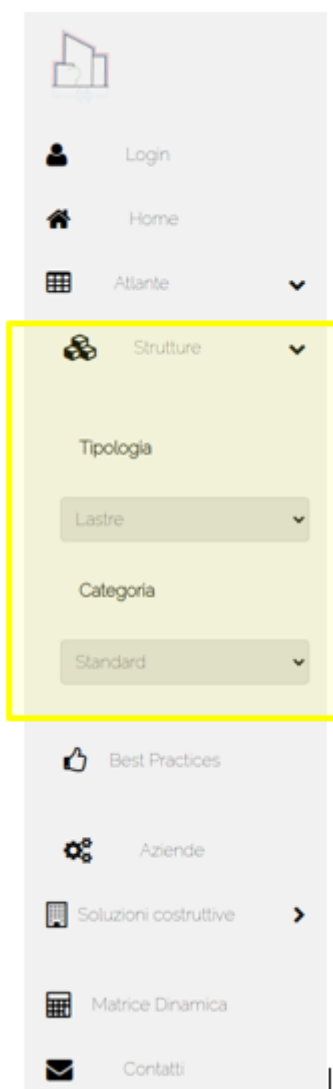


Figura 5: colonna di navigazione con, evidenziata, l'apertura dell'Atlante e dell'elemento Strutture

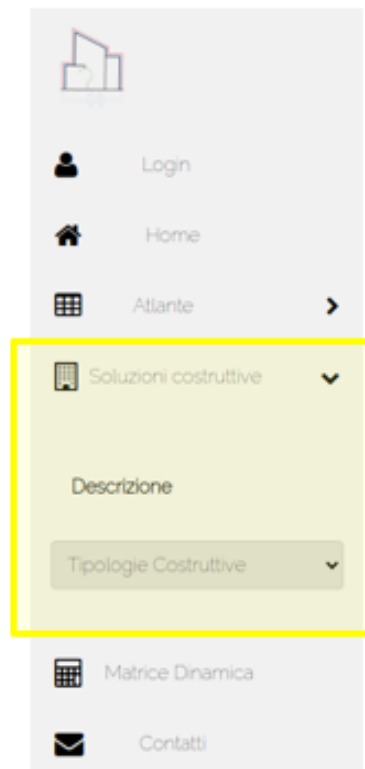


Figura 6: colonna di navigazione con, evidenziata, l'apertura delle Soluzioni Costruttive

2.4 Layout del corpo centrale

Il layout del corpo centrale varia in funzione della tipologia di informazioni che devono essere visualizzate. Tuttavia, nella maggior parte dei casi, la visualizzazione riguarderà una tabella contenente i campi relativi ai dati che si vogliono visualizzare. Poiché si è deciso di rendere quanto più omogenea possibile la visualizzazione del portale, tutte le pagine che mostrano contenuti informativi in forma tabellare ricalcheranno lo stesso layout. Fanno eccezione la pagina di login, la home page, la pagina dei contatti e, in parte, la pagina relativa alla matrice dinamica.

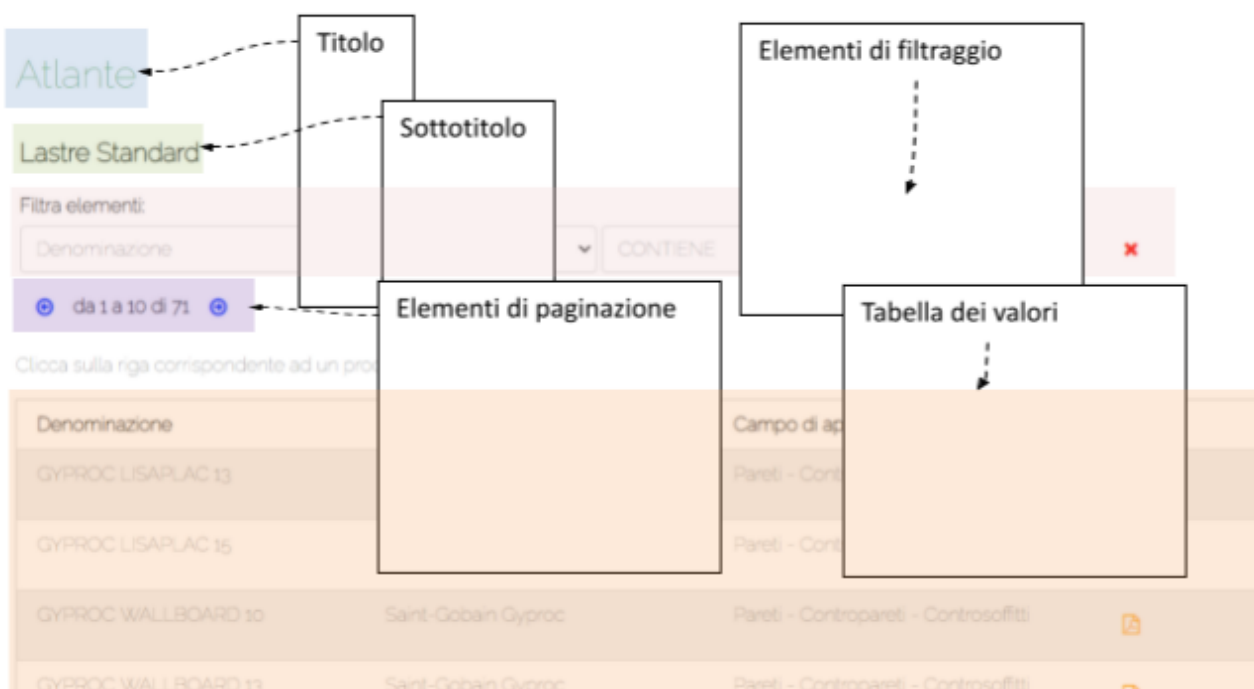


Figura 7: Layout del corpo centrale per la visualizzazione dei valori tabellari

La Figura 7 visualizza il layout progettato per il corpo centrale nel caso in cui gli elementi da visualizzare sono di tipo tabellare. In particolare, si è scelto di inserire all’inizio della pagina due elementi che visualizzano il “titolo” della stessa (determinato dall’elemento di navigazione scelto, nel caso in figura “Atlante”) e il sottotitolo contenente informazioni dettagliate sulla tabella visualizzata (nel caso in figura “Lastre Standard”). Al di sotto del sottotitolo sarà presente la sezione che permetterà l’eventuale selezione degli elementi visualizzati in tabella attraverso l’applicazione di una combinazione di filtri sui campi.



Per evitare che il numero delle righe da visualizzare all’interno della tabella sia troppo elevato, è stato scelto di partizionare la tabella in blocchi contenenti al massimo 10 righe (tale numero è una variabile modificabile in fase di installazione del portale). Inoltre, sono stati posizionati, al di sotto degli elementi di filtraggio, degli elementi che permettono di visualizzare i diversi blocchi della tabella stessa. I dati veri e propri saranno visualizzati in una struttura tabellare, presente al di sotto di tali elementi di selezione blocco.

Si è deciso di visualizzare le tabelle mostrando un numero ridotto di campi (tipicamente non più di 3) in maniera tale da facilitarne la lettura. Gli ulteriori campi di ogni riga possono essere visualizzati facendo click sullo stesso. Questa operazione permette infatti di ingrandire le dimensioni di ogni riga per visualizzare le informazioni aggiuntive come mostrato in Figura 8.

Ambiente costruito

da 1 a 10 di 71

Clicca sulla riga corrispondente ad un prodotto per conoscerne i dettagli.

Denominazione	Azienda	Campo di applicazione	
GYPROC LISAPLAC 13	Saint-Goban Gyproc	Pareti - Cortopareti - Controsoffitti	
GYPROC LISAPLAC 25 Composizione: Gesso rivestito Dimensioni Spessore [mm]: 25 Dimensioni Larghezza [mm]: 1200 Dimensioni Lunghezza [mm]: 3000 Bordi Longitudinale: Assottigliato Bordi Di testa: Dritto Peso [kg/m ²]: 11,60 Densità [kg/m ³]: 750 Percentuale di foratura: - Classe di reazione al fuoco: A1 Carico di rottura a flessione Longitudinale [N]: 650 Carico di rottura a flessione Trasversale [N]: 250 Resistenza a flessione Longitudinale [N/mm ²]: - Resistenza a flessione Trasversale [N/mm ²]: - Modulo elastico [N/mm ²]: - Durezza Superficiale [mm]: - Durezza Brinell [N/mm ²]: - Conduttività termica [W/mK]: 0,21 Fattore di resistenza alla diffusione di vapore in campo secco: 10 Fattore di resistenza alla diffusione di vapore in campo umido: 4 Assorbimento d'acqua Superficiale [g/m ²]: - Assorbimento d'acqua Totale [g]: - Riduzione formaldeide: - Note: -	Saint-Goban Gyproc	Pareti - Cortopareti - Controsoffitti	

Inoltre, su ogni riga, in corrispondenza dell'ultima colonna, si è scelto di inserire le icone che permettono l'apertura della documentazione tecnica associata all'elemento visualizzato nella riga e, nel caso in cui l'operatore abbia effettuato il login da amministratore, le icone che permettono di effettuare le operazioni di modifica dei dati e/o cancellazione dell'elemento stesso.

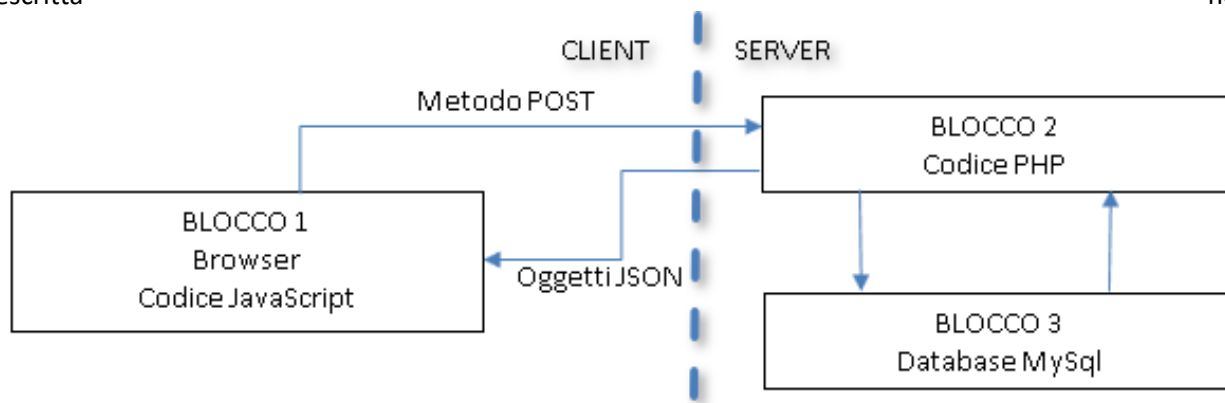
3 Architettura software del portale

3.1 Introduzione

Il software del portale è strutturato in tre diversi blocchi funzionali:

- il primo, scritto in codice HTML (HyperText Markup Language) [2] e JavaScript [3], si occupa della visualizzazione dell'interfaccia grafica e dell'interazione con le operazioni effettuate dall'utente sul client, della realizzazione dinamica delle pagine dell'interfaccia grafica da far apparire sul browser del client, e dell'interazione, attraverso il secondo blocco scritto in PHP, con il database relazionale (per leggerne e/o modificarne il contenuto). Per la realizzazione dinamica delle pagine dell'interfaccia grafica sfrutta il framework w3.css (l'acronimo CSS indica Cascading Style Sheets, un linguaggio che consente di formattare documenti HTML, XHTML e XML usati nelle pagine web) di W3School [4]. Tale framework consente la realizzazione semplificata di tabelle e di effetti grafici nelle pagine WEB. L'esecuzione di tale codice avviene nella macchina client dell'utente che sta navigando nel portale.
- il secondo, scritto in PHP [5], si occupa di interagire con il database e di inserirne/modificarne/estrarne i valori. L'interazione verso il codice scritto in JavaScript è realizzata utilizzando il metodo POST. Utilizzando il metodo POST i dati saranno inviati, incapsulati all'interno di una richiesta HTTP che il browser che gira sulla macchina client invia al server, in maniera da non essere direttamente visibili per l'utente. I dati saranno restituiti al codice JavaScript sotto forma di array di oggetti codificato in formato JSON (JavaScript Object Notation) [6]. Il codice PHP del secondo blocco sarà eseguito all'interno del server.
- Il database è stato realizzato utilizzando MySQL [7], strutturato in maniera tale da contenere tutti i dati e le informazioni necessarie per il funzionamento del portale. Il database risiede sulla macchina server.

Da un punto di vista logico tutte le operazioni funzionali sono eseguite attraverso l'interazione fra i tre blocchi descritti nella



Tutto il codice è stato progettato in maniera tale da garantire che i contenuti dei dati fruibili sul sito possano variare in maniera sufficientemente dinamica. Tale dinamicità si è resa necessaria per due diverse esigenze: 1) la struttura e il numero delle tabelle da visualizzare non era perfettamente definita nel momento in cui è stata iniziata la progettazione del portale; 2) garantire una relativa versatilità del portale tale da permetterne la possibilità di espansione/modifica senza la necessità di dover scrivere nuovo codice JavaScript o PHP ma semplicemente inserendo nel database le strutture dati necessarie (tabelle) o modificando il contenuto di quelle già esistenti.

Il codice sorgente è contenuto in cinque diverse cartelle:

- La cartella *css* contiene i file *loader.css* e *w3.css*. Il file *w3.css* contiene il codice sorgente del framework CSS di W3School e fa internamente uso di JavaScript consentendo in tal modo la realizzazione di pagine web di tipo dinamico.
- La cartella *font* contiene i file relativi ai webfont utilizzati per la visualizzazione dal portale. I webfont sono scaricati dal browser in fase di caricamento della pagina. Essi vengono richiamati dalla libreria CSS in uso (nel nostro caso *w3.css*). Il portale fa uso di due webfont liberamente disponibili:
 - *font-awesome* (nella sottocartella *font/font-awesoe*) [8];
 - *raleway-webfont* (nella sottocartella *font/raleway*) [9].
- La cartella *images* contiene le immagini usate dal browser per la visualizzazione del portale (come le icone o il logo UniME).
- La cartella *js* contiene le librerie JavaScript che vengono usate internamente dal browser per l'implementazione delle pagine web dinamiche (eventi, interrogazione db) e che permettono di semplificare lo sviluppo della pagina web e il file che contiene le funzioni JavaScript opportunamente scritte per il funzionamento del portale:
 - Il file *jquery-3.4.1-min* contiene la libreria jQuery;
 - Il file *w3.js* contiene la libreria W3.JS di W3School;
 - Il file *do.js* contiene le funzioni JavaScript scritte per il funzionamento del portale.
- La cartella *service* contiene gli script e le funzioni PHP che vengono eseguiti dal server durante la visualizzazione delle pagine del portale ed in particolare i seguenti file la cui funzionalità sarà illustrata dettagliatamente nell'apposita sezione:
 - *mysql.php*;
 - *myquery.php*;
 - *mygetdata.php*;
 - *mygetdatafromquery.php*;
 - *mygettablescheema.php*;
 - *mygetonedata.php*;
 - *mylsUser.php*;
 - *myinsertdata.php*;
 - *mydeletedata.php*;
 - *myupdateX.php*.
- La cartella *pdf* conterrà, organizzati per sottocartelle, i file pdf relativi alle schede tecniche dei diversi prodotti. Sarà creata una sottocartella per ogni tabella contenente elementi avente lo stesso nome della tabella. All'interno di ogni sottocartella i file saranno memorizzati con nomi nella forma *<id>.pdf*, dove *<id>* rappresenta la chiave univoca della riga che contiene il relativo elemento.

3.2 Il database MySQL

Le informazioni visualizzate dal portale sono contenute in un database MySQL che opera in sinergia con il codice HTML/PHP/JavaScript eseguito dal server Apache e dal browser del client.

Il database è stato progettato per contenere tre tipi di tabelle:

- Le tabelle per la memorizzazione delle informazioni provenienti dalla linea di attività LA2.12 elaborate da POLIMI;
- Le tabelle per la memorizzazione delle tipologie costruttive e per implementare le funzionalità della matrice dinamica che permette di selezionare le "soluzioni costruttive tipo";
- Le tabelle per la memorizzazione delle informazioni di accesso degli utenti "amministratore" e le informazioni riguardanti le scelte effettuate dall'utente nell'applicazione della matrice dinamica al fine di poterne estrarre in futuro informazioni di tipo statistico.

Per la comprensione della metodologia che è stata usata nella realizzazione della struttura del database è fondamentale considerare che, al momento in cui se ne è iniziata la progettazione, le tipologie di informazioni da memorizzare non erano ancora ben consolidate. Inoltre, si è scelto di voler rendere dinamica la struttura del database in maniera tale che esso possa essere modificato (sia in termini di tabelle sia in termini di campi

che le tabelle stesse possono contenere) mantenendone comunque l’accessibilità da parte del codice PHP/JavaScript scritto per la realizzazione del portale.
 Il database realizzato in MySQL è stato denominato “ambiente_costruito”.

3.2.1 Le tabelle per la memorizzazione dell’Atlante

Come descritto nella sezione introduttiva l’atlante è una raccolta di prodotti (denominati d’ora in avanti “elementi funzionali”), aziende produttrici di questi elementi funzionali (denominate d’ora in avanti “aziende”) e una raccolta di soluzioni costruttive (denominate “best practices”) elaborate da POLIMI in seno alla linea di attività LA2.12.

Il portale è stato progettato in maniera tale che il numero degli “elementi funzionali” memorizzabili e visualizzabili da parte dell’utente non sia prefissato. Per questo motivo è stata realizzata una prima tabella denominata *elementifunzionali* la cui struttura è visualizzata in Figura 10

ambiente_costruito elementifunzionali	
id	int
Tipologia	text
nometabella	varchar(20)

Figura 10: tabella elementi_funzionali

Nella tabella è presente un campo intero *id* che ne costituisce la chiave primaria, un campo *Tipologia* di tipo testo in cui sarà memorizzata la descrizione dell’elemento funzionale (ad esempio “lastre”, “pannelli”, ...) e un campo *nometabella* di tipo caratteri in cui sarà memorizzato il prefisso del nome della tabella che conterrà i dati relativi a tali elementi funzionali.

Le informazioni relative ad uno specifico elemento funzionale possono inoltre essere logicamente separate in diverse categorie (ad esempio “standard”, “protezione dal fuoco”, ...). Ogni elemento funzionale può avere associato un insieme di categorie che è diverso rispetto all’insieme delle categorie relativo agli altri elementi funzionali. È stata quindi realizzata una seconda tabella denominata *elementifunzionali* in cui, per ogni categoria, saranno memorizzati l’*id* dell’elemento funzionale di riferimento nel campo *idelementofunzionale*, un carattere identificativo della categoria nel campo *idtipo* e la descrizione della categoria nel campo *Categoria* di tipo testo.

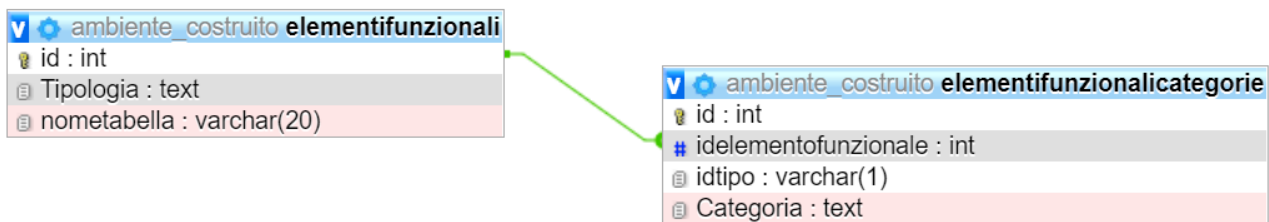


Figura 11: tabella elementifunzionali e sua relazione con la tabella elementifunzionali

L’informazione memorizzata nel campo *idtipo* sarà utilizzata come suffisso del nome della tabella che conterrà i dati relativi allo specifico elemento funzionale determinato dall’associazione dell’elemento funzionale e della categoria. Utilizzando una query di selezione che sfrutta il costrutto LEFT JOIN del tipo

```
SELECT `Tipologia`, `Categoria`, CONCAT(`nometabella`,`_`,`idtipo`) AS `Nome Tabella`
FROM `elementifunzionali`
LEFT JOIN `elementifunzionali`
ON `elementifunzionali`.`idelementofunzionale`=`elementifunzionali`.`id`
```

sarà possibile recuperare i diversi elementi funzionali con le diverse categorie e i relativi nomi delle tabelle associate come, a titolo di esempio, è illustrato nella Figura 12.

Tipologia	Categoria	Nome Tabella
Lastre	Standard	lastre_A
Lastre	Idrorepellenti	lastre_B
Lastre	Protezione dal fuoco	lastre_C
Lastre	Alta resistenza	lastre_D
Lastre	Acustiche	lastre_E
Lastre	Per sottofondi	lastre_F
Lastre	Per esterni	lastre_G
Pannelli e doghe	Standard	pannelli_A
Pannelli e doghe	Idrorepellenti	pannelli_B
Pannelli e doghe	Protezione dal fuoco	pannelli_C
Pannelli e doghe	Acustici	pannelli_D
Pannelli e doghe	Irrigidimento	pannelli_E
Pannelli Sandwich		pannellisandwich_A
Isolamento	Di origine vegetale	isolamento_A

Figura 12: Estratto del risultato della query di selezione per recuperare gli elementi funzionali e le relative categorie con i nomi delle tabelle associate

Considerate tutte le possibili tipologie di elementi funzionali con le rispettive categorie sarà necessario inserire all'interno del database le rispettive tabelle che ne conterranno i campi descrittivi.

Ciascuna tabella, come deciso in fase di progettazione, potrà contenere un numero diverso di campi contenenti diverse tipologie di dati, in modo tale da poter memorizzare le informazioni utili per quella specifica categoria di elemento funzionale considerato.

A titolo di esempio, nella Figura 13, sono illustrate alcune delle strutture delle tabelle relative ad elementi funzionali presenti in atlante. Sulla base dei dati forniti da POLIMI sono state attualmente inserite nel database 31 diverse tabelle.

ambiente_costruito isolamento_A	ambiente_costruito lastre_B	ambiente_costruito pannelli_C
id : int	id : int	id : int
Denominazione : text	Denominazione : text	Denominazione : text
Azienda : text	Azienda : text	Azienda : text
Formato : text	Campo di applicazione : text	Campo di applicazione : text
Campo di applicazione : text	Composizione : text	Composizione : text
Composizione : text	Dimensioni Spessore [mm] : text	Dimensioni Spessore [mm] : text
Dimensioni Spessore [mm] : text	Dimensioni Larghezza [mm] : text	Dimensioni Larghezza [mm] : text
Dimensioni Larghezza [mm] : text	Dimensioni Lunghezza [mm] : text	Dimensioni Lunghezza [mm] : text
Dimensioni Lunghezza [mm] : text	Bordi Longitudinale : text	Bordi Longitudinale : text
Densità [kg/m3] : text	Bordi Di testa : text	Bordi Di testa : text
Fonoisolamento acustico [dB] : text	Peso [kg/m2] : text	Peso [kg/m2] : text
Isolamento al calpestio [dB] : text	Densità [kg/m3] : text	Densità [kg/m3] : text
Assorbimento acustico : text	Percentuale di foratura : text	Percentuale di foratura : text
Classe di reazione al fuoco : text	Classe di reazione al fuoco : text	Foratura : text
Resistenza a rottura Longitudinale [N/mm2] : text	Carico di rottura a flessione Longitudinale [N] : text	Assorbimento acustico : text
Resistenza a rottura Trasversale [N/mm2] : text	Carico di rottura a flessione Trasversale [N] : text	Classe di reazione al fuoco : text
Resistenza a trazione Trasversale [kPa] : text	Resistenza a flessione Longitudinale [N/mm2] : text	Resistenza a flessione Longitudinale [N/mm2] : text
Resistenza a taglio [kPa] : text	Resistenza a flessione Trasversale [N/mm2] : text	Resistenza a flessione Trasversale [N/mm2] : text
Resistenza a flessione [kPa] : text	Modulo elastico [N/mm2] : text	Modulo elastico Longitudinale [N/mm2] : text
Resistenza a compressione [kPa] : text	Durezza Superficiale [mm] : text	Modulo elastico Trasversale [N/mm2] : text
Rigidità dinamica [MN/m3] : text	Durezza Brinell [N/mm2] : text	Resistenza a trazione Trasversale [N/mm2] : text
Conducibilità termica [W/mK] : text	Conducibilità termica [W/mK] : text	Rigonfiamento spessore 24h : text
Calore specifico [J/kgK] : text	Fattore di resistenza alla diffusione di vapore In campo secco : text	Conducibilità termica [W/mK] : text
Fattore di resistenza alla diffusione di vapore : text	Fattore di resistenza alla diffusione di vapore In campo umido : text	Fattore di resistenza alla diffusione di vapore In campo secco : text
Permeabilità al vapore [m] : text	Assorbimento d'acqua Superficiale [g/m2] : text	Fattore di resistenza alla diffusione di vapore In campo umido : text
Permeabilità al vapore [mg/m h Pa] : text	Assorbimento d'acqua Totale [%] : text	Riflessione della luce : text
Assorbimento d'acqua BT/LT [kg/m2] : text	Riduzione formaldeide : text	Resistenza all'umidità : text
Permeabilità all'aria [kPa s/m2] : text	Note : text	Riduzione formaldeide : text
Resistenza all'acqua : text		Note : text
Emissività est:int : text		
Note : text		

Figura 13: Struttura di alcune delle tabelle relative agli elementi funzionali presenti in atlante

La struttura dinamica del portale permette di poter aggiungere/eliminare elementi funzionali, aggiungere/eliminare le categorie da prendere in considerazione per ogni elemento funzionale, aggiungere/eliminare il numero di campi che descrivono ciascuna associazione elementofunzionale-

categoria o eventualmente variarne la tipologia dei singoli campi, tutto senza dover modificare il codice JavaScript/PHP.

Per permettere la costruzione automatica delle pagine di inserimento/modifica degli elementi contenuti all'interno di ogni tabella è necessario associarvi una tabella descrittiva dei campi che permetta al codice di creare il corretto elemento di input. Tali tabelle di associazione avranno lo stesso nome della tabella di origine dei dati con l'aggiunta del suffisso "_schema".

Nella Figura 14, è illustrata la generica struttura di una tabella di tipo "schema". Si può notare la presenza di una chiave univoca *id*, di un campo *nome_campo* dove saranno memorizzati i nomi dei diversi campi che costituiscono la tabella, di un campo *tipo_campo* che conterrà la tipologia di visualizzazione dei campi che costituiscono la tabella da utilizzare in fase di inserimento/modifica, può assumere i valori *TEXT*, *SELECT*, *NUMERIC*, *SELECTCHECK*. Il valore *TEXT* implica che il campo può essere acquisito attraverso una semplice casella di testo. Il valore *SELECT* implica che il campo dovrà essere acquisito attraverso una drop-down list. Il valore *NUMERIC* implica che il campo potrà essere acquisito attraverso una casella di testo ma che i valori dovranno essere esclusivamente numerici. Il valore *SELECTCHECK* sarà successivamente spiegato in questa stessa sezione. I campi *minimo* e *massimo* possono essere sfruttati nel caso in cui la tipologia è di tipo *NUMERIC* se in fase di inserimento/modifica dei dati si volessero segnalare ed eventualmente impedire inserimenti di valori fuori range per quel campo. Il campo *notempty* specifica che quel campo non può essere mantenuto vuoto e quindi deve essere necessariamente assegnato un valore (tipicamente utile quando all'utente è lasciata la libertà di inserire un codice univoco che rappresenta quell'inserimento). Il campo *querystring* specifica l'eventuale query di selezione da utilizzare nel caso in cui la tipologia sia di tipo *SELECT*: la query sarà eseguita sul database per prelevare i valori e il testo con cui riempire la drop-down list.

ambiente_costruito isolamento_F_schema	
id	int
nome_campo	text
tipo_campo	text
minimo	double
massimo	double
notempty	tinyint(1)
querystring	text

Figura 14: struttura di una generica tabella di tipo "schema"

Oltre alla memorizzazione degli elementi funzionali l'atlante conterrà anche le due tabelle (dati e schema) per memorizzare le "best practices" e le due tabelle (dati e schema) per memorizzare la rubrica delle aziende.

ambiente_costruito bestpractices		ambiente_costruito aziende	
id	int	id	int
Edificio	text	Azienda	text
Localizzazione	text	Sede	text
Riferimento	text	Sito web	text
E-mail	text	E-mail	text
Telefono	text	Telefono	text

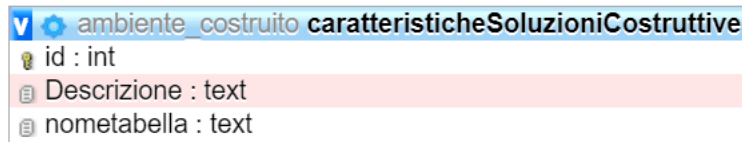
Figura 15: la struttura delle tabelle "best practices" e "aziende"

I campi inseriti in ciascuna tabella dell'atlante sono quelli recuperati dall'output della linea di attività LA2.12, come sarà descritto nella successiva sezione 4.

3.2.2 Le tabelle per la memorizzazione delle soluzioni costruttive e delle soluzioni "costruttive tipo"

Anche nella progettazione della parte di database che contiene i dati relativi alle soluzioni costruttive si è scelto di realizzare una struttura dinamica che possa essere automaticamente incorporata nel portale senza richiedere modifiche al codice JavaScript/PHP. La tabella di riferimento è la tabella denominata

caratteristicheSoluzioniCostruttive che contiene un elenco di tutte le possibili soluzioni costruttive e i nomi delle relative tabelle. In Figura 16 è illustrata la semplice struttura di tale tabella che contiene esclusivamente un campo chiave primaria *id*, la descrizione della soluzione costruttiva da memorizzare nel campo *Descrizione* e, nel campo *nometabella*, il nome della tabella associata che conterrà i dati.



ambiente costruito	caratteristicheSoluzioniCostruttive
id	int
Descrizione	text
nometabella	text

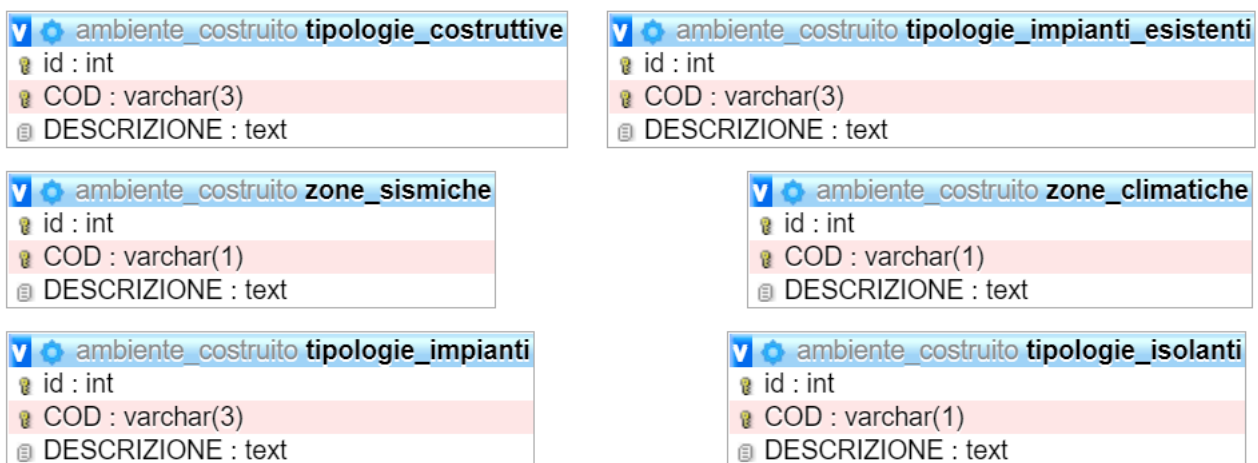
Figura 16: Struttura della tabella *caratteristicheSoluzioniCostruttive*

La tabella attualmente è stata popolata per contenere sette diverse caratteristiche, come illustrato in Figura 17.

id	Descrizione	nometabella
1	Tipologie Costruttive	tipologie_costruttive
2	Tipologie Impianti	tipologie_impianti
3	Zone Climatiche	zone_climatiche
4	Zone Sismiche	zone_sismiche
5	Soluzioni Tipo	soluzionicostruttive_tipo
6	Tipologie Isolanti	tipologie_isolanti
7	Tipologie Impianti Esistenti	tipologie_impianti_esistenti

Figura 17: Contenuto attuale della tabella *caratteristicheSoluzioniCostruttive*

Le strutture delle tabelle *tipologie_costruttive*, *tipologie_impianti*, *zone_climatiche*, *zone_sismiche*, *tipologie_isolanti* e *tipologie_impianti_esistenti*, sono abbastanza semplici e contengono tutte tre campi: una chiave univoca *id*, un codice identificativo *COD* e un campo descrizione *Descrizione* come illustrato nella Figura 18.



ambiente costruito	tipologie_costruttive
id	int
COD	varchar(3)
DESCRIZIONE	text

ambiente costruito	tipologie_impianti_esistenti
id	int
COD	varchar(3)
DESCRIZIONE	text

ambiente costruito	zone_sismiche
id	int
COD	varchar(1)
DESCRIZIONE	text

ambiente costruito	zone_climatiche
id	int
COD	varchar(1)
DESCRIZIONE	text

ambiente costruito	tipologie_impianti
id	int
COD	varchar(3)
DESCRIZIONE	text

ambiente costruito	tipologie_isolanti
id	int
COD	varchar(1)
DESCRIZIONE	text

Figura 18: struttura delle tabelle *tipologie costruttive*, *impianti esistenti*, *impianti*, *isolanti* e *zone sismiche e climatiche*

Per ognuna di queste tabelle sarà presente nel database la relativa tabella con il suffisso “_schema” in maniera tale da permettere la creazione automatica delle pagine di inserimento/modifica come illustrato nella sezione precedente.

Secondo i dati forniti da ENEA, le tabelle sono state popolate come illustrato nella Figura 19.

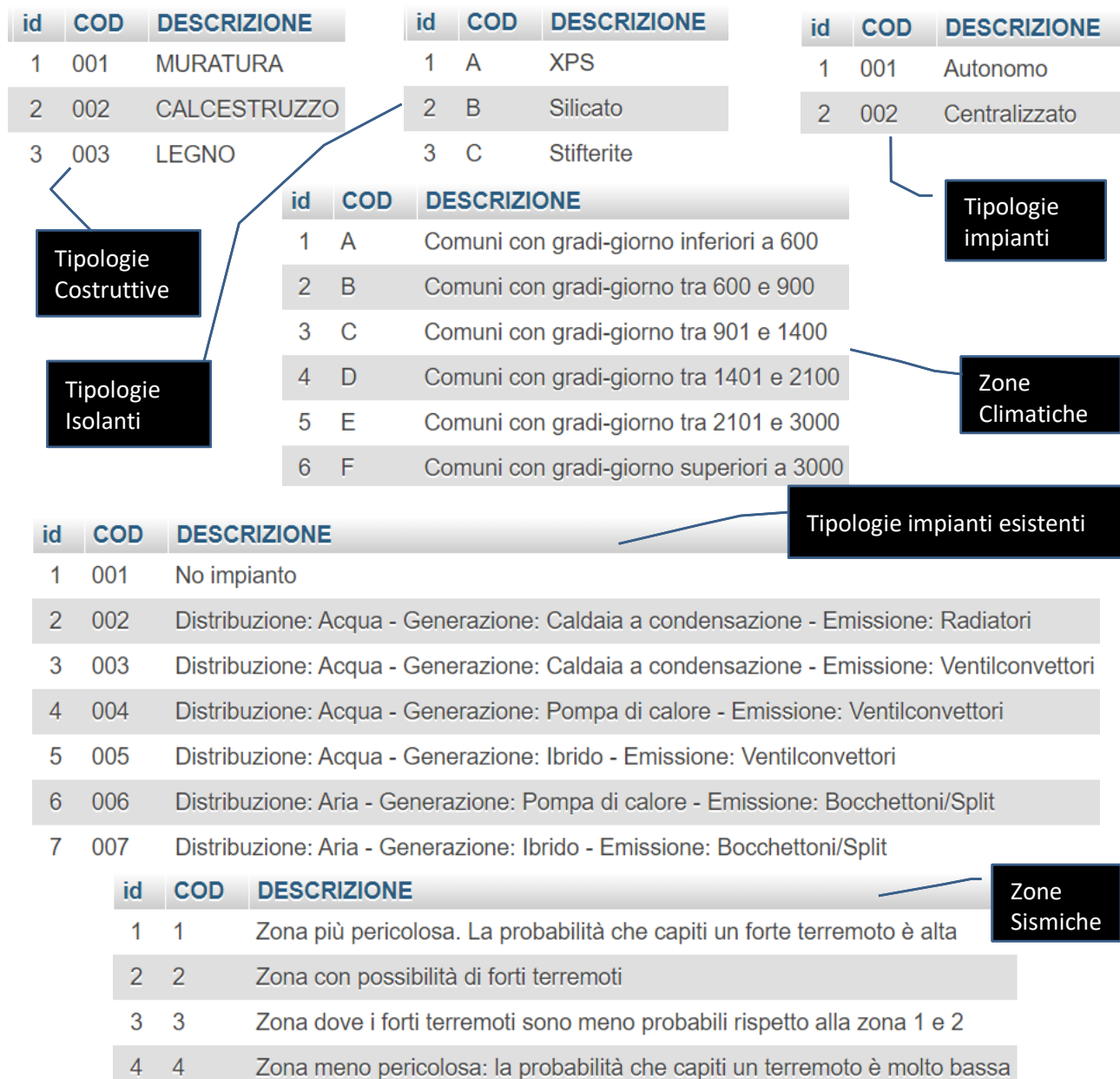


Figura 19: Contenuto delle tabelle tipologie costruttive, impianti, impianti esistenti, isolanti, zone sismiche e climatiche all'atto di rilascio del portale

Una struttura più articolata è invece stata realizzata per la memorizzazione delle “soluzioni costruttive tipo”. La struttura della tabella da associare alle “soluzioni costruttive tipo” infatti, oltre a contenere i campi chiave primaria, codice univoco e descrizione dovrà fare riferimento a tutte le altre proprietà legate alla tipologia costruttiva, alla tipologia di impianto da realizzare, alla tipologia di impianto esistente, alla tipologia di isolanti, e alla zona climatica e sismica di possibile ubicazione. Tutte queste informazioni saranno necessarie per poi poter realizzare la logica della matrice dinamica delle scelte. Per la realizzazione della matrice dinamica è stato quindi necessario implementare delle tabelle ausiliarie che permettessero l’associazione molti a molti fra gli elementi della tabella delle “soluzioni costruttive tipo” e gli elementi delle tabelle

contenenti le informazioni delle possibili tipologie costruttive, tipologie impianti, zone climatiche e zone sismiche come illustrato nella Figura 20, attraverso una tabella di collegamento

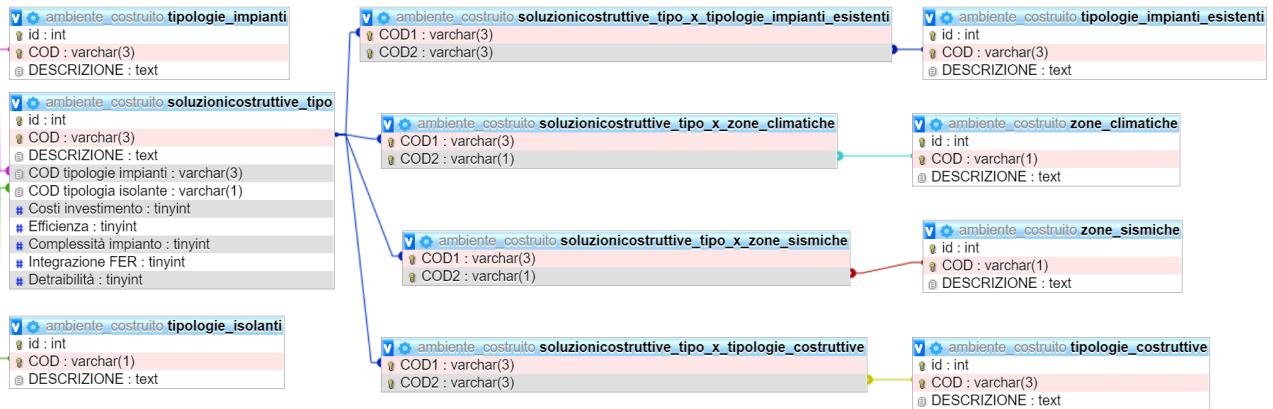


Figura 20: tabella *soluzionicostruttive_tipo* e relative tabelle per implementare la matrice dinamica delle scelte

La tabella *soluzionicostruttive_tipo* conterrà quindi una serie di campi denominati *Costi investimento*, *Efficienza*, *Complessità Impianto*, *IntegrazioneFER*, *Detraibilità* che potranno contenere degli indici interi di qualità in maniera tale da poter comparare le prestazioni delle diverse “soluzioni costruttive tipo” e proporre al visitatore una lista ordinata in funzione di tali prestazioni. Attualmente, la funzione implementata per calcolare l’indice di ordinamento delle “soluzioni costruttive tipo” valuta semplicemente la somma algebrica dei diversi indici memorizzati. Qualora dovesse essere necessario, può essere implementato un algoritmo più sofisticato.

La Figura 21, illustra la struttura della tabella “_schema” utilizzata per definire l’interfaccia grafica di inserimento/modifica. Come si può notare oltre ai campi precedentemente descritti per descrivere le tabelle “_schema” è presente un ulteriore campo, denominato “*xtablename*” che conterrà il nome della tabella di collegamento. La motivazione è legata alla necessità di dover inserire un ulteriore valore per il campo *tipo_campo* che è stato denominato *SELECTCHECK*.

ambiente_costruito_soluzionicostruttive_tipo_schema	
id	int
nome_campo	text
tipo_campo	text
minimo	double
massimo	double
notempty	tinyint(1)
querystring	text
xtablename	varchar(255)

Figura 21: struttura della tabella “_schema” per la tabella “soluzionicostruttive_tipo”

Nel momento in cui dovrà essere visualizzata la pagina di inserimento o modifica delle “soluzioni costruttive tipo”, quando il *tipo_campo* da visualizzare contiene il valore *SELECTCHECK* il codice effettuerà la query contenuta nel campo *querystring*. Il risultato della query permetterà di prelevare tutti i possibili valori che possono essere attribuiti a quel campo. Saranno quindi creati e visualizzati tanti elementi di tipo *checkbox* quante sono le righe restituite dalla query (etichettandole opportunamente). Nel momento in cui l’utente decide di inserire/modificare la “soluzione costruttiva tipo” in oggetto, saranno inseriti nella tabella di collegamento di nome *xtablename* tutte le corrispondenze fra il codice della “soluzione costruttivo tipo” inserita o modificata e i codici delle *checkbox* che risultano essere nello stato *checked*. A titolo di esempio, in riferimento alla definizione delle possibili zone climatiche, sarà presente nella tabella *soluzionicostruttive_tipo_schema* una riga contenente le informazioni illustrate in Figura 22. All’atto della

visualizzazione sarà quindi effettuata la query, indicata nel campo *querystring*, che preleva tutti i codici e le descrizioni (formando un'unica stringa costituita dalla concatenazione delle stringhe di codice e di descrizione vera e propria) dalla tabella *zone_climatiche*.

nome_campo	tipo_campo	querystring	tablename
COD zona climatica	SELECTCHECK	SELECT COD, CONCAT(COD, ' - ', DESCRIZIONE) AS DESCRIZIONE FROM 'zone_climatiche';	soluzionicostruttive_tipo_x_zone_climatiche

Figura 22: contenuto della tabella *soluzionicostruttive_tipo_schema* in relazione al campo COD zona climatica

Quindi nella pagina saranno inseriti il corrispondente numero di *checkbox* etichettati con il codice e aventi per tooltip la descrizione completa. Nel momento in cui l'utente inserisce/modifica il nuovo elemento delle "soluzioni costruzioni tipo", sarà effettuata una query di cancellazione nella tabella di collegamento *soluzionicostruttive_tipo_x_zone_climatiche* di tutti gli elementi aventi per codice *COD1* il codice della "soluzione costruttiva tipo in oggetto" e, successivamente, saranno inserite tutte le coppie (*COD1*, *COD2*) contenenti il "codice della soluzione costruttiva tipo" in oggetto e il codice delle zone climatiche che risultano nello stato *checked*.

3.2.3 Le tabelle di login e di log

La tabella di *login* contiene, oltre al campo chiave primaria *id*, un campo *name* per contenere la stringa descrittiva del nome utente, un campo *user* per definire la stringa univoca d'accesso dell'utente, il campo *password* in cui sarà memorizzata la password di accesso codificata in MD5 e un campo booleano *isAdmin* che specifica se l'utente ha i privilegi di amministratore.

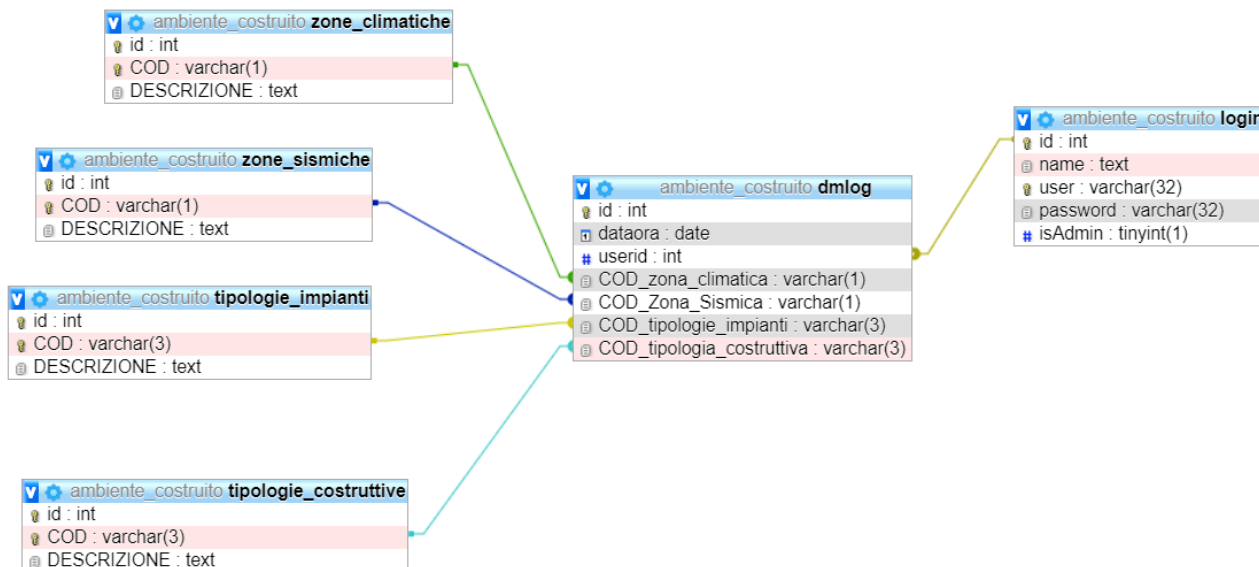


Figura 23: tabelle di login e di log con le relative relazioni

La tabella *dmlog* è strutturata per contenere i log di tutte le ricerche effettuate utilizzando la tabella dinamica delle scelte. Saranno memorizzati, oltre alla chiave primaria *id*, le informazioni relative alla data e ora in cui è stata effettuata la richiesta di elaborazione e i dati in ingresso utilizzati specificati dai campi *COD_zona_climatica*, *COD_Zona_Sismica*, *COD_tipologie_impianti* e *COD_tipologia_costruttiva*. Tali informazioni potranno essere sfruttate per eventuali future analisi statistiche sulla tipologia di ingressi utilizzati.

3.2.4 La tabella dm_schema

La tabella *dm_schema* è l'unica tabella di tipo “_schema” non associata a una tabella di base contenente informazioni da visualizzare. Il suo scopo è esclusivamente di supporto alla visualizzazione della pagina che permetterà all'utente di inserire i valori di selezione per utilizzare la matrice dinamica delle scelte. La sua struttura, riportata in Figura 24, ha, nell'implementazione attuale, il contenuto illustrato in Figura 25. Come si può osservare sono contenute 4 diverse righe relative a 4 rispettivi campi tutti di tipo SELECT. Questo permetterà alle routine JavaScript di visualizzare le corrispondenti drop-down list che permetteranno all'utente di impostare i valori della sua condizione operativa.

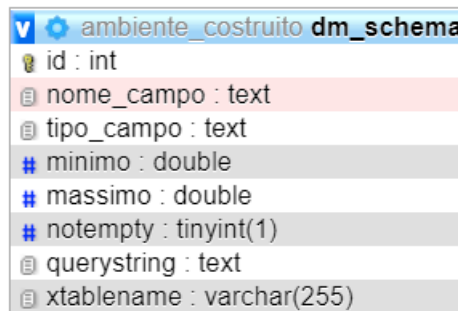


Figura 24: struttura della tabella dm_schema

id	nome_campo	tipo_campo	minimo	massimo	notempty	querystring	xtablename
1	COD tipologia costruttiva	SELECT	NULL	NULL	NULL	SELECT COD, CONCAT(COD, '-', DESCRIZIONE) AS DESCRIZIONE FROM 'tipologie_costruttive';	'soluzionicostruttive_tipo_x_tipologie_costruttive': 'COD2'
2	COD tipologie impianti	SELECT	NULL	NULL	NULL	SELECT COD, CONCAT(COD, '-', DESCRIZIONE) AS DESCRIZIONE FROM 'tipologie_impianti_esistenti';	'soluzionicostruttive_tipo_x_tipologie_impianti_esistenti': 'COD2'
3	COD zona climatica	SELECT	NULL	NULL	NULL	SELECT COD, CONCAT(COD, '-', DESCRIZIONE) AS DESCRIZIONE FROM 'zone_climatiche';	'soluzionicostruttive_tipo_x_zone_climatiche': 'COD2'
4	COD zona sismica	SELECT	NULL	NULL	NULL	SELECT COD, CONCAT(COD, '-', DESCRIZIONE) AS DESCRIZIONE FROM 'zone_sismiche';	'soluzionicostruttive_tipo_x_zone_sismiche': 'COD2'

Figura 25: contenuto della tabella dm_schema nell'attuale rilascio del portale

In questo caso il campo *xtablename* non conterrà semplicemente il nome della tabella di collegamento ma il nome della tabella di collegamento associato al campo su cui effettuare la selezione attraverso la clausola “WHERE” di SQL.

3.3 Il codice PHP

3.3.1 Il file mysql.php

Il file *mysql.php* è un modulo che fornisce le routine di accesso al database MySQL che è attivo sul server Apache. Il modulo si appoggia all'estensione MySQLi, una nuova API per Apache che, a partire dalla versione 5 di PHP, consente un'interrogazione più versatile del server MySQL.

3.3.2 Il file myquery.php

Il file *myquery.php* contiene le informazioni per accedere al database MySQL (nome utente, password, nome del database) e un array di query SQL opportunamente indicizzate.

3.3.3 Il file mygetdata.php

Il file *mygetdata.php*, utilizzando il metodo POST, provvede ad eseguire una query di selezione sul database restituendo in uscita, in formato di array di oggetti codificati in JSON, diverse informazioni: numero delle righe restituite dalla SELECT, nomi dei campi, valori delle diverse righe restituite dalla query. I parametri di ingresso necessari che devono essere specificati tramite il metodo POST sono:

- La chiave di accesso all'array di query memorizzato nel file *myquery.php*;
- Un'eventuale clausola da aggiungere alla query (la quale, ad esempio, può specificare una condizione di selezione tramite il costrutto WHERE, può limitare il numero di righe restituite tramite il costrutto

LIMIT o ordinare le righe restituite tramite il costrutto ORDER BY); se non devono essere specificate clausole basta passare la stringa vuota;

- Il numero di colonne da restituire;
- La posizione di un'eventuale chiave univoca all'interno della tabella in maniera tale da non considerare tale campo nel conteggio delle colonne da restituire e da restituire il valore assunto da tale campo per ogni riga in maniera opportuna.

Nello specifico la riga relativa ai nomi dei campi sarà formattata nella forma:

..., {{Campo_XX}}="NOME CAMPO", ...

dove XX rappresenta la colonna all'interno della tabella a partire dalla posizione 01 (il numero massimo di colonne restituibili è quindi pari a 99).

Analogamente le righe relative all'esecuzione della query di selezione sul database saranno restituite nella forma

```
...
...
{{Campo_00}}="VALORE CHIAVE", ...,{{Campo_XX}}="VALORE CAMPO", ...
...
...
```

con lo stesso significato per il suffisso XX relativo alla riga dei nomi dei campi. Come si evince, la posizione 00 è riservata per inserire il valore della chiave univoca di ogni riga il cui "nome campo" è stato specificato attraverso uno dei parametri di ingresso.

Specificando il numero C di colonne da restituire la routine formatterà nei campi delle righe della tabella di uscita esclusivamente le prime C-1 colonne (escludendo, come detto, l'eventuale chiave primaria). All'interno delle C-esima colonna sarà invece inserito il contenuto di tutte le rimanenti colonne della selezione effettuata nella forma

```
...
...
<NOME CAMPO_YY> = <VALORE_YY>
<NOME CAMPO_YY> = <VALORE_YY>
...
...
```

dove YY sono tutte le colonne restituite dalla query di selezione a partire dalla C-esima.

3.3.4 Il file mygetdatafromquery.php

Il file *mygetdatafromquery.php* contiene il codice per eseguire una specifica query SQL di tipo SELECT passata come parametro di ingresso attraverso il metodo POST. Differentemente dalla routine precedente la query è passata come stringa di testo ed è previsto esclusivamente un ulteriore parametro di ingresso che specifica il numero di colonne da restituire. In uscita sarà restituito esclusivamente l'insieme delle righe ottenute dall'esecuzione della query di selezione con un formato analogo a quello visto per il file *mygetdata.php* (esclusa la presenza del campo che specifica il valore della chiave primaria).

3.3.5 Il file mygetondata.php

Il file *mygetondata.php* accetta come parametri di ingresso il nome di una tabella del database e l'eventuale clausola da applicare. Dopo aver applicato al database una query di selezione che preleva tutti i campi della tabella, restituisce in uscita un oggetto JSON contenente righe codificate nella forma

```
...  
...  
... <NOME CAMPO>=<VALORE CAMPO>, <NOME CAMPO>=<VALORE CAMPO>, ...  
...  
...
```

3.3.6 Il file *mygettableschema.php*

Il file *mygettableschema.php* è utilizzato per eseguire sul database delle query di selezione che prelevano tutti i campi e i relativi valori delle tabelle che hanno un nome che finisce con il suffisso “_schema”. Tali tabelle ausiliarie, come visto nella sezione 3.2, sono utilizzate per implementare automaticamente l'interfaccia grafica che permette l'inserimento e la modifica di nuovi elementi delle tabelle contenenti i dati veri e propri. Dopo aver eseguito la query di selezione che permette di prelevare tutti i campi della tabella, sarà restituito un oggetto JSON in cui ogni riga contiene i valori

```
...  
...  
... <NOME CAMPO>=<VALORE CAMPO>, <NOME CAMPO>=<VALORE CAMPO>, ...  
...  
...
```

3.3.7 Il file *myinsertdata.php*

Il file *myinsertdata.php* è utilizzato per effettuare sia le operazioni di inserimento sia le operazioni di modifica del contenuto delle righe di tutte le tabelle dati. Sempre utilizzando il metodo POST devono essere forniti in ingresso quattro parametri. Il primo di questi è un valore stringa che identifica la tabella oggetto delle operazioni di inserimento o modifica. Il secondo è un valore stringa contenente il nome di tutti i campi i cui valori devono essere inseriti come nuova riga o modificati. Il terzo campo conterrà l'elenco dei valori da associare in ciascuna riga ai campi inseriti come secondo parametro. L'ultimo parametro permette di specificare se si tratta di un'operazione di inserimento o di modifica.

L'operazione di inserimento sarà effettuata utilizzando il costrutto SQL mostrato in Figura 26

```
INSERT INTO <nometabella> ( <elencocampi> ) VALUES ( <elencovalori> )
```

Figura 26: Costrutto SQL per l'inserimento nelle tabelle

mentre le operazioni di modifica saranno effettuate attraverso il costrutto SQL illustrato in Figura 27

```
REPLACE INTO <nometabella> ( <elencocampi> ) VALUES ( <elencovalori> )
```

Figura 27: Costrutto SQL per la modifica dei valori degli elementi già inseriti in tabella

Quest'ultima sintassi per la modifica dei contenuti delle righe già presenti nelle tabelle è possibile grazie alla presenza in ognuna di esse della chiave primaria *id*. Il parametro *elencocampi* e il parametro *elencovalori* dovranno specificare necessariamente in questo caso, rispettivamente, il nome del campo della chiave primaria (che nel nostro caso è sempre *id*) e il valore ad esso associato della riga che si vuole modificare. Il file restituirà in uscita un oggetto codificato in formato JSON contenente un unico valore booleano impostato a VERO se l'operazione di inserimento/modifica è andata a buon fine.

3.3.8 Il file *mydeletedata.php*

Questo file php è richiamato per cancellare una specifica riga di una tabella. Deve essere richiamato passando due parametri in ingresso utilizzando il metodo POST. Il primo parametro deve contenere una stringa con il nome della tabella dalla quale si vuole cancellare la riga. Il secondo parametro deve contenere il valore assunto dalla chiave univoca relativo alla riga che si vuole cancellare. Analogamente al caso precedente sarà restituito un oggetto booleano codificato in formato JSON equivalente a VERO solo se l'operazione è stata correttamente eseguita.

3.3.9 Il file *myupdateX.php*

Il file *myupdateX.php* è utilizzato per aggiornare i valori delle tabelle che contengono le informazioni necessarie ad implementare la matrice dinamica associando le "soluzioni costruttive tipo" con diverse possibili combinazioni di tipologie costruttive, tipologie di impianti, zone climatiche e zone sismiche (vedi sezione 3.2.2). È necessario specificare tre diversi parametri in ingresso utilizzando il metodo POST: il nome della tabella di collegamento in cui si vuole agire (nel nostro caso sarà una delle tabelle *soluzionicostruttive_tipo_x_tipologie_costruttive*, *soluzionicostruttive_tipo_x_tipologie_impianti_esistenti*, *soluzionicostruttive_tipo_x_zone_climatiche* o *soluzionicostruttive_tipo_x_zone_sismiche*) l'insieme delle coppie nella forma (<COD1>, <COD2>) che specificano rispettivamente il codice identificativo dell'elemento della prima tabella (nel nostro caso sarà sempre il valore di un codice identificativo univoco di una riga della tabella *soluzionicostruttive_tipo*) e il codice identificativo univoco di una riga della seconda tabella; il terzo parametro contiene invece semplicemente il valore dell'identificativo <COD1> della prima tabella. La funzione PHP si occuperà di eseguire in cascata due query SQL. Nella prima cancellerà dalla tabella di collegamento tutti gli elementi che hanno il valore nel campo <COD1> corrispondente a quello passato come terzo parametro. Nella seconda saranno inseriti nella tabella di collegamento tutte le coppie di valori passati come secondo parametro. La routine restituirà un oggetto JSON rappresentante un valore booleano corrispondente a VERO se entrambe le query saranno eseguite correttamente.

3.3.10 Il file *myIsUser.php*

Attraverso il file *myIsUser.php* sarà possibile verificare se le credenziali inserite dall'utente corrispondono a uno degli utenti memorizzati nella tabella di *login*. Attraverso il metodo POST è necessario passare al file i campi contenenti al nome utente e la password (codificata in MD5) dell'utente. Se la combinazione viene correttamente trovata all'interno della tabella *login* il file PHP restituirà un oggetto codificato in JSON contenente il nome completo dell'utente e il valore del campo *IsAdmin* che specifica se l'utente è un amministratore. Quest'ultimo campo potrebbe essere utilizzato anche per specificare diversi livelli di accesso al sistema permettendo di creare diverse categorie di utenti abilitati a diverse tipologie di operazioni.

Si noti che, come richiesto, molte delle funzionalità del portale possono funzionare anche senza un'autenticazione esplicita da parte dell'utente (modalità *anonymous user*). Solo funzionalità specifiche, come l'inserimento, la modifica o la cancellazione di elementi del database, richiedono che l'utente si autentichi espressamente. Attualmente è previsto esclusivamente il privilegio di amministratore e l'inserimento degli utenti nella tabella *login* non è permesso attraverso l'interfaccia WEB del portale (è il *sysadmin* del portale web che deve registrare gli utenti ed eventualmente concedere questo privilegio agendo direttamente sulla tabella *login*).

3.4 Il codice HTML/JavaScript

3.4.1 Il file *index.html*

Il file *index.html* contiene il codice HTML caricato dal browser al momento della visualizzazione del portale. Il codice contenuto in tale file ha il ruolo di "radice" del progetto, nel senso che tutti le altre funzioni scritte in JavaScript o PHP sono caricate ed eseguite dal browser o dal server Apache perché sono richiamate (direttamente o indirettamente) dall'interno di questo file. Analizzeremo di tale file le parti più salienti. La parte iniziale del file (Figura 28) stabilisce lo stile che permette di visualizzare i "tooltip". I "tooltip" sono delle particolari caselle di testo che vengono visualizzate al di sopra degli elementi che costituiscono la pagina web quando il puntatore si sofferma su tali valori. A seguire viene definita l'intestazione della pagina e vengono

caricate le librerie e i font (Figura 29). Successivamente all'interno dell'elemento "`<body>`", a seguito dell'evento "`onload`" del file, sarà richiamata più volte la routine "`mycreateDIV`" che effettua la creazione dinamica delle pagine da visualizzare all'interno del corpo centrale del portale. Tale routine sarà richiamata per le pagine che conterranno rispettivamente:

- gli elementi strutturali contenuti nell'atlante (`<div> mydivcontainer1`);
- gli elementi "best practices" contenuti nell'atlante (`<div> mydivcontainer3`);
- la rubrica delle aziende contenute nell'atlante (`<div> mydivcontainer4`);
- le soluzioni costruttive e le "soluzioni costruttive tipo" (`<div> mydivcontainer2`);

Inoltre, sarà richiamata la routine "`mycreateODMForm`" per la creazione dinamica della pagina che permetterà all'utente di accedere alla matrice dinamica delle scelte.

Utilizzando questo approccio le pagine a contenuto dinamico saranno create quando il file HTML sarà caricato dal browser.

```
<!DOCTYPE html>
<html>
<style>
.tooltip {
  position: relative;
  display: inline-block;
  border-bottom: 1px dotted black;
}

.tooltip .tooltiptext {
  visibility: hidden;
  width: 120px;
  ...
  ..
}
</style>
```

Figura 28: File index.html (sezione iniziale)

```
<head>
<title>Ambiente costruito</title>
<meta charset="UTF-8">
<meta content="Author" name="Salvatore SERRANO 1972.04.06 Dipartimento di Ingegneria - Università di Messina">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="icon" href="images/Favicon.ico" />
<link rel="stylesheet" href="css/loader.css">
<link rel="stylesheet" href="css/w3.css">
<link rel="stylesheet" href="font/raleway/stylesheets.css">
<link rel="stylesheet" href="font/font-awesome/h5p-font-awesome.min.css">
<style>
  body,h1,h2,h3,h4,h5,h6
  {
    font-family: "Raleway", Arial, Helvetica, sans-serif;
  }
</style>
<script src="js/w3.js" type="text/javascript"></script>
<script src="js/jquery-3.4.1.min.js" type="text/javascript"></script>
</head>
```

Figura 29: File index.html (intestazione)

```
<body onload="mycreateDIV('mydivcontainer1',3);
mycreateDIV('mydivcontainer2',2);
mycreateDIV('mydivcontainer3',2);
mycreateDIV('mydivcontainer4',2);
mycreateODMForm('dm', 0);
w3.hide('.unime-details, .w3-logout-button, [class*='unime-page-'], ... );
w3.show([class*='unime-page-default-'], [class*='unime-subpage-default-'], ... );"
>
```

Figura 30: File index.html (routine richiamate su evento "onload" della pagina)

Il primo elemento inserito all'interno del "`<body>`" sarà la colonna di navigazione attraverso un tag di tipo "`<nav>`". La colonna di navigazione conterrà a sua volta tutti gli elementi che permettono di accedere alle varie sezioni del portale e, in particolare, alle sezioni:

- *Login/Logout;*
- *Home page;*
- *Atlante:*
 - *Strutture (Tipologia, Categoria)*
 - *Best Practices*
 - *Aziende*
- *Soluzioni costruttive;*
- *Matrice dinamica;*
- *Contatti.*

Di queste, le sezioni *Login/Logout*, *Home page*, *Matrice dinamica* e *Contatti* sono delle semplici sezioni che si attivano direttamente e unicamente selezionando il relativo elemento di navigazione. La sezione *Atlante* invece comporterà l'apertura di una ulteriore sottosezione costituita dagli elementi *Strutture*, *Best Practices*, *Aziende*; l'elemento *Strutture* permetterà la selezione sia della *Tipologia* sia dell'eventuale *Categoria* delle stesse. Alla stessa maniera l'attivazione dell'elemento *Soluzioni Costruttive* permetterà la selezione da parte dell'utente di diverse categorie associate a questo elemento.

A titolo esemplificativo nella Figura 31 è illustrata la parte iniziale del codice HTML che implementa la colonna di navigazione verticale. Il primo elemento `<div>` contiene l'immagine con il logo del progetto. Il secondo elemento `<div>` conterrà gli elementi che sono utilizzati per effettuare le operazioni di login/logout da parte dell'amministratore. La struttura utilizzata per l'implementazione dei pulsanti di selezione all'interno della colonna di selezione è sempre la stessa. Essa è costituita da un tag `<a>` di definizione di un hyperlink a cui è associata una funzione JavaScript che viene richiamata a seguito dell'evento *onclick*. La disposizione degli elementi è realizzata sfruttando una tabella con un'unica riga a tre celle: la prima cella contiene un'eventuale icona descrittiva della funzionalità, la seconda cella contiene il testo associato all'elemento funzionale, la terza cella contiene un'eventuale altra icona che evidenzia l'apertura o chiusura della sottosezione quando è presente. Solo nel caso della visualizzazione del pulsante di logout all'interno della tabella saranno presenti due righe in quanto, nella prima, sarà visualizzato il nominativo dell'utente attualmente loggato nel portale.

```

<nav class="w3-sidebar w3-light-grey w3-collapse w3-top" style="z-index:3;width:260px" id="unimeSidebar">
  <div class="w3-container w3-display-container w3-padding-16" <!-- DIU TITOLO-->
    <i onclick="w3.hide('#unimeSidebar, #unimeOverlay');"
      class="fa fa-remove w3-hide-large w3-button w3-transparent w3-display-topright">
    </i>
    
  </div>
  <div class="w3-bar-block" <!-- DIU LEVEL 1 -->
    <a href="javascript:void(0)" onclick="logoutClick()"
      class="w3-bar-item w3-button w3-logout-button">
      <table style="width:100%;">
        <tr>
          <td>
            <td>
            </td>
          <td>
            <label id="loginname">_</label>
          </td>
        </tr>
        <tr>
          <td>
            <td style="text-align:left;font-size:1.5em;">
              <i class="fa fa-arrow-circle-o-left"></i>
            </td>
          <td>
            Logout
          </td>
        </tr>
      </table>
    </a>
    <a href="javascript:void(0)" onclick="loginClick()"
      class="w3-bar-item w3-button w3-login-button">
      <table style="width:100%;">
        <tr>
          <td style="text-align:left;font-size:1.5em;">
            <i class="fa fa-user"></i>
          </td>
          <td>
            Login
          </td>
        </tr>
      </table>
    </a>
  </div>

```

Figura 31: file "index.html" (parte del codice utilizzato per implementare la colonna di navigazione)

```

<div>
  <a onclick="abachiClick();" href="javascript:void(0)" class="w3-bar-item w3-button">
    <table style="width:100%;"><tr>
      <td style="text-align:left;font-size:1.5em;"><i class="fa fa-table"></i></td>
      <td>Atlante</td>
      <td style="text-align:right;">
        <i class="fa fa-chevron-right unime-element-default-abacoitem"></i>
        <i class="fa fa-chevron-down unime-element-abacoitem">
      </td>
    </tr></table>
  </a>
  <div class="unime-submenu-abacoitem" style="margin-left: 15px;">
    <a onclick="struttureClick();" href="javascript:void(0)" class="w3-bar-item w3-button">
      <table style="width:100%;"><tr>
        <td style="text-align:left;font-size:1.5em;"><i class="fa fa-cubes"></i></td>
        <td>Strutture</td>
        <td style="text-align:right;">
          <i class="fa fa-chevron-right unime-element-default-abacoitem"></i>
          <i class="fa fa-chevron-down unime-element-abacoitem">
        </td>
      </tr></table>
    </a>
    <div class="w3-container w3-display-container w3-padding-16 unime-submenu-abacoitem">
      <div class="w3-container w3-display-container w3-padding-16 unime-submenu-abacoitem"
        id="unime_elementifunzionali_field">
        <p><label><b>{{Campo_02}}</b></label></p>
      </div>
      <select class="w3-input w3-border w3-round" name="elementifunzionali"
        id="unime_elementifunzionali_data" onchange="elementifunzionaliChange();">
        <option w3-repeat="elementifunzionali_data" value="{{Campo_03}}>{{Campo_02}}</option>
      </select>
      <div class="w3-container w3-display-container w3-padding-16 unime-submenu-abacoitem"
        id="unime_elementifunzionali_categorie_field">
        <p><label><b>{{Campo_02}}</b></label></p>
      </div>
      <select class="w3-input w3-border w3-round" name="categorie"
        id="unime_elementifunzionali_categorie_data" onchange="unime_elementifunzionali_categorieChange();">
        <option w3-repeat="elementifunzionali_categorie_data" value="{{Campo_01}}>{{Campo_02}}</option>
      </select>
    </div>
    <a onclick="bestpracticesClick();" href="javascript:void(0)" class="w3-bar-item w3-button">
      <table style="width:100%;"><tr>
        <td style="text-align:left;font-size:1.5em;"><i class="fa fa-thumbs-o-up"></i></td>
        <td>Best Practices</td>&nbsp;
        <td style="text-align:right;">
      </tr></table>
    </a>
    <a onclick="aziendeClick();" href="javascript:void(0)" class="w3-bar-item w3-button">
      <table style="width:100%;"><tr>
        <td style="text-align:left;font-size:1.5em;"><i class="fa fa-cogs"></i></td>
        <td>Aziende</td>&nbsp;
        <td style="text-align:right;">
      </tr></table>
    </a>
  </div>
</div>

```

Figura 32: file index.html (codice HTML che implementa la selezione degli elementi dell'Atlante)

In Figura 32 è invece illustrata la sezione di codice HTML più complessa che implementa la selezione degli elementi dell'Atlante. Il primo tag <a> (Atlante) permette che possano essere visualizzati o nascosti gli elementi contenuti nella sezione ed in particolare il submenu Strutture, il quale a sua volta permetterà l'apertura e la chiusura di una sezione contenente due elementi di tipo "drop-down list" implementati attraverso i relativi tag <select>. L'analisi del codice presente all'interno di questi tag di tipo <select> permette di individuare un primo elemento dinamico con cui lo stesso è stato realizzato. Come si può infatti notare sia i nomi delle etichette (tag <label>) sia le diverse possibili opzioni delle liste non sono definite in maniera statica. Il testo da visualizzare nelle etichette e, i valori e il testo da visualizzare nelle drop-down list, saranno successivamente inseriti durante il funzionamento stesso del portale attraverso l'interazione con il database tramite il codice PHP.

Successivamente alla chiusura del tag <nav> contenente i link di navigazione della colonna verticale, il file *index.html* contiene il codice che implementa il corpo centrale del portale. In particolare, come illustrato nella Figura 33 sarà presente il codice che realizza l'intestazione fissa e l'effetto di apertura e chiusura della colonna verticale sugli schermi di piccole dimensioni.


```

</nav>
<header class="w3-bar w3-top w3-black w3-xlarge" style="height:50px">
  
  <span class="w3-bar-item w3-hide-small w3-hide-medium" style="padding:0px;margin-left:196px">&nbsp;</span>
  <span class="w3-bar-item">Ambiente costruito</span>
  <a href="javascript:void(0)" class="w3-right w3-bar-item w3-button w3-hide-large"
    onclick="w3.show('#unimeSidebar, #unimeOverlay');"><i class="fa fa-bars"></i></a>
</header>

<!-- Overlay effect when opening sidebar on small screens -->
<div class="w3-overlay w3-hide-large" onclick="w3.hide('#unimeSidebar, #unimeOverlay');"
  style="cursor:pointer" title="Chiudi il side menu" id="unimeOverlay"></div>

<!-- !PAGE CONTENT! -->
<div class="w3-main w3-white" style="margin-left:260px">

```

Figura 33: file index.html (sezione del codice per l'implementazione del corpo centrale del portale)

A seguire il codice sarà costituito esclusivamente da sezioni di segnaposto che saranno opportunamente e dinamicamente riempite durante l'utilizzazione del portale da parte dell'utente. La Figura 34 visualizza alcuni esempi di tali segnaposto relativi alle pagine di inserimento, di modifica e di visualizzazione della matrice dinamica. Il corpo centrale del portale è sempre chiuso da un piè di pagina fluttuante e allineato a destra (implementato attraverso il tag `<footer>`) contenente il logo, il link e il nominativo del Dipartimento di Ingegneria dell'Università degli Studi di Messina.

```

<div class="unime-page-inserimenti", id="mydivcontainernuovo">
  <h2 class="w3-text-green">INSERIMENTO</h2>
</div>

<div class="unime-page-modifiche", id="mydivcontainermodifica">
  <h2 class="w3-text-green">MODIFICA</h2>
</div>

<div class="unime-page-dm", id="mydivcontainerdm">
  <h2 class="w3-text-green">MATRICE DINAMICA</h2>
</div>

<footer class="w3-container w3-padding-16" style="margin-top:32px; float:right;">
  Powered by
  <a href="https://www.unime.it" title="W3.CSS" target="_blank"
    class="w3-hover-text-green">Università&grave; degli Studi di Messina - Ingegneria
    
  </a>
</footer>
</div>
</body>
<script src="js/do.js" type="text/javascript"></script>
</html>

```

Figura 34: file index.html (sezione finale)

3.4.2 Le funzioni del file *do.js*

Il file *do.js* contiene tutte le funzioni scritte in *JavaScript* che permettono il funzionamento del portale. Di seguito saranno analizzate le principali funzioni implementate descrivendone i parametri di ingresso e le uscite prodotte.

- La funzione *mycreateDIV(container, columns)* riceve in ingresso l'identificativo di un container e il numero di colonne da visualizzare all'interno della tabella che sarà creata. La routine si occupa di creare tutti gli elementi HTML che saranno utilizzati all'interno delle pagine di visualizzazione delle tabelle e appenderle al container il cui id è stato passato come parametro di ingresso. Nello specifico sono creati:
 - gli elementi per la visualizzazione del sottotitolo e dell'eventuale pulsante di inserimento di un nuovo elemento in tabella (pulsante che sarà visibile solo in modalità "amministratore");
 - gli elementi che consentono di filtrare in maniera dinamica le righe visualizzate nella tabella e in particolare: a) una "drop-down list" che conterrà tutti i campi della tabella visualizzata;

b) una seconda “drop-down list” che permetterà di selezionare l’operatore di confronto da utilizzare e tra i valori:

- “CONTIENE” equivalente all’operatore di confronto “LIKE”,
- “NON CONTIENE” equivalente all’operatore di confronto “NOT REGEXP”,
- “UGUALE A” equivalente all’operatore di confronto “=”,
- “MINORE DI” equivalente all’operatore di confronto “<”,
- “MAGGIORE DI” equivalente all’operatore di confronto “>”;

c) un elemento “input box” in cui l’utente potrà inserire il valore di confronto; d) un elemento per poter aggiungere la condizione di filtraggio; e) un elemento per poter cancellare tutte le condizioni di filtraggio precedentemente aggiunte;

- gli elementi che consentono di traslare avanti o indietro l’offset iniziale di visualizzazione delle righe della tabella e gli elementi che consentono di visualizzare l’attuale posizionamento dell’offset nonché il numero totale di righe presenti nella tabella;
- gli elementi che consentono di visualizzare la tabella con i dati. In ogni riga saranno presenti i pulsanti per richiamare la visualizzazione di un’eventuale scheda associata memorizzata sul server in formato pdf ed eventualmente (nel caso in cui l’operatore sia un amministratore) i pulsanti di modifica e cancellazione delle singole righe.

È importante sottolineare che la routine crea esclusivamente gli elementi di visualizzazione senza riempirne il contenuto. Tutti gli elementi saranno infatti riempiti dinamicamente nel momento in cui l’utente selezionerà la tabella da visualizzare attraverso i pulsanti presenti nella colonna laterale di navigazione.

- Le funzioni *mycreateODMForm(tablename, attempt)*, *mycreateDMForm(tablename)*, *mycreate1DMForm()*, *myshowTableDM()* sono utilizzate per creare e visualizzare la pagina contenente le funzionalità che permettono all’utente di visualizzare le soluzioni tipo che si adattano a specifiche condizioni costruttive e che sfrutta la matrice dinamica delle scelte. I parametri di ingresso della funzione *mycreateODMForm(tablename, attempt)*, sono il nome della tabella contenente le informazioni per effettuare la scelta sulla matrice dinamica e un valore intero che identifica il tentativo di creazione in atto. Il meccanismo di creazione è infatti implementato tramite l’oggetto *JavaScript* denominato *Promise*. Per la corretta creazione della pagina è infatti necessario interrogare il database attraverso gli script PHP, ricevere i risultati e riempire correttamente i contenuti di tutti gli elementi. Quanto il valore della variabile “*attempt*” è pari a 0 (il che significa che siamo al primo tentativo), viene richiamata la routine *mycreateDMForm(tablename)* che eseguirà in modalità asincrona il proprio task. Si imposta quindi un timeout pari al tempo specificato nella variabile *promiseTimeOut* (impostato per default a 250ms) e si attende il suo scadere. Nel frattempo, la routine *mycreateDMForm(tablename)*, attraverso il codice PHP, avrà interrogato il database per ricevere informazioni relative ai possibili input dell’utente memorizzate all’interno della tabella *tablename* e avrà iniziato la creazione degli elementi corrispondenti. L’ultima operazione che sarà effettuata dalla routine sarà quella di impostare il valore di un semaforo che indica che tutti gli elementi della pagina sono pronti. Allo scadere del *timeout*, la funzione *mycreateODMForm*, valuterà il valore del semaforo: se questo non è ancora impostato chiamerà nuovamente se stessa incrementando il valore della variabile *attempt* di una unità; se invece il semaforo è stato impostato vuol dire che la routine di creazione ha completato il suo lavoro e si potrà proseguire alla visualizzazione della pagina richiamando la funzione *mycreate1DMForm()*. Il numero di tentativi massimi è pari a *promiseAttempts* (valore impostato a 10 per default), dopo i quali viene scritto sulla console del browser un messaggio informativo di errore e si conclude l’attesa (l’utente non vedrà la pagina visualizzata correttamente). La funzione *mycreate1DMForm()* interrogherà nuovamente il database per riempire i campi presenti nella pagina e si occuperà di richiamare la funzione *myshowTableDM()* che ha l’unico compito di rendere visibile all’utente la pagina così creata.

È importante sottolineare che la struttura della tabella del database da passare a queste funzioni deve essere di quelle nella forma “<nometabella>_schema”. In queste tabelle, infatti, sono

memorizzate informazioni relative alla struttura degli elementi da visualizzare nella pagina per effettuare eventuali operazioni di inserimento/modifica degli elementi della tabella originale. La stessa struttura può comunque essere utilizzata per creare dinamicamente delle pagine la cui tipologia di elementi è specificata all'interno della tabella del database.

Nel caso della matrice dinamica, come evidenziato nella sezione 3.2.4, la tabella "*dm_schema*" contiene, nell'attuale implementazione, 4 righe tutte caratterizzate dal valore "*tipo_campo*" impostato a *SELECT*. Nella creazione della pagina contenente la matrice dinamica saranno di conseguenza visualizzati 4 elementi di tipo drop-down list a cui è associata una etichetta data dal valore "*nome_campo*" di una specifica riga della tabella e le cui opzioni di selezione sono riempite interrogando il database eseguendo la query memorizzata nel valore "*querystring*" della corrispondente specifica riga della tabella.

L'eventuale inserimento, eliminazione e modifica dei campi di selezione che implementano la logica della matrice dinamica delle scelte potrà quindi essere effettuata semplicemente inserendo, modificando o eliminando righe dalla tabella "*dm_schema*" senza modificare il codice JavaScript e PHP.

- Le funzioni *loginClick()* e *logoutClick()* sono utilizzate rispettivamente per visualizzare la pagina di login e per effettuare il logout dal portale. La funzione *loginClick()* attiverà la visualizzazione nel corpo centrale della pagina contenente i campi necessari per inserire le credenziali dell'utente (username e password) e i pulsanti per procedere con l'autenticazione. La funzione *logoutClick()* ricaricherà semplicemente il contenuto del file *index.html* nel browser. Questa operazione comporterà automaticamente il log-out dal portale stesso.
- Le funzioni *homeClick()*, *matriceDinamicaClick()*, *contattiClick()* sono utilizzate per visualizzare nel corpo centrale le rispettive pagine *home*, *matrice dinamica* e *contatti*. Effettuano semplicemente la chiusura degli elementi di navigazione a più livelli se aperti, rendono non visibili tutte le pagine del corpo centrale e, quindi, rendono visibile la corrispondente pagina selezionata. Le funzioni utilizzate per rendere visibili e non visibili gli oggetti sono rispettivamente *w3.show* e *w3.hide* della libreria *w3.js*.
- Le funzioni *abachiClick()* e *soluzioniCostruttiveClick()* sono utilizzate per aprire rispettivamente i sottomenu relativi all'Atlante e alle Soluzioni costruttive. Si occupano di cambiare la rispettiva variabile di stato globale che indica se il sottomenu è visibile o meno, di rendere visibili o invisibili gli elementi che fanno parte del sottomenu e, nel caso in cui il sottomenu sta passando dallo stato non visibile a quello visibile, di richiamare le funzioni che renderanno non visibili tutti gli altri sottomenu della colonna di navigazione. Considerato che il sottomenu delle soluzioni non ha ulteriori sottomenu interni, la funzione *soluzioniCostruttiveClick()* si occuperà anche di richiamare la funzione *filterDelClick0* che cancella il contenuto di eventuali filtri attivi, e, in fase di apertura, di richiamare la funzione *myloadcaratteristicheSoluzioniCostruttive* per caricare dal database l'elenco delle opzioni da visualizzare nella relativa drop-down list.
- Le funzioni *struttureClick()*, *bestpracticesClick()* e *aziendeClick()* sono utilizzate per attivare i rispettivi elementi Strutture, Best Practices e Aziende del sottomenu Atlante. Tutte le funzioni si occuperanno di rendere visibile nel corpo del portale la relativa pagina Strutture, Best Practices o Aziende ed eventualmente, in fase di apertura, di chiudere eventuali altri sottomenu aperti presenti nel menu della colonna di navigazione. La funzione *struttureClick()* si occuperà inoltre di cambiare la rispettiva variabile di stato globale che indica se il sottomenu è visibile o meno, di richiamare la funzione *filterDelClick0* per cancellare il contenuto di eventuali filtri attivi, e, in fase di apertura, di richiamare le funzioni *myloadElementiFunzionaliAndCategories* e *myfirstshowTable* per, rispettivamente, caricare dal database l'elenco delle opzioni da visualizzare nelle due drop-down list che individuano

la tipologia e la categoria delle soluzioni costruttive e visualizzare la pagina relativa alle opzioni visualizzate per default nelle drop-down list.

- Le funzioni *myloadElementiFunzionaliAndCategories(attempt)*, *myloadCategories(attempt)*, *myloadcaratteristicheSoluzioniCostruttive(attempt)* sono utilizzate per caricare gli elementi delle drop-down list presenti nella colonna di navigazione laterale. Utilizzano tutte l'oggetto *Promise* per sincronizzare le operazioni di interrogazione del database con quelle di visualizzazione. Richiamano la funzione *getStrutture* che si occuperà di effettuare le interrogazioni del database ed aggiornare gli elementi di visualizzazione. La funzione *myloadElementiFunzionaliAndCategories()* si occuperà di caricare le opzioni della drop-down list relativa agli elementi funzionali e chiamerà, al suo completamento, la funzione *myloadCategories()* per caricare le opzioni della drop-down list relativa alle categorie dell'elemento funzionale di default. La funzione *myloadCategories()* si occuperà di caricare le opzioni della drop-down list relativa alle categorie dell'elemento funzionale attualmente selezionato nella relativa drop-down list e, al completamento, di visualizzare nel corpo centrale le informazioni relative all'elemento funzionale e alla categoria selezionate nelle rispettive drop-down list richiamando la funzione *myfirstshowTable()*. La funzione *myloadcaratteristicheSoluzioniCostruttive()* si occuperà di caricare le opzioni della drop-down list relativa alle diverse soluzioni costruttive e, al completamento, di visualizzare nel corpo centrale le informazioni relative alla soluzione costruttiva di default richiamando la funzione *myshowTable2()*.
- Le funzioni *getStrutture(strtable, strid, strwhere, cols, idpos)*, *mygetStrutture(container, strid, strwhere, cols, idpos)* e *mygetStruttureFromQuery(container, strquery, cols)* sono utilizzate per richiamare il file PHP *mygetdata.php* secondo tre diverse modalità. La funzione *getStrutture* ha come parametri di ingresso il nome di una tabella (*strtable*), l'id della query nell'array memorizzato nel file *myquery.php* (*strid*), le eventuali clausole *WHERE*, *LIMIT*, *ORDER BY* da aggiungere alla query (*strwhere*), il numero di colonne da visualizzare (*cols*) e la posizione del campo contenente la chiave univoca nella tabella (*idpos*). Dopo aver richiamato il file PHP *mygetdata.php*, se l'esecuzione è andata a buon fine, saranno automaticamente aggiornati gli oggetti corrispondenti dell'interfaccia grafica aventi per identificativo le stringhe *unime_<strtable>_data* e *unime_<strtable>_field*. Il primo oggetto sarà quello che conterrà i dati del risultato dell'applicazione della query. Il secondo oggetto sarà quello che conterrà i nomi dei campi restituiti dalla query. La funzione *getStrutture* è utilizzata per aggiornare dinamicamente il contenuto degli elementi di tipo drop-down list. A differenza della funzione *getStrutture*, la funzione *mygetStrutture* è utilizzata per aggiornare dinamicamente il contenuto degli elementi di tipo *table* con i valori contenuti nel database. In quest'ultimo caso il primo parametro di ingresso è l'identificativo dell'elemento all'interno del quale si trova la tabella e le strutture corrispondenti che saranno aggiornate automaticamente avranno identificativo *<container>_<strtable>_data* e *<container>_<strtable>_field*. Sarà inoltre aggiornato il campo che ha per identificativo *textPageLabel* presente nella pagina, che visualizza l'attuale posizionamento dell'offset nella visualizzazione della tabella, utilizzando l'informazione restituita dallo script PHP relativa al numero totale di elementi restituiti dalla query. Inoltre, a seguito della chiamata, saranno inizializzati gli elementi che permetteranno l'esecuzione delle operazioni di selezione da parte dell'utente. A differenza delle due funzioni precedenti, la funzione *mygetStruttureFromQuery* non farà riferimento all'array di query memorizzate all'interno del file *myquery.php*. Ci sono infatti dei casi in cui le query da eseguire sono memorizzate all'interno delle tabelle del database. Quest'ultima funzione, quindi, avrà un comportamento analogo alla funzione *mygetStrutture* ma la query di tipo *SELECT* da eseguire sarà direttamente passata in formato stringa come parametro di ingresso (*strquery*). In tutti i casi gli elementi di visualizzazione saranno "riempiti" utilizzando la funzione *w3.displayObject* della libreria *w3.js*.

- Le funzioni *myfirstshowTable(attempt)*, *myshowTable()*, *myshowTable2()*, *myshowTable3()*, *myshowTable4()* sono utilizzate per attivare la visualizzazione delle pagine relative alle strutture dell'atlante, alle soluzioni costruttive, alle best practices, alle aziende e alla pagina che visualizza l'interfaccia di selezione delle "soluzioni costruttive tipo" attraverso la matrice dinamica. Nello specifico la funzione *myfirstshowTable* utilizzando un oggetto *Promise* richiamerà la funzione *myshowTable* solo quando i dati sono stati correttamente importati dal database. Le funzioni *myshowTable()*, *myshowTable2()*, *myshowTable3()*, *myshowTable4()* hanno tutte la stessa struttura: aggiornano la clausola WHERE con il valore dell'attuale offset di visualizzazione all'interno della tabella; aggiornano il contenuto del sottotitolo della pagina; inseriscono il tag *<a>* per inserire nuovi elementi in maniera tale che l'evento *onclick* richiami correttamente la funzione *mycreateAndShowInsertForm* passando come parametro il nome della tabella oggetto dell'inserimento; richiamano le routine della libreria *w3.js* per visualizzare o nascondere i *<tag>* che abilitano l'inserimento, la cancellazione e la modifica a seconda che l'utente che sta utilizzando il sistema sia loggato come amministratore o meno; visualizzano la pagina così creata all'interno del corpo centrale del layout.
- Le funzioni *elementifunzionaliChange()*, *elementifunzionaliCategorieChange()*, *caratteristicheSoluzioniCostruttiveChange()* sono richiamate a seguito dell'evento *onchange* delle rispettive drop-down list. La funzione *elementifunzionaliChange* chiamerà al suo interno la funzione *myloadCategories* per caricare le categorie relative all'elemento funzionale attualmente selezionato. Quindi sarà aggiornato il corpo centrale del portale richiamando la funzione *myfirstshowTable*. Le funzioni *elementifunzionaliCategorieChange()* e *caratteristicheSoluzioniCostruttiveChange()* si limitano ad aggiornare il corpo centrale del portale richiamando rispettivamente la funzione *myfirstshowTable* o la funzione *myshowTable()*. In tutti e tre i casi saranno cancellati gli eventuali filtri inseriti dall'utente chiamando la funzione *filterDelClick0*.
- Le funzioni *nextClick(container)*, *prevClick(container)* spostano rispettivamente avanti e indietro l'attuale offset di visualizzazione delle tabelle di una quantità pari al passo prefissato (per default pari a 10). Lo spostamento viene effettuato verificando che non si scenda al di sotto del valore 0 e non si salga al di sopra del numero massimo di righe contenute nella tabella. Effettuato lo spostamento, si richiamerà la funzione di visualizzazione relativa alla tabella contenuta nell'elemento contenitore dal quale le funzioni sono state richiamate (l'identificativo dell'elemento contenitore è passato come parametro) in maniera tale da aggiornarne la visualizzazione a partire dal nuovo offset impostato.
- Le funzioni *filterClick(container)*, *filterDelClick0(container)* e *filterDelClick(container)* sono utilizzate per gestire la clausola WHERE di filtraggio degli elementi della tabella. In particolare la funzione *filterClick()* permette di aggiungere una condizione alla clausola WHERE. La condizione è impostata dall'utente scegliendo negli appositi elementi di tipo drop-down list il nome del campo e il tipo di condizione ("CONTIENE", "NON CONTIENE", "UGUALE A", "MINORE DI", "MAGGIORE DI") e inserendo nell'apposita casella di testo il valore di confronto. Agendo sul tasto di inserimento sarà richiamata la funzione *filterClick()* che aggiungerà alle eventuali clausole già inserite quest'ultima, concatenandola utilizzando l'operatore AND. Le funzioni *filterDelClick0()* e *filterDelClick()* sono invece utilizzate per cancellare tutte le clausole inserite di filtraggio. La funzione *filterDelClick0()* si limita a cancellare le clausole inserite e ad aggiornare la visualizzazione degli elementi del filtro. La funzione *filterDelClick()*, dopo aver richiamato la funzione *filterDelClick0()*, procede a richiamare le funzioni per aggiornare la visualizzazione delle tabelle.
- La funzione *myDMFilter()* implementa le operazioni necessarie per selezionare e visualizzare le "soluzioni costruttive tipo" che soddisfano ai valori impostati dall'utente nella pagina denominata matrice dinamica. In funzione del numero dei campi di selezione presenti nella tabella *dm_schema*

come illustrato nella sezione 3.2.4, viene creata una clausola WHERE contenente una cascata di condizioni legate dall'operatore AND. Nella struttura della query è presente un campo *TOT* che totalizza per ogni riga i valori dei campi *Costi investimento*, *Efficienza*, *Complessità Impianto*, *Integrazione FER*, *Detraibilità*. Tali campi contengono degli indici prestazionali della corrispondente soluzione memorizzata nella riga. La loro somma (l'operazione di somma potrebbe essere sostituita con qualunque altro tipo di funzione che lega eventualmente in maniera pesata tali indici) sarà quindi considerata come indice di prestazione globale della soluzione e utilizzata per presentare le soluzioni stesse in ordine decrescente di tale indice utilizzando la clausola ORDER BY. Effettuata la query di selezione sarà quindi richiamata la funzione *mygetStrutture()* per visualizzare le soluzioni costruttive tipo selezionate.

- Le funzioni *mycreateInsertForm(tablename)* e *mycreateModifyForm(id, container)* sono utilizzate per creare le pagine che permettono rispettivamente l'inserimento e la modifica degli elementi delle tabelle contenute nel database. Entrambe richiamano il file PHP *mygettableschema.php* per ottenere le informazioni relative alla tipologia di campi contenute nella tabella. Utilizzando un ciclo che itera su tutti i campi inseriranno all'interno della pagina le etichette che identificano il nome di ogni campo e poi, in funzione della tipologia del campo (che ricordiamo può essere TEXT, SELECT, NUMERIC, SELECTCHECK) inseriranno:
 - una casella di inserimento testo (se il tipo campo è TEXT o NUMERIC);
 - una drop-down list (se il tipo campo è SELECT);
 - una tabella dove saranno successivamente inseriti gli elementi di tipo checkbox (se il tipo campo è SELECTCHECK)

Infine, saranno inseriti gli appositi tag *<a>* contenenti le icone che, al verificarsi dell'evento *onclick*, permetteranno di richiamare le funzioni per effettuare l'inserimento/modifica del nuovo elemento o annullare l'operazione.

- Le funzioni *mycreateAndShowInsertForm(tablename, attempt)* e *myshowInsertForm()* sono utilizzate per visualizzare all'interno del corpo centrale del portale le pagine di inserimento di nuovi elementi nella tabella. La prima funzione, utilizzando un oggetto di tipo *Promise*, richiama la routine *mycreateInsertForm(tablename)* e attende che siano creati tutti gli elementi di visualizzazione attraverso un meccanismo a semaforo. La funzione *myshowInsertForm()* itera su tutti i campi della tabella: nel caso in cui la tipologia di visualizzazione del campo è di tipo SELECT richiama la funzione *mygetStruttureFromQuery()* per inizializzare correttamente i valori delle opzioni della drop-down list; nel caso in cui la tipologia del campo di visualizzazione è di tipo SELECTCHECK richiama la routine *myshowSelectCheck()* che si occuperà di creare e visualizzare il numero corretto di elementi di tipo check-box con le relative etichette all'interno della corrispondente tabella creata.
- Le funzioni *mycreateAndShowModifyForm(id, container, attempt)*, *myshowModifyForm0(attempt)* e *myshowModifyForm()* operano analogamente a quelle viste nel punto precedente ma per visualizzare nel corpo centrale le pagine di modifica degli elementi delle tabelle. Per questo motivo, rispetto al caso precedente, è stato necessario inserire la funzione aggiuntiva *myshowModifyForm()* che si occupa di caricare all'interno di tutte le caselle di testo i valori già memorizzati per ogni rispettivo campo la cui visualizzazione è di tipo TEXT o NUMERIC, di impostare la visualizzazione corretta della opzione memorizzata negli elementi di tipo drop-down list per i campi il cui tipo di visualizzazione è SELECT e di impostare correttamente a checked/not checked i checkbox dei campi il cui tipo di visualizzazione è SELECTCHECK. La funzione *myshowModifyForm0*, analogamente alla funzione *myshowInsertForm()*, itera su tutti i campi della tabella e crea le etichette e i rispettivi elementi di modifica a seconda della tipologia di visualizzazione.
- La funzione *myshowSelectCheck(container, strquery, cols)* esegue la visualizzazione degli elementi di tipo checkbox all'interno dell'elemento contenitore di tipo *table* il cui *id* è specificato nel

parametro *container*. Utilizzando la funzione PHP *mygetdatafromquery.php* viene eseguita la query specificata nel parametro *strquery* per recuperare il numero e le etichette da assegnare a ciascun checkbox. Quindi saranno creati gli elementi di tipo checkbox e inseriti nella tabella in ordine a partire dalla prima riga in alto a sinistra. Ogni qualvolta il numero di checkbox inseriti nella riga supera il valore specificato dal parametro *cols* sarà creata una nuova riga. A ciascun elemento di tipo *checkbox* sarà assegnato un identificatore univoco dipendente dal nome specificato nel parametro *container* e dal nome del campo ad esso relativo estratto dalla query. In questo modo sarà possibile successivamente identificarlo univocamente all'interno del codice.

- La funzione *myOKInsertModifyForm(strtable, insertmodify, id)* viene richiamata quando l'utente decide di procedere con l'inserimento o la modifica di una riga di una tabella. Il campo *strtable* identifica il nome della tabella, il campo *insertmodify* specifica se si tratta di un inserimento o una modifica, il campo *id*, valido solo nel caso di modifica, specifica la chiave univoca della riga della tabella che si sta per modificare. La funzione itera su tutti i nomi dei campi della tabella in oggetto per creare la stringa *columnnamesstr* contenente tutti i nomi dei campi secondo la sintassi della query di inserimento/modifica di SQL (saranno esclusi i campi che contengono il tipo di visualizzazione impostato a SELECTCHECK). All'interno dello stesso ciclo sarà anche creata la stringa *valuesstr* contenente tutti i valori da assegnare ai campi secondo la stessa sintassi. La generazione dei valori da inserire in questa seconda stringa sarà condizionata dalla tipologia di visualizzazione del campo in oggetto: nel caso di campi con visualizzazione di tipo TEXT/NUMERIC sarà semplicemente letto il contenuto della rispettiva casella di testo ed eventualmente verificate le condizioni di validità del valore; nel caso di campi con visualizzazione di tipo SELECT sarà letto il valore dell'opzione della corrispondente drop-down list attualmente selezionata; nel caso di campi con visualizzazione di tipo SELECTCHECK i valori non dovranno essere inseriti direttamente nella tabella in oggetto perché dovrà essere modificata la corrispondente tabella di collegamento, sarà quindi aggiornata un'opportuna struttura dati che sarà successivamente utilizzata senza modificare la stringa *valuesstr*. Create le stringhe *columnnamesstr* e *valuesstr* sarà richiamato il file PHP *myinsertdata.php* per inserire/modificare i valori nella tabella. Se l'inserimento va a buon fine sarà anche richiamato per ogni campo di tipo SELECTCHECK il file *myupdateX.php* in maniera tale da aggiornare anche i valori contenuti nelle tabelle di collegamento secondo le selezioni che l'utente avrà fatto su ogni gruppo di checkbox.

Se tutte le operazioni legate all'inserimento/modifica saranno correttamente effettuate sulla tabella del database verrà dato un messaggio informativo di corretto inserimento/modifica e la pagina sarà chiusa distruggendo tutti gli elementi appositamente creati. In caso contrario verrà segnalato che qualcosa non è andato a buon fine nell'operazione e si rimarrà nella stessa pagina.

La funzione *myNOKInsertModifyForm()* sarà invece richiamata se l'utente decide di annullare l'operazione di inserimento o modifica, si limita a distruggere gli elementi creati per la pagina di inserimento/modifica e richiamare la visualizzazione della home page.

- La funzione *mycreateAndShowDeleteForm(id, container, attempt)* viene richiamata quando un utente decide di cancellare una riga di una determinata tabella. Dopo aver letto i valori dei primi tre campi della riga che si desidera cancellare utilizzando il file PHP *mygetondata.php*, viene visualizzata all'utente una finestra che richiede una conferma di cancellazione. Se la risposta dell'utente è positiva si tenta di cancellare l'elemento utilizzando il file PHP *mydeletedata.php*. Se la riga sarà correttamente cancellata verrà mostrato un messaggio informativo di corretta cancellazione, altrimenti verrà segnalato che non è stato possibile procedere con la cancellazione.
- Le funzioni *myOklogin()* e *myNOKlogin()* sono richiamate rispettivamente quando l'utente decide di procedere con l'operazione di login dopo aver inserito *username* e *password* nei rispettivi campi oppure quando decide di non procedere con il login. Nel primo caso sarà richiamato il file *myIsUser.php* per verificare se le credenziali inserite sono presenti nella tabella di login del database.

In caso positivo viene impostata la variabile *userType*, visualizzato nella colonna di navigazione il nome completo dell'utente e sostituita la label di login con quella di logout. La funzione *myNOKlogin()* procede semplicemente a nascondere la pagina di login dal corpo centrale per visualizzare la home page.

- La funzione *mypdfshow(id, container)* è utilizzata per visualizzare i file pdf eventualmente associati alle varie righe delle tabelle. Dopo aver recuperato il nome della tabella attualmente visualizzata nell'elemento *container*, chiede al browser di visualizzare in una nuova finestra il file contenuto all'interno della cartella che si trova nel percorso *pdf/<nometabella>* e che ha per nome *<id>.pdf*.

4 Popolamento dell'Atlante con i dati forniti da POLIMI prodotti nella LA2.12

L'output dell'attività condotta da POLIMI nella LA2.12 è consistito nella produzione di una serie di schede in formato pdf relative a diversi prodotti (suddivisi per tipologia e categoria), di una serie di schede in formato pdf relative alle aziende produttrici di tali prodotti e ad una serie di schede in formato pdf relative a soluzioni costruttive denominate "Best Practices". L'insieme di tali schede è stato denominato "atlante".

Gli obiettivi della nostra linea di attività includevano l'analisi di queste schede in maniera tale da realizzare le strutture dati del database necessarie per potere contenere tutte le informazioni dell'atlante e il successivo popolamento del database con le informazioni contenute nelle schede ricevute.

Data la mole elevata di schede da analizzare si è deciso di automatizzare questo processo scrivendo delle routine che utilizzano il linguaggio di comandi *bash* [10], il linguaggio di manipolazione dei testi *awk* [11], il programma di manipolazione di file pdf *pdftk* [12], il programma *pdftotext* [13] per convertire il contenuto dei file pdf in formato testo e alcuni comandi della shell di linux come *grep*, *echo*, *mkdir*, *sed*.

4.1 L'acquisizione delle schede prodotti

Le schede prodotti sono state fornite in file pdf per ciascun elemento funzionale in 9 distinti file pdf. Ogni file pdf contiene diverse schede prodotti ordinate per categoria. A seconda dell'elemento funzionale una scheda prodotto può essere costituita da una o due pagine. Nella Figura 35 è visibile un esempio di scheda prodotto a due pagine. Come si può osservare le informazioni sono inserite all'interno di una struttura tabellare anche se il formato non è uniforme sia sulle varie righe sia fra le diverse schede (in alcuni casi la prima colonna è vuota perché il dato contiene diverse categorie elencate nella seconda colonna, in alcuni casi i nomi dei campi o i valori dei campi si estendono su più righe, il formato della struttura tabellare non è omogeneo per tutte le schede).

Lo script *bash* illustrato in Figura 36 calcola il numero di pagine del file pdf passato come parametro d'ingresso, crea una cartella con lo stesso nome del file ma senza estensione e poi richiama, all'interno di un ciclo, lo script illustrato dalla Figura 37 alla Figura 40 per analizzare le singole schede prodotto contenute nel file.

La parte iniziale dello script, illustrato nella Figura 37, utilizzando il comando *pdftotext* estrae le pagine relative a una singola scheda prodotto mantenendone il layout e, utilizzando il comando *sed*, inserisce il carattere ":" come delimitatore di campo degli elementi di ogni riga.

Successivamente, la seconda parte dello script illustrata in Figura 38, utilizza il comando *awk* per riempire i campi vuoti delle prime due colonne su ciascuna riga con il nome del campo presente nella riga precedente ed eliminare la presenza di eventuali caratteri spazio multipli e caratteri spazio e tabulazioni all'inizio e alla fine di ogni riga.

La terza parte dello script, illustrata nella Figura 39, utilizzando il comando *awk* si occupa di fare il merge dei nomi dei campi quando la colonna dei valori è distribuita su più righe.

Infine, l'ultima parte dello script illustrata nella Figura 40, sempre utilizzando il comando *awk*, sposta le eventuali unità di misura presenti nella colonna dei valori alla colonna del nome del campo.


Le operazioni illustrate dalla Figura 37 alla Figura 40 sono poi ripetute più o meno similmente per analizzare la seconda pagina.

Lo standard output dello script sarà ridiretto in un file testuale che ha per nome un progressivo numerico e per estensione il suffisso ".txt".


A seguito dell'esecuzione dello script illustrato in Figura 36, all'interno della cartella creata si genereranno tanti file con estensione ".txt" quante sono le schede prodotto presenti nell'originario file pdf.

A titolo d'esempio nella Figura 41 è illustrato il file di testo generato analizzando la scheda prodotto illustrata nella Figura 35.

Gli stessi script sono stati poi opportunamente modificati per analizzare le schede prodotto contenenti un'unica pagina e anche le schede contenenti le informazioni delle aziende e le schede Best Practices.



RICERCA DI SISTEMA ELETTRICO



POLITECNICO MILANO 1863
POLO TERRITORIALE DI LECCO

01.A - Lastre Standard
GYPROC FLEX 6

Produttore	Saint-Gobain Gyproc	
Campo di applicazione	Pareti - Contropareti - Controsoffitti	
Composizione	Gesso rivestito	
Dimensioni	<i>Spessore</i>	6 mm
	<i>Larghezza</i>	1200 mm
	<i>Lunghezza</i>	3000 mm
Bordi	<i>Longitudinale</i>	Assottigliato
	<i>Di testa</i>	Dritto
Peso	5,40 kg/m ²	
Densità	900 kg/m ³	
Percentuale di foratura	-	
Classe di reazione al fuoco	A2-s1,d0	
Carico di rottura a flessione	<i>Longitudinale</i>	258 N
	<i>Trasversale</i>	100 N
Resistenza a flessione	<i>Longitudinale</i>	- N/mm ²
	<i>Trasversale</i>	- N/mm ²
Modulo elastico	- N/mm ²	
Durezza	<i>Superficiale</i>	- mm
	<i>Brinell</i>	- N/mm ²
Conducibilità termica	0,25 W/mK	
Fattore di resistenza alla diffusione di vapore	<i>In campo secco</i>	10
	<i>In campo umido</i>	4
Assorbimento d'acqua	<i>Superficiale</i>	- g/m ²
	<i>Totale</i>	- %
Riduzione formaldeide	-	
Note	Per soluzioni curve	

Accordo di Programma Ministero dello Sviluppo Economico - ENEA / Piano Triennale di Realizzazione 2019/2021 - II annualità
 Progetto: 1.6 "Efficienza energetica dei prodotti e dei processi industriali"
 Work package: WP2 - Miglioramento dell'efficienza energetica di processi di produzione e di gestione dell'ambiente costruito
 Linea di attività: LA2.12 - "La costruzione stratificata a secco: analisi dello stato dell'arte, raccolta e catalogazione di relativi sistemi e tecnologie"

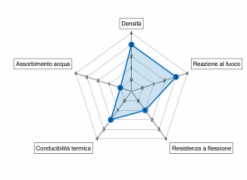


Figura 35: Generico formato di una scheda prodotto

```
#!/bin/bash
echo -n "Calcolo numero delle pagine "
N=`pdftk $1 dump_data | grep NumberOfPages | awk 'BEGIN {FS=":"} {print $2}'`
echo $N
dirname=${1%.*}
if ! [ -d "$dirname" ]; then
    mkdir $dirname
fi
n=1;
for (( i=1; i<=$N; i+=2 ))
do
    echo "ANALIZZO PAGINA: "$i;
    nstr=$(printf "%04d" $n)
    ./analisi_una_scheda.sh $1 $i > $dirname/$nstr.txt
    n=$((n+1))
done
```

Figura 36: script bash per analizzare un file pdf di schede prodotti

```
#!/bin/bash

# CREO IL FILE TXT DELLA PAGINA PRIMA PAGINA
pdftotext -f $2 -l $2 -layout $1 page1.txt
# NECESSARIO TOGLIERE EVENTUALI ":"
sed -E 's/://=/' < page1.txt > page1_00.txt
sed -E 's/ +/ /g' < page1_00.txt > page1_0.txt
sed -E 's/Produttore/\nAzienda/g' < page1_0.txt > page1_1.txt
```

Figura 37: script bash per analizzare una singola scheda prodotto (parte 1)

```
awk 'BEGIN {FS=":"; i=0;}
{
    i=i+1;
    gsub("^[ ]+", "", $0);
    if (length($0)==0) {
        a0="";
        for (j=1; j<i; j++) {
            a0 = a0 " " a[j]
        };
        #lastb=b[1];
        for (j=1; j<i; j++) {
            if (length(c[j])>0) {
                print a0, b[j], ":", c[j];
                lastb=b[j];
            }
            else {
                if (i==2)
                    print a0, ":", b[j];
                else
                    print a0, lastb, ":", b[j];
            }
        };
        i=0;
        lastb="";
    };
    a[i]=$1; b[i]=$2; c[i]=$3;
}' <page1_1.txt > page1_2.txt

awk '{ gsub(/[ ]+/, " "); print $0}' < page1_2.txt > page1_3.txt
awk '{ gsub(/^[ \t]+|[ \t]+$/, ""); print $0}' <page1_3.txt > page1_4.txt
```

Figura 38: script bash per analizzare una singola scheda prodotto (parte 2)

```
awk 'BEGIN {FS=":"; i=0; prev=""; value="";}
{
    if (i<2) {
        if (i==0)
            print "Tipo", $0;
        else
            print "Denominazione", $0;
    } else {
        if (($1 == prev) && (index($1, "Permeabilità al vapore")==0)) {
            gsub(/-N/, "- N", $2);
            gsub(/-%/, "- %", $2);
            value=value " " $2;
        }
        else {
            if (i>2) {
                print prev,":", value;
            }
            prev=$1;
            gsub(/-N/, "- N", $2);
            gsub(/-%/, "- %", $2);
            gsub(/-m/, "- m", $2);
            value=$2;
        }
    }
    i=i+1;
}
END {print prev,":", value;}' <page1_4.txt > page1_5.txt
```

Figura 39: script bash per analizzare una singola scheda prodotto (parte 3)

```
awk 'BEGIN {FS=":"; i=0; split(" m, mm, N, g/m2, kg/m3, N/mm2, W/mK, kg/m2, %, %, dB,
W/m2K, mg/m h Pa, kPa, MN/m3, J/kgK, kPa s/m2", units, "");
split("m,mm,N,g/m2,kg/m3,N/mm2,W/mK,kg/m2,%,%,dB,
W/m2K,mg/m h Pa,kPa,MN/m3,J/kgK,kPa s/m2 ", unitsstr, "")}
{
    if (i<6) {
        print $0;
    } else {
        jpos=0;
        for (j=1; j<=length(units); j++) {
            pos=index($2, units[j]);
            if (pos > 0) {
                jpos=pos;
                junits=j;
            }
        }
        if (jpos==0) {
            app=$2;
            gsub(",",".",app);
            if (strtonum(app)==0)
                print $0;
            else
                print $1" : "strtonum(app);
        }
        else {
            gsub(/[ ]+/, " ", $1);
            valustr=substr($2, 1, jpos-1);
            gsub(",",".",valustr);
            print $1["unitsstr[junits]"] ":"valustr
        }
    }
    i=i+1;
}' <page1_5.txt
```

Figura 40: script bash per analizzare una singola scheda prodotto (parte 4)

Tipo : 01.A - Lastre Standard
Denominazione : GYPROC FLEX 6
Azienda : Saint-Gobain Gyproc
Campo di applicazione : Pareti - Contropareti - Controsoffitti
Composizione : Gesso rivestito
Dimensioni Spessore [mm] : 6
Dimensioni Larghezza [mm] : 1200
Dimensioni Lunghezza [mm] : 3000
Bordi Longitudinale : Assottigliato
Bordi Di testa : Dritto
Peso [kg/m²] : 5.40
Densità [kg/m³] : 900
Percentuale di foratura : -
Classe di reazione al fuoco : A2-s1,d0
Carico di rottura a flessione Longitudinale [N] : 258
Carico di rottura a flessione Trasversale [N] : 100
Resistenza a flessione Longitudinale [N/mm²] : -
Resistenza a flessione Trasversale [N/mm²] : -
Modulo elastico [N/mm²] : -
Durezza Superficiale [mm] : -
Durezza Brinell [N/mm²] : -
Conducibilità termica [W/mK] : 0.25
Fattore di resistenza alla diffusione di vapore In campo secco : 10
Fattore di resistenza alla diffusione di vapore In campo umido : 4
Assorbimento d'acqua Superficiale [g/m²] : -
Assorbimento d'acqua Totale [%] : -
Riduzione formaldeide : -
Note : Per soluzioni curve

Figura 41: esempio file di testo generato dall'analisi di una scheda prodotto

4.2 La generazione delle query SQL di creazione tabelle e inserimento dati

Ottenuti i file in formato testo, si sono creati due ulteriori script *bash* per generare automaticamente le query SQL per creare le tabelle e per inserire i dati nelle stesse.

Il primo script, illustrato nella Figura 42, genera la query di creazione delle tabelle relative ad uno specifico elemento funzionale. Iterando su tutti i file con estensione “.txt” presenti nella cartella specificata e aventi il formato illustrato nella sottosezione precedente, scriverà su due appositi file con estensione “.sql” le istruzioni di creazione di una tabella ogni qualvolta verrà riscontrata una variazione del contenuto del campo “Tipo”. Sono contemporaneamente generati i file con le query per creare le tabelle con i dati e per creare le tabelle di tipo “schema” con l’inserimento delle informazioni relative ai vari campi.

Nella Figura 43 è visualizzata la parte iniziale del file generato contenente le istruzioni SQL per creare le strutture delle tabelle dati relative agli elementi funzionali “lastre”. Nella Figura 44 è invece visualizzata la parte iniziale del file generato contenente le istruzioni SQL per creare le strutture delle tabelle di tipo “_schema”, sempre in relazione agli elementi funzionali “lastre”. Come si può osservare, oltre alla query che permette la creazione della tabella, è anche presente una query di inserimento attraverso la quale vengono generate le righe relative a tutti i campi presenti nella tabella dati. La generazione automatica di tali righe prevede l'impostazione sempre a “TEXT” della tipologia di visualizzazione mentre saranno posti a NULL tutti gli altri campi della riga. Se si vorranno variare questi contenuti bisognerà poi agire manualmente all'interno del database, utilizzando ad esempio *PHPMysqlAdmin*.

```

#!/bin/bash
echo -n "" >create_$.sql
echo -n "" >createschema_$.sql
lasttipo=""
for filename in */*.txt
do
    tipo=`head -n 1 $filename | awk 'BEGIN {FS=":"} {gsub(/[ ]+/, " ",$2); print $2}`
    if [[ $tipo != $lasttipo ]]; then
        tablename=`echo $tipo | awk 'BEGIN {FS="-"} {gsub(/[ ]+/, "",$1);
            gsub(/^[ ]+/, "",$2); pos=index($2, " ");
            print tolower(substr($2,1,pos-1))"_"toupper(substr($1,length($1),1))}`
        echo "CREATE TABLE `ambiente_costruito`.$tablename
            (`id` INT NOT NULL AUTO_INCREMENT," >> create_$.sql
        awk 'BEGIN {FS=":"; i=0;} {if (i>0) {gsub(/^[ ]+/, "",$1); gsub(/[ ]+$/, "",$1);
            print ""$1` TEXT NULL,`; i=i+1;}' < $filename >> create_$.sql
        echo "PRIMARY KEY (`id`)) ENGINE = InnoDB;" >> create_$.sql

        echo "CREATE TABLE `ambiente_costruito`.$tablename`_`schema
            (`id` INT NOT NULL AUTO_INCREMENT," >> createschema_$.sql
        echo "none_campo TEXT NULL," >> createschema_$.sql
        echo "tipo_campo TEXT NULL," >> createschema_$.sql
        echo "minimo DOUBLE NULL," >> createschema_$.sql
        echo "massimo DOUBLE NULL," >> createschema_$.sql
        echo "notempty TINYINT(1) NULL," >> createschema_$.sql
        echo "querystring TEXT NULL," >> createschema_$.sql
        echo "PRIMARY KEY (`id`)) ENGINE = InnoDB;" >> createschema_$.sql
        echo "TRUNCATE `ambiente_costruito`.$tablename`_`schema;" >> createschema_$.sql
        echo "ALTER TABLE `ambiente_costruito`.$tablename`_`schema AUTO_INCREMENT = 1;" >> createschema_$.sql
        echo "INSERT INTO `ambiente_costruito`.$tablename`_`schema" >> createschema_$.sql
        echo "(`none_campo`, `tipo_campo`)" >> createschema_$.sql
        echo "VALUES" >> createschema_$.sql
        a=$( awk 'BEGIN {FS=":"; i=0;} {if (i>0) {gsub(/^[ ]+/, "",$1); gsub(/[ ]+$/, "",$1);
            print "("$1"`, "`TEXT`)",`; i=i+1;}' < $filename )
        echo "${a::-1};" >> createschema_$.sql
        lasttipo=$tipo;
    fi
done

```

Figura 42: script bash per generare la query SQL di creazione delle tabelle

```

CREATE TABLE `ambiente_costruito`.lastre_A (`id` INT NOT NULL AUTO_INCREMENT,
`Denominazione` TEXT NULL,
`Azienda` TEXT NULL,
`Campo di applicazione` TEXT NULL,
`Composizione` TEXT NULL,
`Dimensioni Spessore [mm]` TEXT NULL,
`Dimensioni Larghezza [mm]` TEXT NULL,
`Dimensioni Lunghezza [mm]` TEXT NULL,
`Bordi Longitudinale` TEXT NULL,
`Bordi Di testa` TEXT NULL,
`Peso [kg/m2]` TEXT NULL,
`Densità [kg/m3]` TEXT NULL,
`Percentuale di foratura` TEXT NULL,
`Classe di reazione al fuoco` TEXT NULL,
`Carico di rottura a flessione Longitudinale [N]` TEXT NULL,
`Carico di rottura a flessione Trasversale [N]` TEXT NULL,
`Resistenza a flessione Longitudinale [N/mm2]` TEXT NULL,
`Resistenza a flessione Trasversale [N/mm2]` TEXT NULL,
`Modulo elastico [N/mm2]` TEXT NULL,
`Durezza Superficiale [mm]` TEXT NULL,
`Durezza Brinell [N/mm2]` TEXT NULL,
`Conducibilità termica [W/mK]` TEXT NULL,
`Fattore di resistenza alla diffusione di vapore In campo secco` TEXT NULL,
`Fattore di resistenza alla diffusione di vapore In campo umido` TEXT NULL,
`Assorbimento d'acqua Superficiale [g/m2]` TEXT NULL,
`Assorbimento d'acqua Totale [%]` TEXT NULL,
`Riduzione formaldeide` TEXT NULL,
`Note` TEXT NULL,
PRIMARY KEY (`id`)) ENGINE = InnoDB;
CREATE TABLE `ambiente_costruito`.lastre_B (`id` INT NOT NULL AUTO_INCREMENT,
`Denominazione` TEXT NULL,
`Azienda` TEXT NULL,
`Campo di applicazione` TEXT NULL,
`Composizione` TEXT NULL,
`Dimensioni Spessore [mm]` TEXT NULL,
`Dimensioni Larghezza [mm]` TEXT NULL,
`Dimensioni Lunghezza [mm]` TEXT NULL,
...
...

```

Figura 43: spezzone di file contenente le query di creazione delle tabelle dati

```

CREATE TABLE `ambiente_costruito`.lastre_A_schema (`id` INT NOT NULL AUTO_INCREMENT,
nome_campo TEXT NULL,
tipo_campo TEXT NULL,
minimo DOUBLE NULL,
massimo DOUBLE NULL,
notempty TINYINT(1) NULL,
querystring TEXT NULL,
PRIMARY KEY (`id`)) ENGINE = InnoDB;
TRUNCATE `ambiente_costruito`.lastre_A_schema;
ALTER TABLE `ambiente_costruito`.lastre_A_schema AUTO_INCREMENT = 1;
INSERT INTO `ambiente_costruito`.lastre_A_schema
(nome_campo`, `tipo_campo`)
VALUES
("Denominazione", "TEXT"),
("Azienda", "TEXT"),
("Campo di applicazione", "TEXT"),
("Composizione", "TEXT"),
("Dimensioni Spessore [mm]", "TEXT"),
("Dimensioni Larghezza [mm]", "TEXT"),
("Dimensioni Lunghezza [mm]", "TEXT"),
("Bordi Longitudinale", "TEXT"),
("Bordi Di testa", "TEXT"),
("Peso [kg/m2]", "TEXT"),
("Densità [kg/m3]", "TEXT"),
("Percentuale di foratura", "TEXT"),
("Classe di reazione al fuoco", "TEXT"),
("Carico di rottura a flessione Longitudinale [N]", "TEXT"),
("Carico di rottura a flessione Trasversale [N]", "TEXT"),
("Resistenza a flessione Longitudinale [N/mm2]", "TEXT"),
("Resistenza a flessione Trasversale [N/mm2]", "TEXT"),
("Modulo elastico [N/mm2]", "TEXT"),
("Durezza Superficiale [mm]", "TEXT"),
("Durezza Brinell [N/mm2]", "TEXT"),
("Conducibilità termica [W/mK]", "TEXT"),
("Fattore di resistenza alla diffusione di vapore In campo secco", "TEXT"),
("Fattore di resistenza alla diffusione di vapore In campo umido", "TEXT"),
("Assorbimento d'acqua Superficiale [g/m2]", "TEXT"),
("Assorbimento d'acqua Totale [%]", "TEXT"),
("Riduzione formaldeide", "TEXT"),
("Note", "TEXT");
CREATE TABLE `ambiente_costruito`.lastre_B_schema (`id` INT NOT NULL AUTO_INCREMENT,
nome_campo TEXT NULL,
tipo_campo TEXT NULL,
minimo DOUBLE NULL,
massimo DOUBLE NULL,
notempty TINYINT(1) NULL,
querystring TEXT NULL,
...
...

```

Figura 44: spezzone di file contenente le query di creazione delle tabelle "_schema"

Lo script *bash* illustrato in Figura 45 è stato utilizzato per generare i file contenenti le istruzioni SQL necessarie a popolare le tabelle dati degli elementi funzionali presenti nell'atlante. Anche in questo caso, attraverso un ciclo che itera per tutti i file di testo (formato ".txt") contenuti nella cartella specificata come parametro, sarà creata una query di inserimento che specifica il valore assunto da ciascuna riga della tabella. Sarà creata una istruzione di inserimento SQL per ciascuna delle categorie presenti specificando di volta in volta i nomi dei campi corrispondenti. La Figura 46 visualizza la parte iniziale del file generato automaticamente contenente le istruzioni SQL per popolare la tabella relative agli elementi funzionali "lastre" di tipologia "standard" (tabella "lastre_A").


```
#!/bin/bash
echo -n "" >riempi_1.sql
lasttipo=""
for filename in $(ls *.txt)
do
    tipo=$(head -n 1 $filename | awk 'BEGIN {FS=":"} {gsub(/[ ]+/, " ",$2); print $2}')
    if [[ $tipo != $lasttipo ]]; then
        if [[ $lasttipo != "" ]]; then
            echo ";" >> riempi_1.sql
        fi
        tablename=$(echo $tipo | awk 'BEGIN {FS="@"} {gsub(/[ ]+/, "",$1);
            gsub(/^[ ]+/, "",$2); pos=index($2, " ");
            print tolower(substr($2,1,pos-1)) "toupper(substr($1,length($1,1))}')
        echo "INSERT INTO `ambiente_costruito`.$tablename" >> riempi_1.sql
        echo "(" >> riempi_1.sql
        a=$( awk 'BEGIN {FS=":"; i=0;} {if (i>0) {gsub(/^[ ]+/, "",$1);
            gsub(/[ ]+$/, "",$1); print "\"" $1 "\""; i=i+1;}' < $filename )
        echo "${a::-1}" >> riempi_1.sql
        echo ") VALUES" >> riempi_1.sql
        lasttipo=$tipo;
        echo "(" >> riempi_1.sql
    else
        echo ",(" >> riempi_1.sql
    fi
    a=$( awk 'BEGIN {FS=":"; i=0;} {if (i>0) {gsub(/^[ ]+/, "",$2);
        gsub(/[ ]+$/, "",$2); print "\"" $2 "\""; i=i+1;}' < $filename )
    echo "${a::-1}" >> riempi_1.sql
    echo -n ")" >> riempi_1.sql
done
echo ";" >> riempi_1.sql
```

Figura 45: script bash utilizzato per popolare automaticamente le tabelle degli elementi funzionali dell'atlante

```

INSERT INTO `ambiente_costruito`.lastre_A
(`Denominazione`,
`Azienda`,
`Campo di applicazione`,
`Composizione`,
`Dimensioni Spessore [mm]`,
`Dimensioni Larghezza [mm]`,
`Dimensioni Lunghezza [mm]`,
`Bordi Longitudinale`,
`Bordi Di testa`,
`Peso [kg/m2]`,
`Densità [kg/m3]`,
`Percentuale di foratura`,
`Classe di reazione al fuoco`,
`Carico di rottura a flessione Longitudinale [N]`,
`Carico di rottura a flessione Trasversale [N]`,
`Resistenza a flessione Longitudinale [N/mm2]`,
`Resistenza a flessione Trasversale [N/mm2]`,
`Modulo elastico [N/mm2]`,
`Durezza Superficiale [mm]`,
`Durezza Brinell [N/mm2]`,
`Conducibilità termica [W/mK]`,
`Fattore di resistenza alla diffusione di vapore In campo secco`,
`Fattore di resistenza alla diffusione di vapore In campo umido`,
`Assorbimento d'acqua Superficiale [g/m2]`,
`Assorbimento d'acqua Totale [%]`,
`Riduzione formaldeide`,
`Note`)
VALUES
("GYPROC LISAPLAC 13","Saint-Gobain Gyproc","Pareti - Contropareti - Controsoffitti",
"Gesso rivestito","12.5","1200","3000","Assottigliato","Dritto",
"9.70","770","-","A1","550","210","-","-","-","-","0.21","10","4","-","-","-","-"),
("GYPROC LISAPLAC 15","Saint-Gobain Gyproc","Pareti - Contropareti - Controsoffitti",
"Gesso rivestito","15","1200","3000","Assottigliato","Dritto",
"11.80","780","-","A1","650","250","-","-","-","-","0.21","10","4","-","-","-","-"),
("GYPROC WALLBOARD 10","Saint-Gobain Gyproc","Pareti - Contropareti - Controsoffitti",
"Gesso rivestito","9.5","1200","2000 - 2500 - 2800 - 3000","Assottigliato","Dritto",
"7.30","770","-","A2-s1,d0","400","160","-","-","-","-","0.21","10","4","-","-","-","-"),
...
...
...

```

Figura 46: spezzone di file contenente le istruzioni SQL per popolare automaticamente le tabelle contenenti gli elementi funzionali dell’atlante

Per popolare le rimanenti tabelle dell’atlante relative alle schede delle “Best Practices” e delle Aziende sono stati utilizzati analoghi script con le modifiche necessarie per gestire i diversi formati.

Utilizzando il tool *pdftk* sono stati quindi creati, a partire dai file pdf che raggruppavano tutte le schede, dei file pdf singoli aventi per nome la struttura *<id>.pdf* e memorizzati all’interno di una cartella avente lo stesso nome della corrispondente tabella. L’intero contenuto di queste cartelle è stato quindi trasferito all’interno del filesystem del server in maniera tale da rendere i file accessibili per la visualizzazione se richiesto durante la navigazione del portale. L’associazione fra nome del file e corrispondente riga della tabella è stata effettuata sfruttando la chiave primaria della stessa, costituita da un campo intero che si incrementa automaticamente ad ogni nuovo inserimento: al primo inserimento le varie righe avranno quindi una chiave primaria corrispondente alla loro posizione relativa rispetto all’inizio della tabella e, quindi, corrispondente al nome assegnato a ciascun file pdf.

Il processo ha comportato l’inserimento del seguente numero di schede e il popolamento delle rispettive tabelle con altrettante righe:

- 84 aziende
- 40 bestpractices
- 71 lastre – standard
- 45 lastre – idrorepellenti
- 65 lastre – protezione dal fuoco
- 47 lastre – alta resistenza

- 71 Lastre – acustiche
- 30 Lastre – per sottofondi
- 21 Lastre – per esterni
- 44 Pannelli a doghe – standard
- 43 Pannelli a doghe – idrorepellenti
- 76 Pannelli a doghe – protezione dal fuoco
- 169 Pannelli a doghe – acustici
- 56 Pannelli a doghe – irrigidimento
- 2435 Pannelli Sandwich
- 206 Isolamento – di origine vegetale
- 299 Isolamento – di origine minerale
- 4 Isolamento – di origine animale
- 505 Isolamento – di origine sintetica
- 299 Isolamento – di origine composita
- 221 Isolamento – al calpestio
- 299 Isolamento – di origine minerale
- 67 Membrane traspiranti – barriera al vapore
- 78 Membrane traspiranti – freno al vapore
- 139 Membrane traspiranti – membrane traspiranti
- 234 Membrane impermeabilizzanti – membrane bituminose
- 35 Membrane impermeabilizzanti – membrane polimeriche
- 91 Membrane impermeabilizzanti – membrane autoprotette e ardesiate
- 1 Membrane impermeabilizzanti – membrane vegetali
- 186 Rasanti, collanti e stucchi
- 38 Sottofondi alleggeriti
- 323 Profili e accessori – orditure
- 115 Profili e accessori – accessori per controsoffitto
- 119 Profili e accessori – profili angolari e paraspigoli

5 Descrizione del portale

5.1 Introduzione

Il portale web è attualmente disponibile all'indirizzo <http://eneadi.unime.it/portale/>. Accedendo a tale indirizzo, utilizzando un qualunque browser, sarà visualizzata la home page come illustrato in Figura 47. La home page contiene una descrizione degli obiettivi generali con un video di presentazione del progetto 1.6, un diagramma con la struttura del progetto e un diagramma descrittivo del WP2.

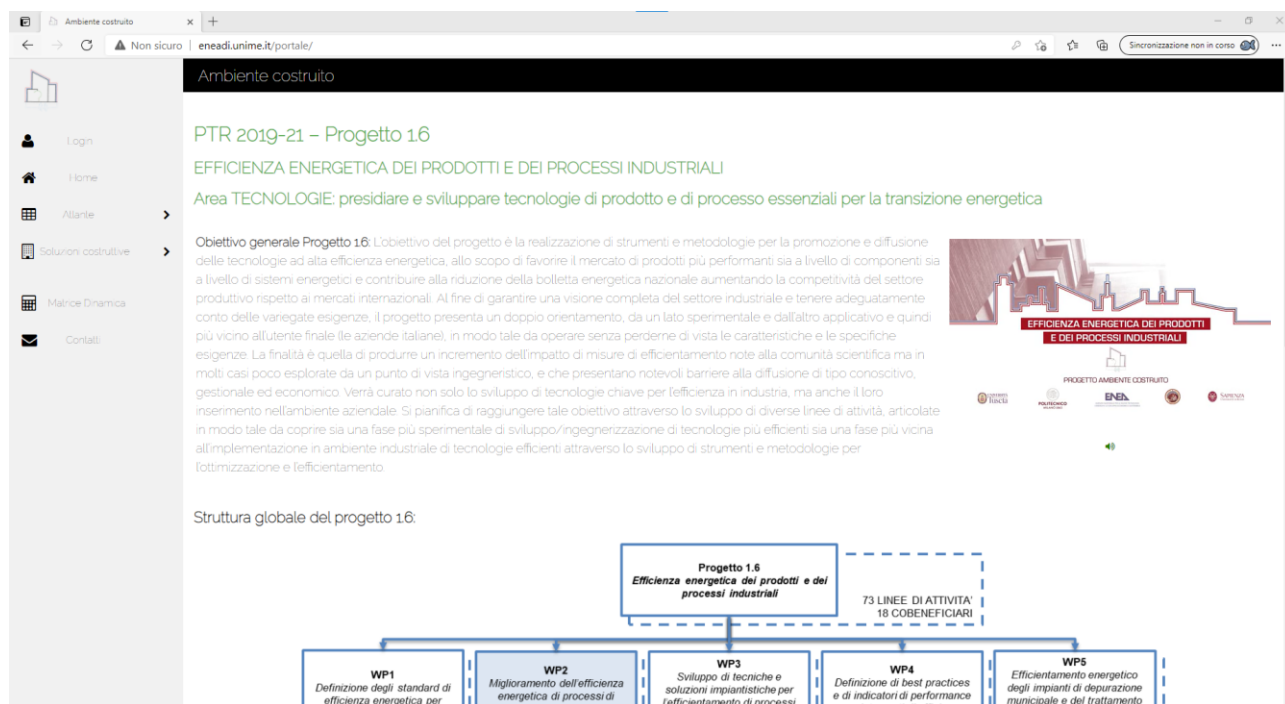


Figura 47: Home page del portale

Saranno descritte in un primo momento le funzionalità del portale che possono funzionare anche senza un'autenticazione esplicita da parte dell'utente (modalità *anonymous user*).

A tali utenti il portale permette di:

- visualizzare (anche utilizzando dei filtri di selezione per la ricerca) le informazioni contenute nell'atlante degli elementi funzionali, delle "best practices" e delle aziende fornito da POLIMI attraverso la LA2.12 e visualizzare le relative schede tecniche associate;
- visualizzare le soluzioni costruttive per il miglioramento dell'efficienza energetica dell'edificio (tipologie costruttive, tipologie impianti, zone climatiche e sismiche), che devono essere tenute in considerazione dal progettista secondo le disposizioni impostate da ENEA;
- visualizzare le "soluzioni costruttive tipo" preparate da ENEA (anche utilizzando dei filtri di selezione per la ricerca) all'interno del catalogo impostato da ENEA e visualizzare le relative schede tecniche associate a ciascuna soluzione;
- visualizzare e utilizzare la pagina della "matrice dinamica" che permette di impostare delle specifiche condizioni costruttive a partire da determinate condizioni costruttive impostate da ENEA e di valutare quali "soluzioni costruttive tipo" ne soddisfano i requisiti.

5.2 La sezione Atlante

5.2.1 La sezione "Strutture" dell'Atlante

Facendo click su "Atlante" e su "Strutture" della colonna di navigazione saranno visibili sulla stessa due elementi di tipo drop-down list che permetteranno la selezione di un elemento funzionale e della relativa categoria. Effettuando una selezione su tali drop-down list sarà visualizzato nel corpo centrale del portale la

relativa pagina come illustrato nella Figura 48. Come illustrato in figura, il sottotitolo della pagina visualizzerà la tipologia e la categoria dell'elemento selezionato. Sarà possibile applicare dei filtri per visualizzare nella tabella solo i prodotti che soddisfano a certe condizioni. Per applicare i filtri bisogna scegliere un campo da filtrare nella drop-down list per selezionare i campi, selezionare un operatore di confronto fra quelli disponibili (CONTIENE, NON CONTIENE, UGUALE A, MINORE DI, MAGGIORE DI), scrivere nella input-box il valore di confronto e fare click sull'icona in blu rappresentante un imbuto.

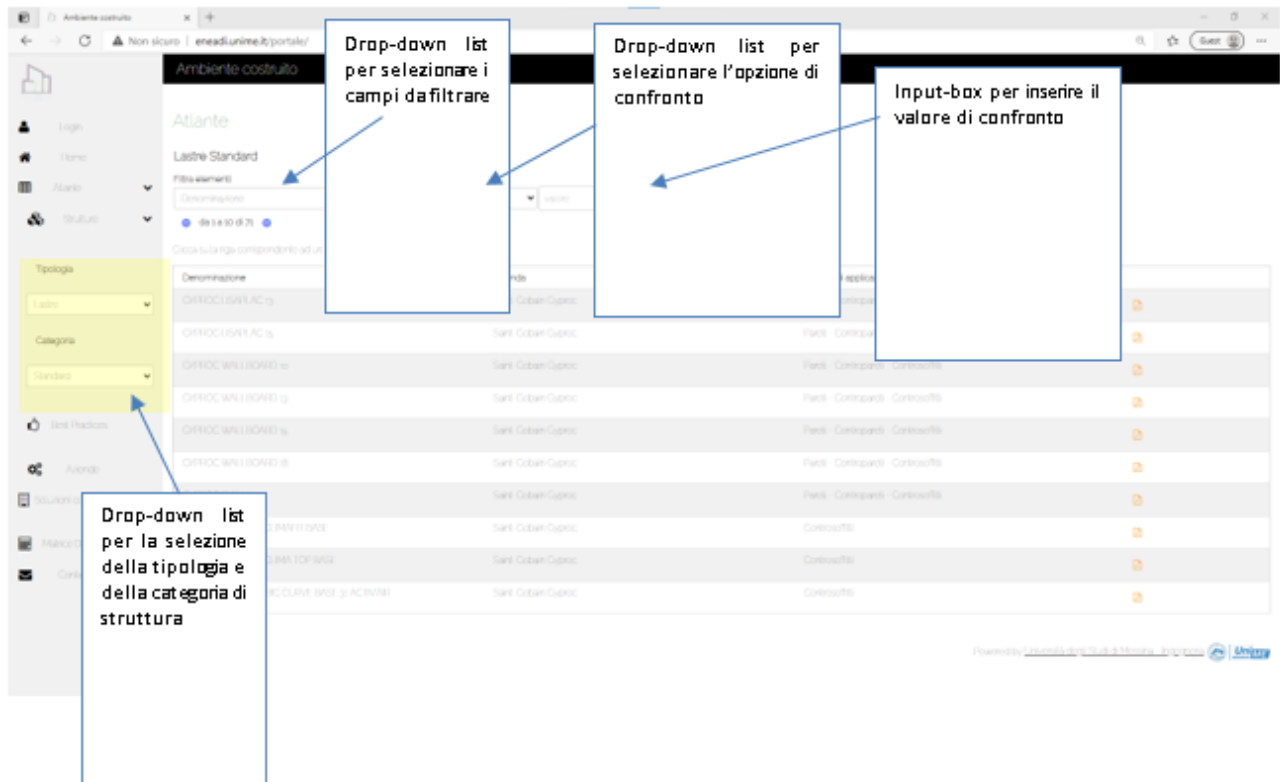


Figura 48: Visualizzazione del contenuto del portale relativa ad Atlante - Strutture Lastre Standard

È possibile concatenare più condizioni di filtraggio selezionando nuovamente un campo, un operatore di confronto, un valore di confronto e facendo nuovamente click sul tasto di applicazione del filtro. Le diverse condizioni inserite saranno concatenate mediante l'operatore logico AND, quindi, devono essere tutte soddisfatte. La Figura 49 illustra il risultato dell'applicazione della condizione (visibile in basso ai campi di applicazione del filtro) per selezionare le lastre che riportano un valore del campo "Dimensioni Spessore [mm]" "UGUALE A" "25" e contestualmente un valore del campo "Dimensioni Lunghezza [mm]" "UGUALE A" "600". Come si può vedere solo un elemento soddisfa le condizioni richieste. Per rimuovere tutte le condizioni di filtraggio applicate si può fare click sull'icona con la forma a croce di colore rosso visualizzando nuovamente tutti gli elementi presenti in tabella. Le due icone a forma di freccia presenti sotto il filtro permettono di spostarsi fra i vari elementi della tabella che sono visualizzati a blocchi di 10.

Facendo click su una singola riga della tabella è possibile visualizzare il contenuto di tutti i valori dei campi come illustrato nella Figura 50. Il comportamento sarà di tipo "toggle", quindi facendo click su una riga il cui contenuto è stato allargato per la visualizzazione di tutti i campi, la visualizzazione della riga tornerà ad essere di tipo compatto.

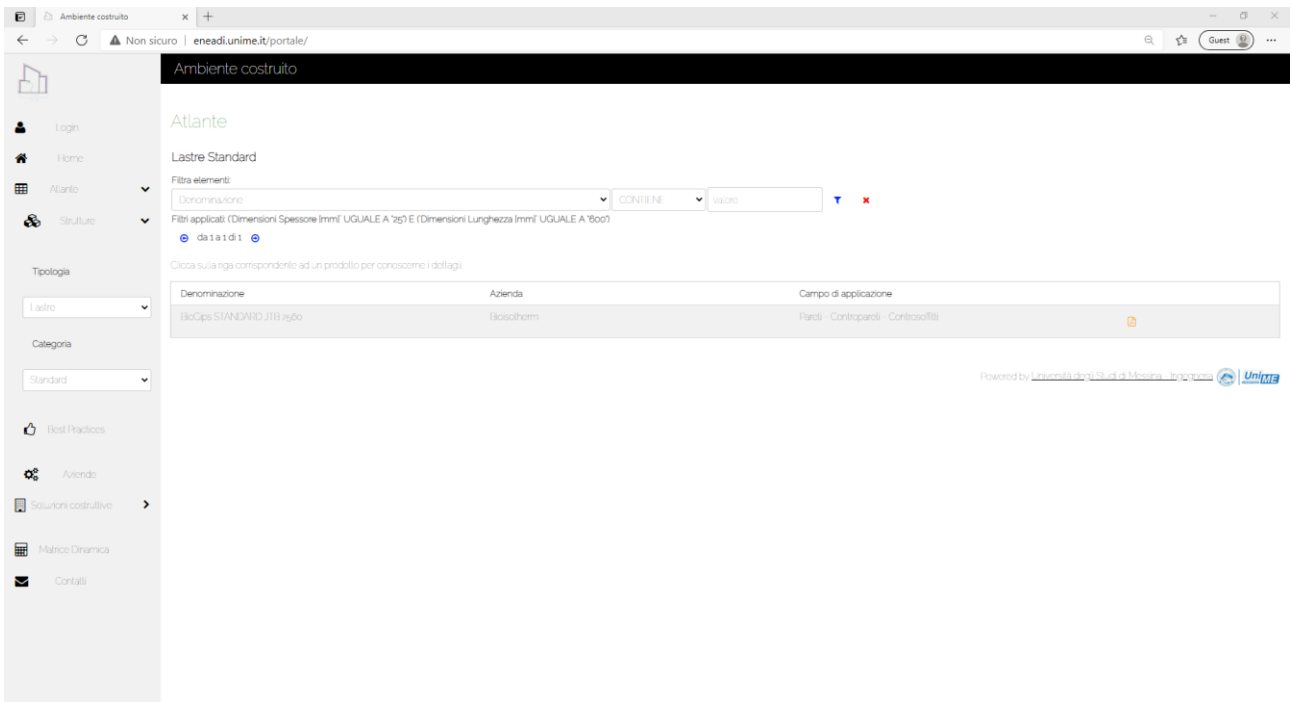


Figura 49: Visualizzazione del contenuto del portale relativa ad Atlante - Strutture - Lastre Standard con l'applicazione di un filtro costituito da due condizioni

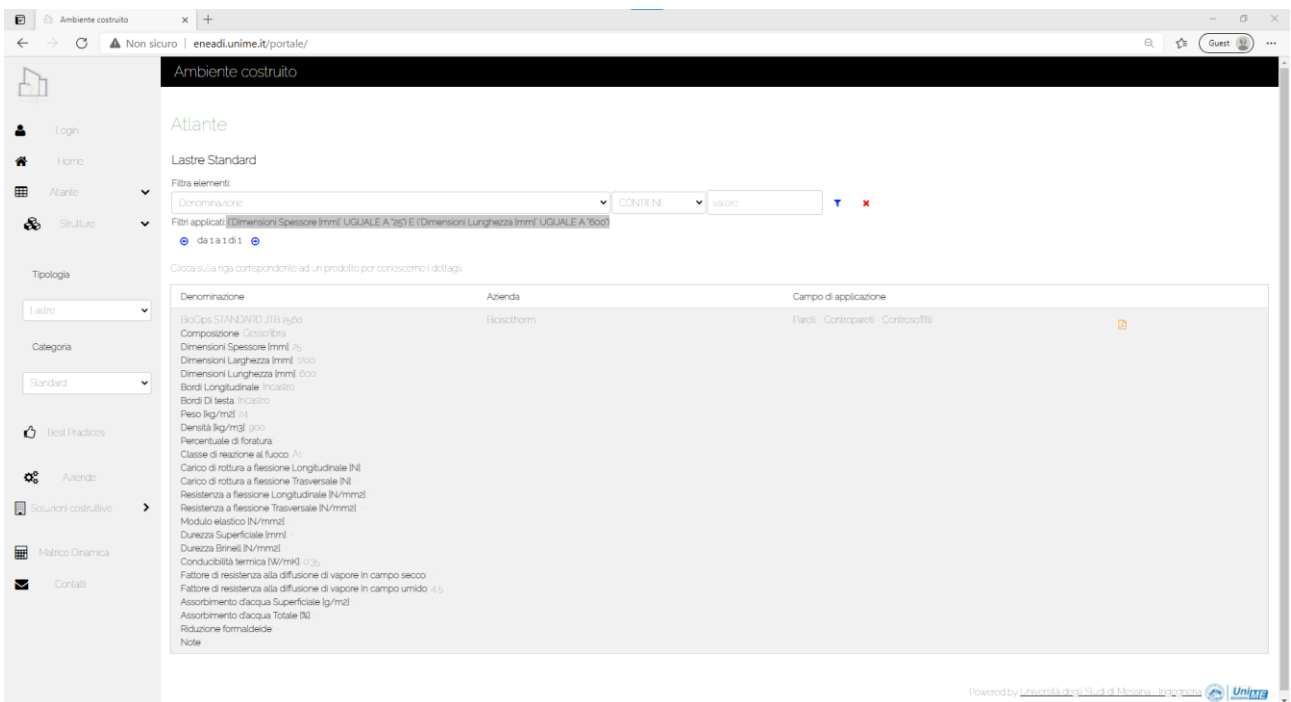


Figura 50: Visualizzazione del contenuto del portale relativa ad Atlante - Strutture - Lastre Standard con l'applicazione di un filtro costituito da due condizioni e l'espansione della visualizzazione dei campi di una riga della tabella

Su ogni riga sarà inoltre presente un'icona con il simbolo dei file di tipo "pdf" di colore arancio che permetterà di aprire su una nuova finestra del browser la scheda tecnica del relativo prodotto.

La Figura 51 visualizza l'apertura di una scheda prodotto relativa alla sezione Atlante – Strutture – Lastre Standard. La scheda così aperta e visualizzata potrà anche essere memorizzata o stampata dall'utente utilizzando la barra degli strumenti messa a disposizione dal proprio browser.

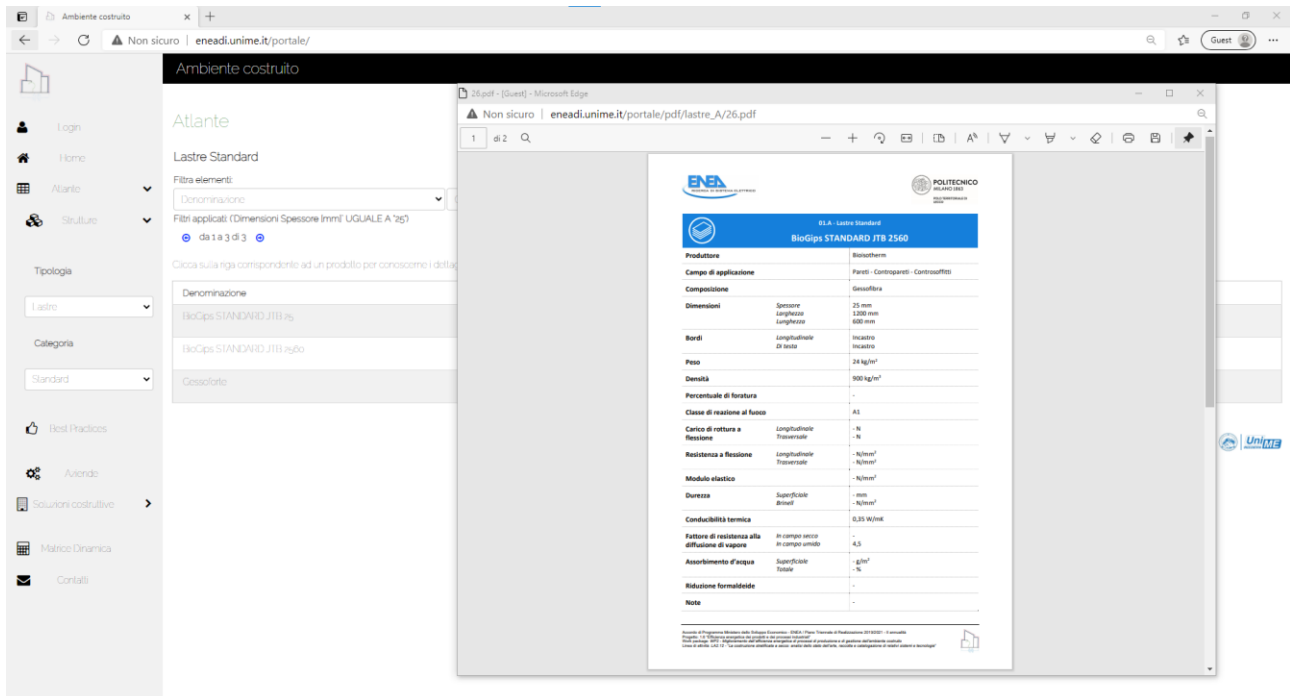


Figura 51: Apertura di una scheda prodotto relativa ad uno specifico elemento di Atlante – Strutture – Lastre Standard

Gli utenti amministratori avranno una visualizzazione che permette di inserire nuovi elementi per ciascuna tipologia e categoria di elemento funzionale appartenente alla struttura e anche di modificare e/o cancellare elementi già inseriti.

Questa possibilità sarà data dalla visualizzazione di un'apposita icona di inserimento e, per ciascuna riga di un'icona per la modifica e un'icona per la cancellazione come evidenziato nella Figura 52. Facendo click sulle rispettive icone di inserimento o modifica saranno visualizzate nel corpo centrale le pagine che permettono l'inserimento o la modifica dei valori dei diversi campi dello specifico elemento funzionale. Inseriti o modificati i valori l'utente potrà decidere se confermare l'operazione o annullarla. Facendo click sul pulsante di cancellazione verrà richiesto all'utente una conferma dell'operazione; l'elemento corrispondente sarà cancellato esclusivamente a seguito della risposta positiva a tale conferma. Le operazioni di modifica e cancellazione, una volta eseguite, non sono annullabili.

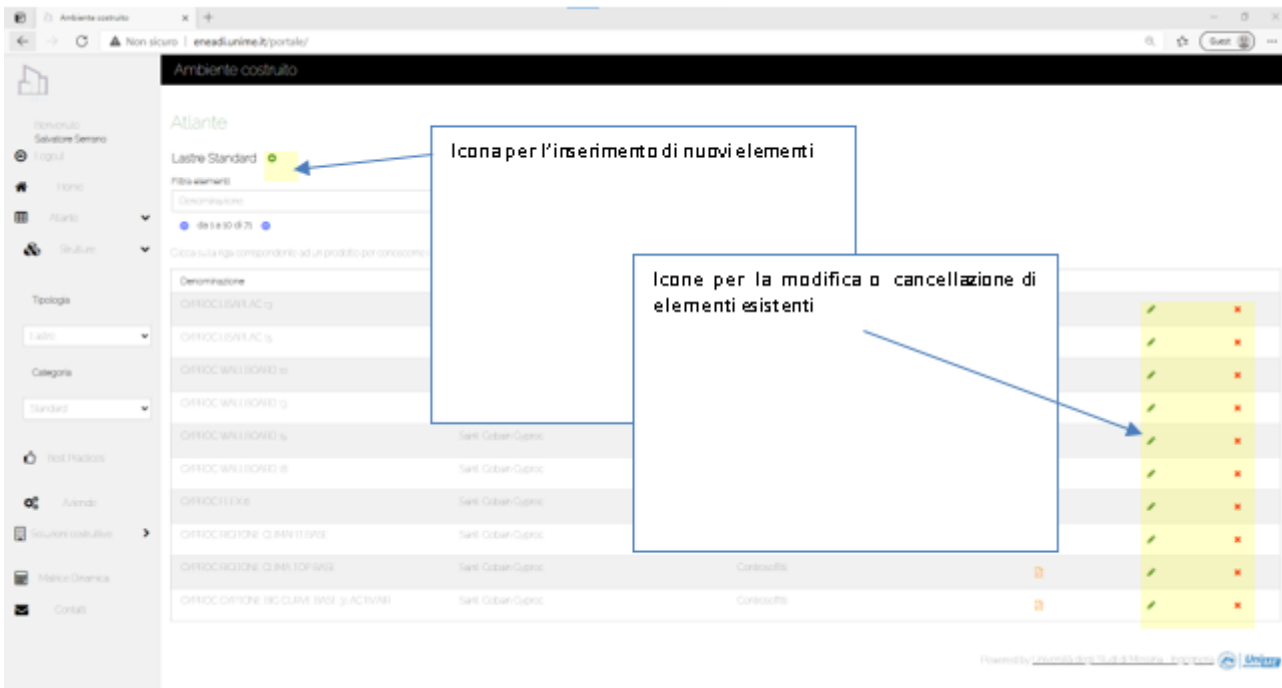


Figura 52: Visualizzazione del contenuto del portale relativa ad Atlante - Strutture Lastre Standard con privilegi di amministratore

5.2.2 La sezione “Best Practices” dell’Atlante

Facendo click su “Atlante” e su “Best Practices” della colonna di navigazione si aprirà sul corpo centrale del portale la pagina di visualizzazione delle “Best Practices” come illustrato in Figura 53.

Come descritto nel punto precedente sarà possibile applicare dei filtri di ricerca e visualizzare le schede correlate ad una specifica riga.

La Figura 54 visualizza l’apertura di una scheda relativa a una specifica Best Practice dell’Atlante. La scheda così aperta e visualizzata potrà anche essere memorizzata o stampata dall’utente utilizzando la barra degli strumenti messa a disposizione dal proprio browser.

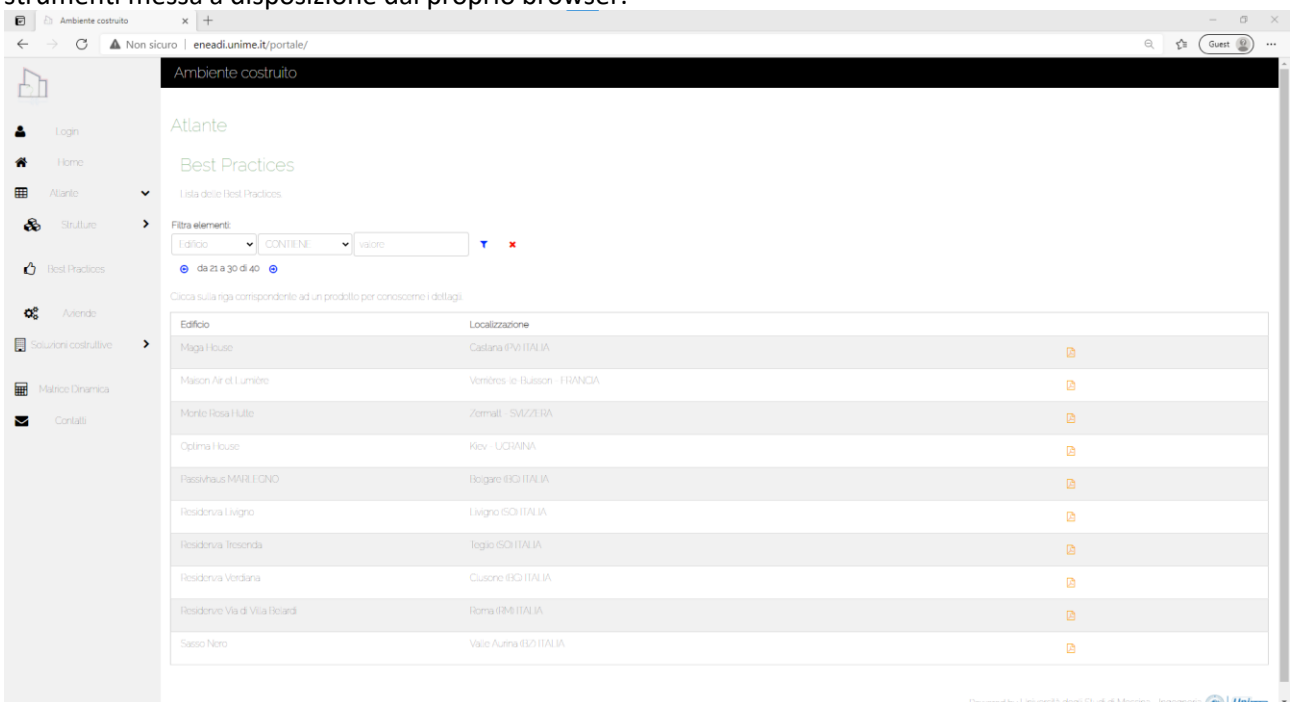


Figura 53: Visualizzazione del contenuto del portale relativa ad Atlante - Best Practices

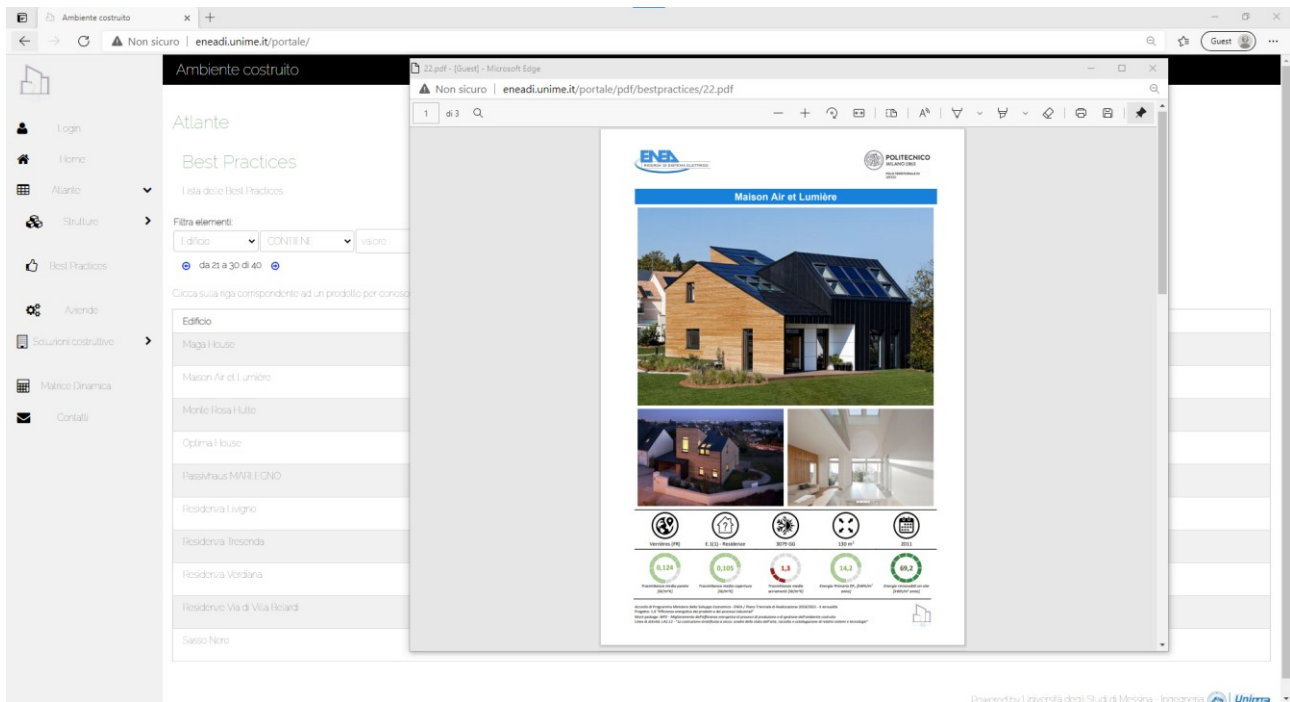


Figura 54: Apertura di una scheda relativa ad una specifica “Best Practice” dell’Atlante

Gli utenti amministratori avranno una visualizzazione che permette di inserire nuovi elementi e anche di modificare e/o cancellare elementi già inseriti analoga a quella illustrata nel punto precedente.

5.2.3 La sezione “Aziende” dell’Atlante

Facendo click su “Atlante” e su “Aziende” della colonna di navigazione si aprirà sul corpo centrale del portale la pagina di visualizzazione delle “Aziende” come illustrato in Figura 55.

Come descritto nei punti precedenti sarà possibile applicare dei filtri di ricerca e visualizzare le schede relative ad una specifica riga.

La Figura 56 visualizza l’apertura di una scheda relativa a una specifica Azienda dell’Atlante. La scheda così aperta e visualizzata potrà anche essere memorizzata o stampata dall’utente utilizzando la barra degli strumenti messa a disposizione dal proprio browser.

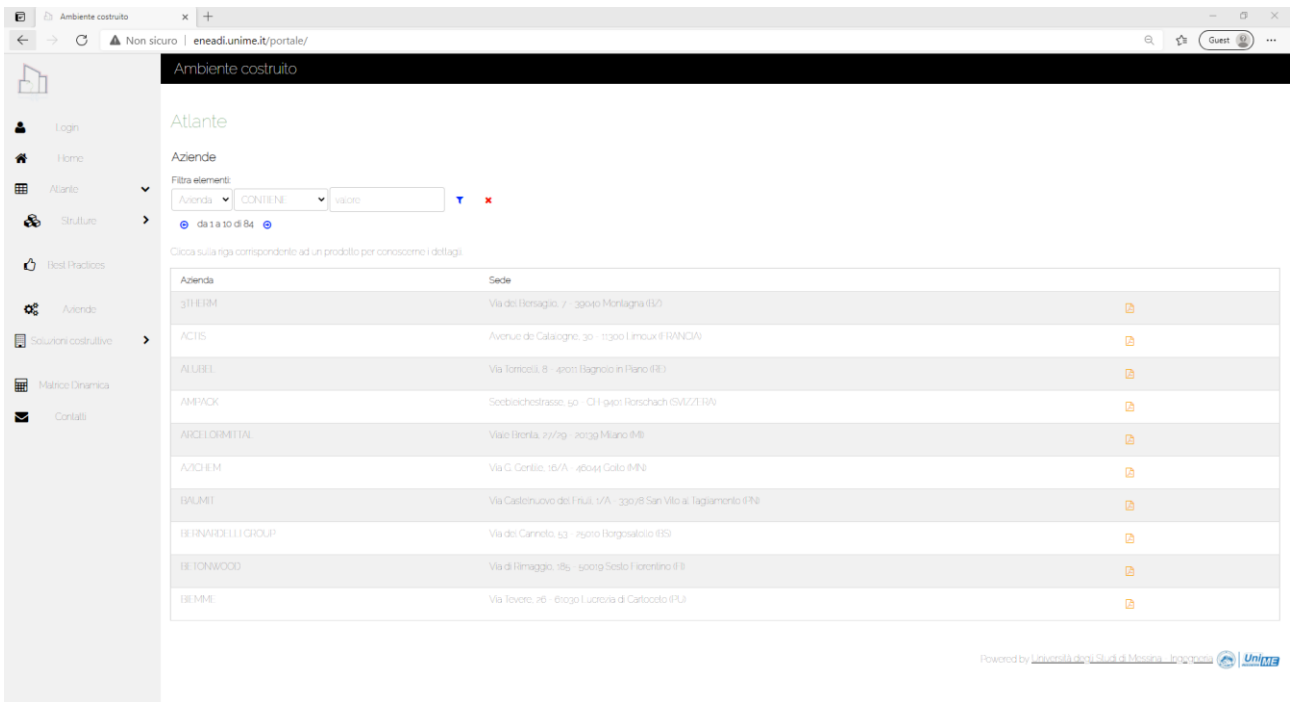


Figura 55: Visualizzazione del contenuto del portale relativa ad Atlante – Aziende

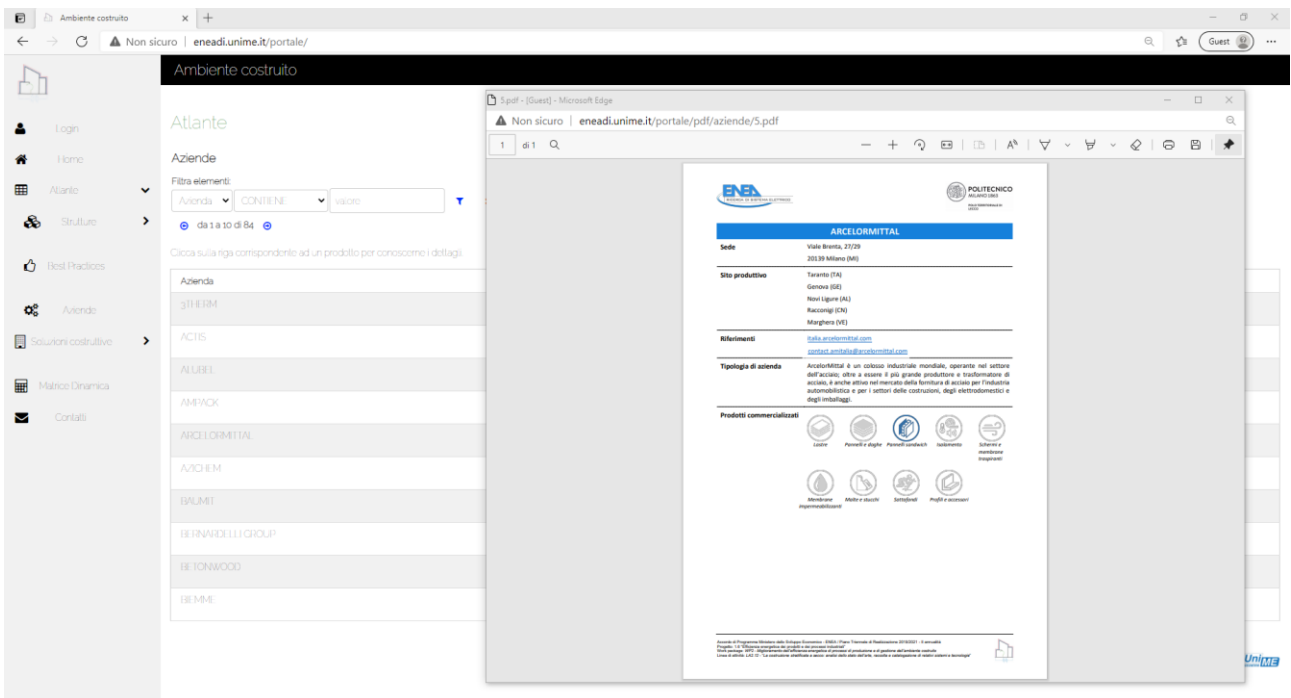


Figura 56: Apertura di una scheda relativa ad una specifica azienda presente in Atlante

Gli utenti amministratori avranno una visualizzazione che permette di inserire nuovi elementi e anche di modificare e/o cancellare elementi già inseriti analoga a quella illustrata nei punti precedenti.

5.3 La sezione Soluzioni Costruttive

La sezione “Soluzioni Costruttive” serve, essenzialmente agli amministratori, per definire i contenuti delle tabelle che permettono di implementare la matrice dinamica delle scelte.

Gli utenti anonimi potranno comunque consultarne il contenuto senza alterare le modifiche delle tabelle.

Dopo aver fatto click sull'apposito elemento "Soluzioni Costruttive" presente nella colonna di navigazione verticale, sarà mostrata sempre sulla stessa colonna, una drop-down list contenente come opzioni di scelta le condizioni costruttive valutabili ("Tipologie Costruttive", "Tipologie Impianti", "Zone Climatiche", "Zone Sismiche", "Tipologie Isolanti", "Tipologie Impianti Esistenti") e l'opzione "Soluzioni Tipo".

Selezionando una specifica opzione nella drop-down list sarà possibile visualizzare nel corpo centrale del portale la rispettiva tabella come illustrato dalla Figura 57 alla Figura 62 in relazione rispettivamente a "Tipologie Costruttive", "Tipologie Impianti", "Zone Climatiche", "Zone Sismiche", "Tipologie Isolanti", "Tipologie Impianti Esistenti". Allo stato attuale tutte queste tabelle contengono esclusivamente due campi: il campo codice e il campo descrizione. Come si può osservare quindi tutte le tabelle visualizzate avranno la stessa struttura e non sarà possibile espandere il contenuto delle righe perché tutti i campi sono già visualizzati.

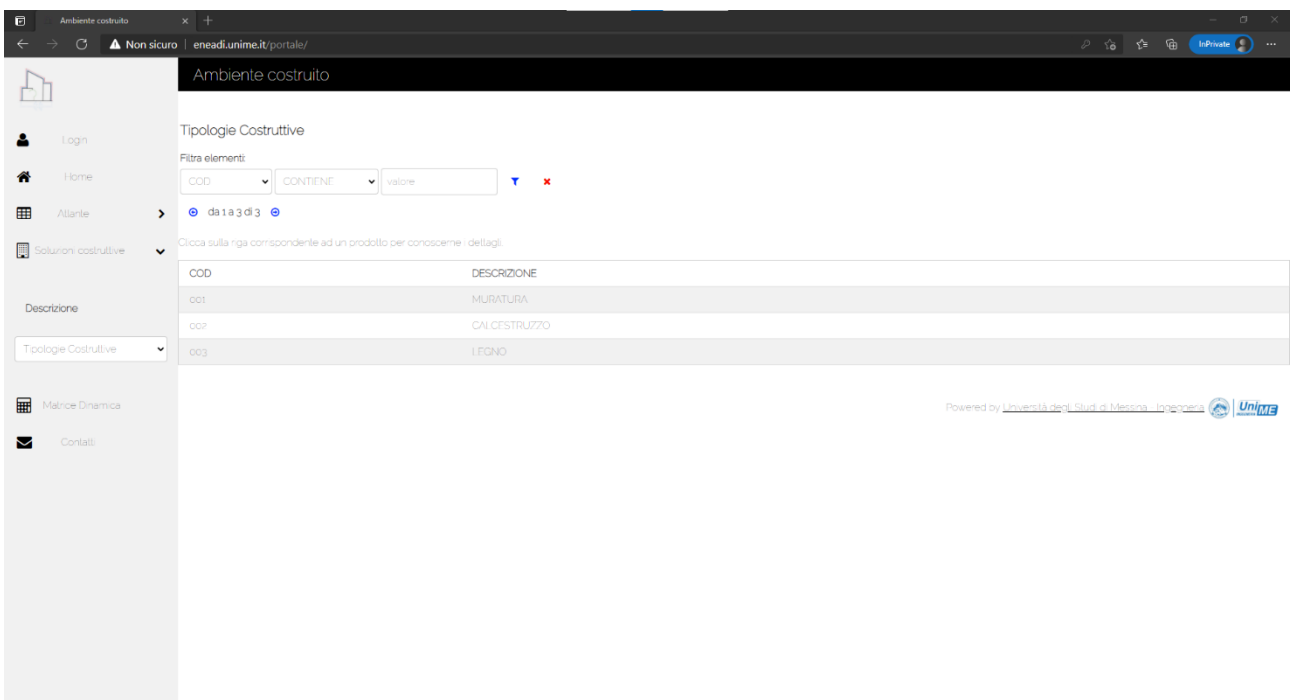


Figura 57: Visualizzazione del contenuto del portale relativa a "Soluzioni Costruttive" - "Tipologie Costruttive"

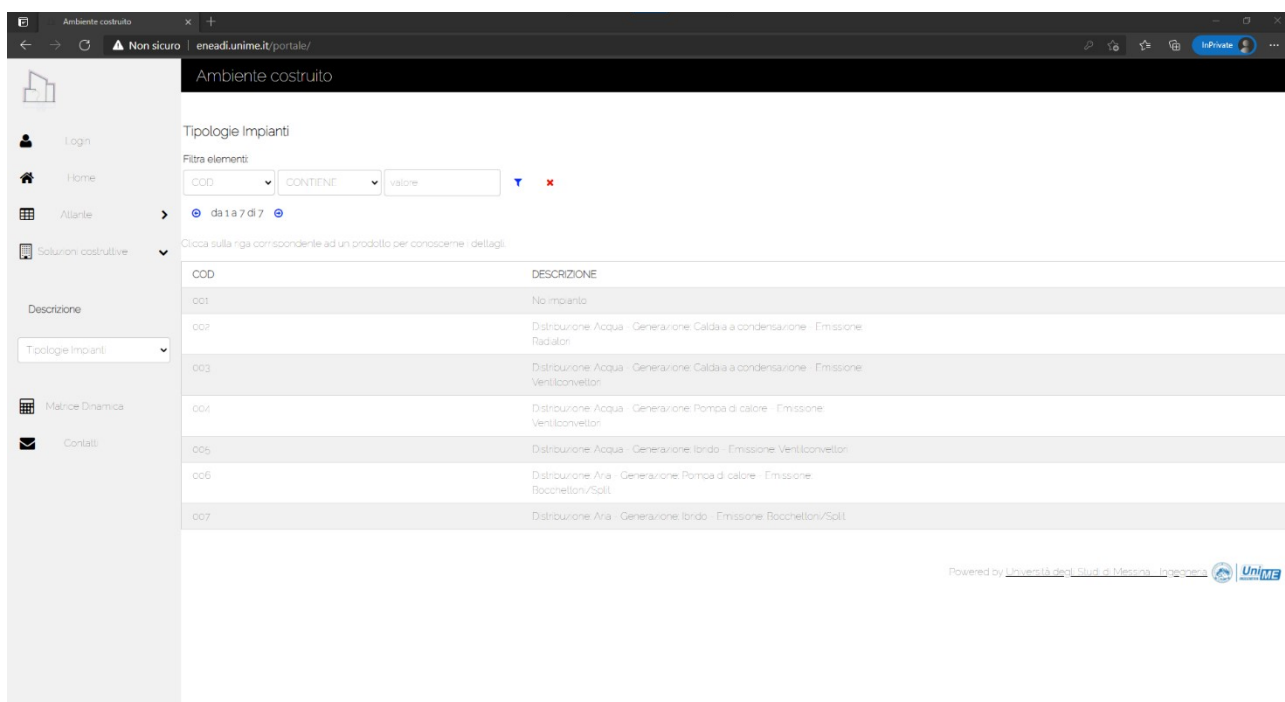


Figura 58: Visualizzazione del contenuto del portale relativa a “Soluzioni Costruttive” - “Tipologie Impianti”

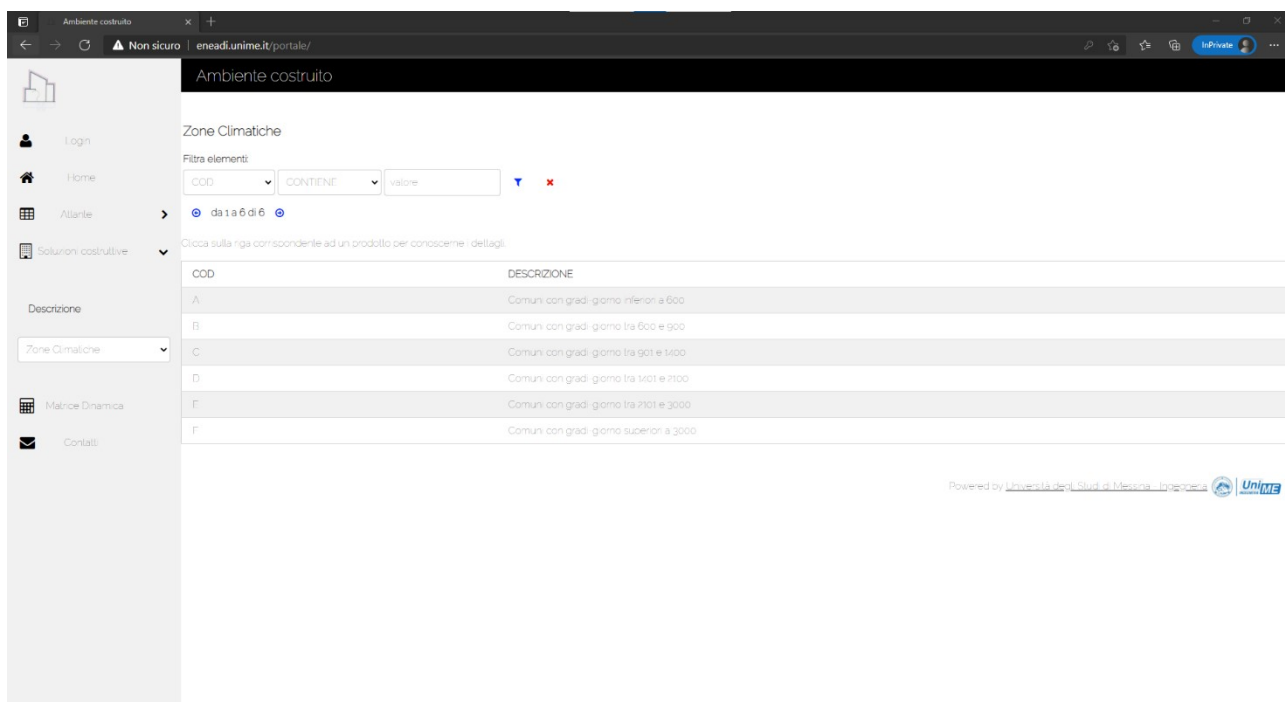


Figura 59: Visualizzazione del contenuto del portale relativa a “Soluzioni Costruttive” - “Zone Climatiche”

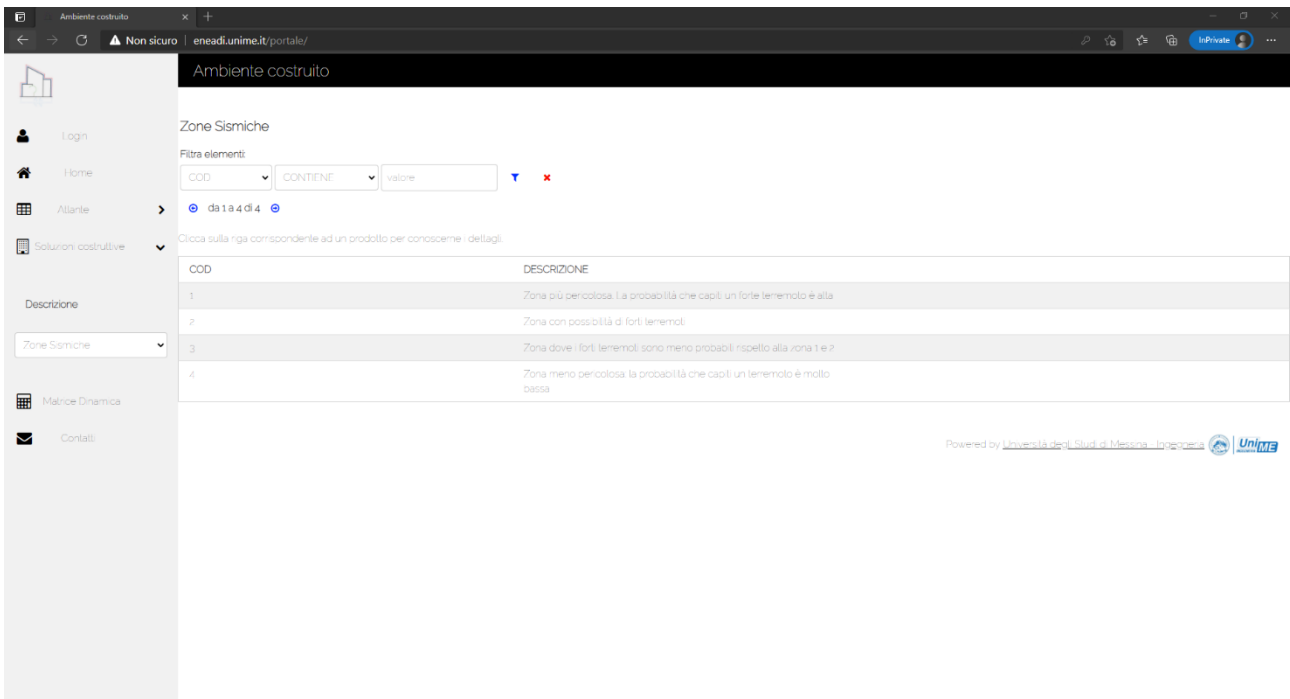


Figura 60: Visualizzazione del contenuto del portale relativa a “Soluzioni Costruttive” - “Zone Sismiche”

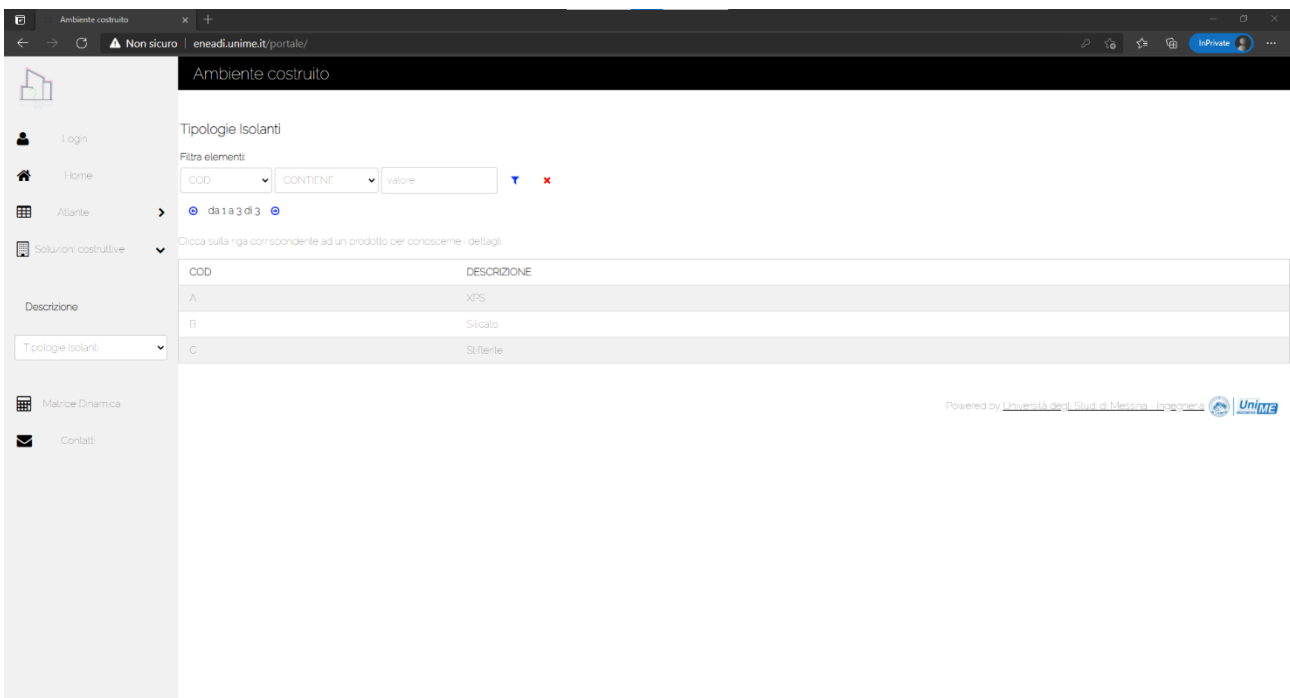


Figura 61: Visualizzazione del contenuto del portale relativa a “Soluzioni Costruttive” - “Tipologie Isolanti”

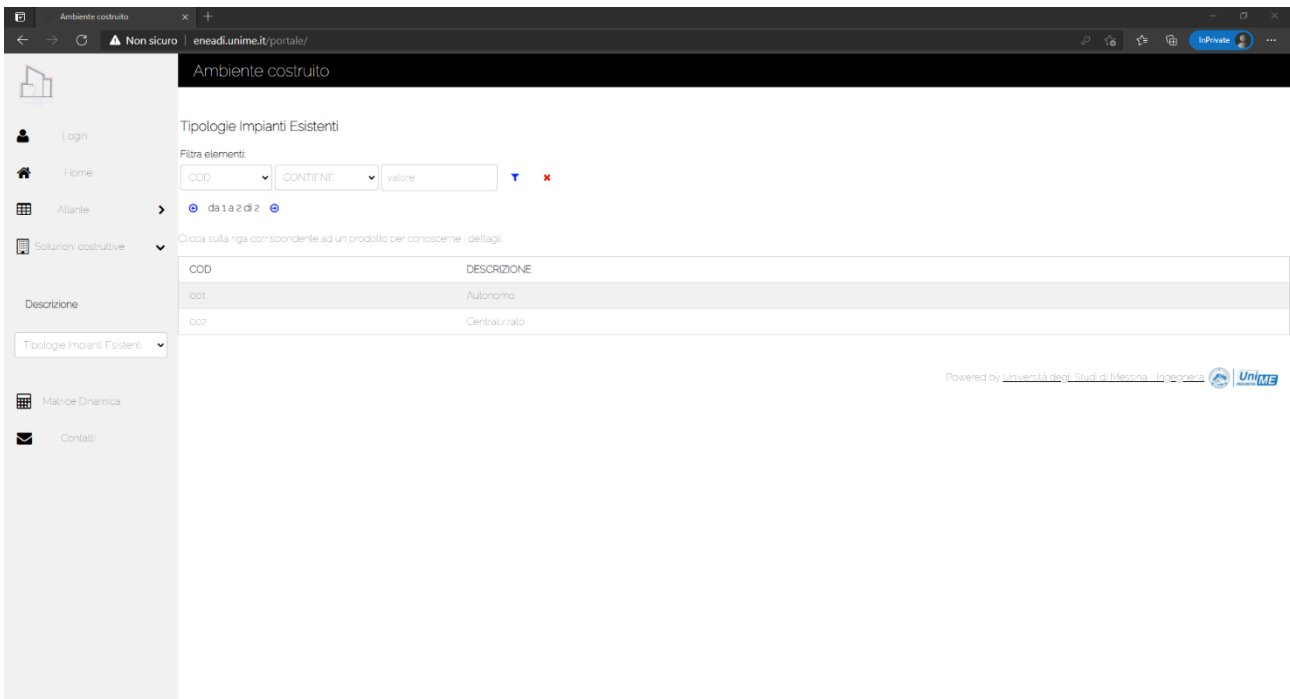


Figura 62: Visualizzazione del contenuto del portale relativa a “Soluzioni Costruttive” - “Tipologie Impianti Esistenti”

La Figura 63 illustra il contenuto del portale nel caso della visualizzazione delle “Soluzioni Costruttive Tipo”.

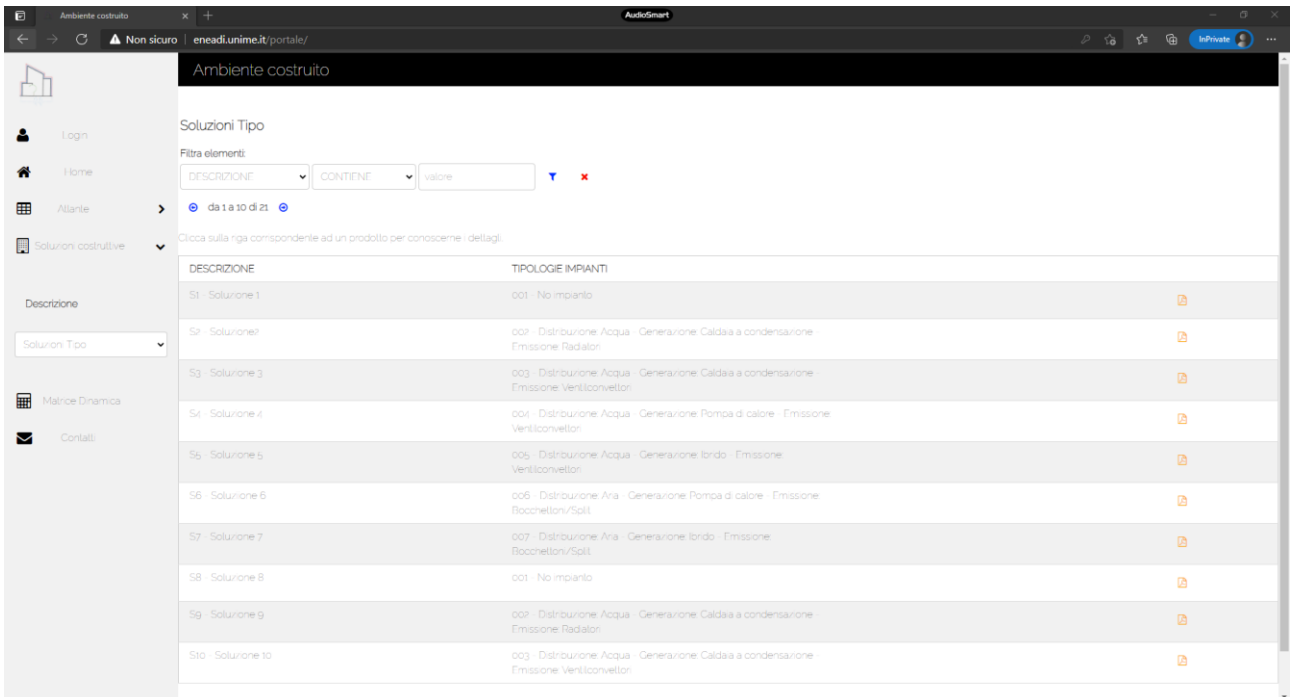


Figura 63: Visualizzazione del contenuto del portale relativa a “Soluzioni Costruttive” - “Soluzioni Tipo”

Differentemente dalle tabelle precedenti in questo caso sono presenti diversi campi il cui contenuto può essere visualizzato facendo click sulle rispettive righe e, inoltre, è presente l’icona per visualizzare il file pdf della scheda tecnica associata ad ogni soluzione.

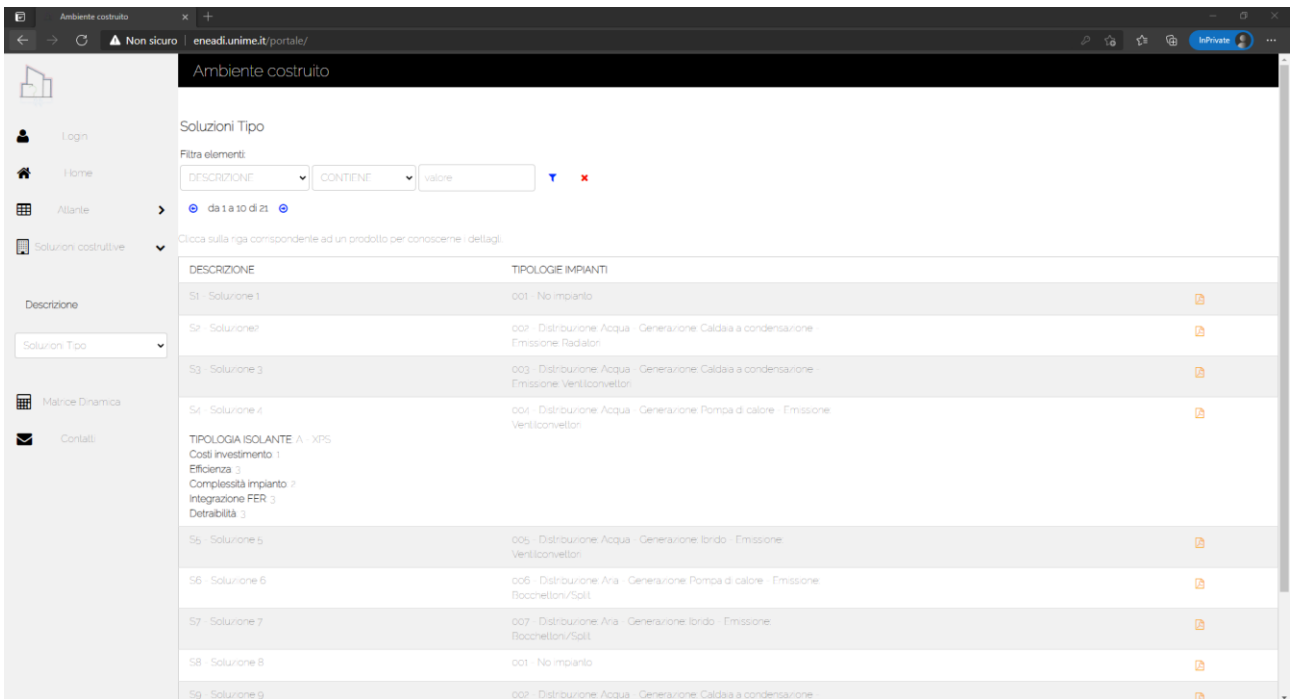


Figura 64: Visualizzazione del contenuto del portale relativa a “Soluzioni Costruttive” - “Soluzioni Tipo” con ingrandita una specifica riga

Nella Figura 64 è illustrato un esempio di visualizzazione delle “soluzioni costruttive tipo” con ingrandita la riga relativa alla soluzione “S4”. L’utente potrà così visualizzare i valori associati a tutti gli altri parametri di interesse (oltre ai campi “Codice”, “Descrizione” e “Tipologie Impianti” già visibili nella visualizzazione compatta) quali “Tipologia Isolante”, “Costi Investimento”, Efficienza”, “Complessità impianto”, “Integrazione FER”, “Detraibilità”.

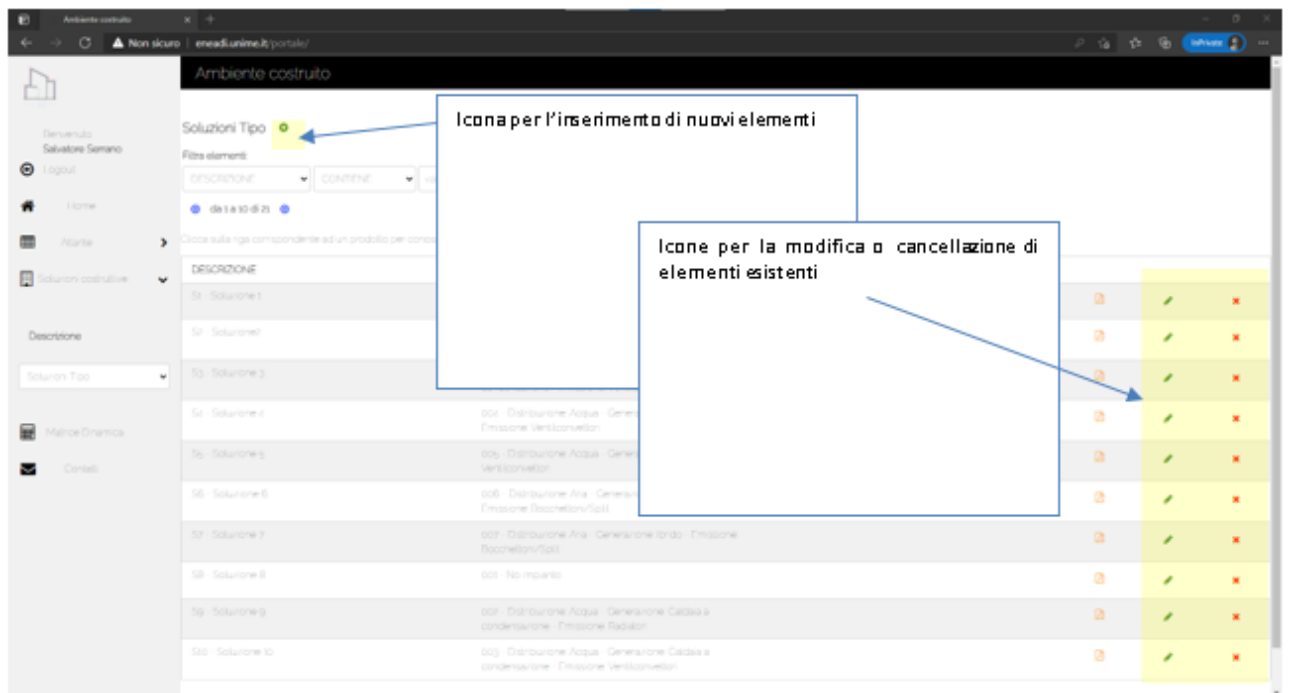


Figura 65: Visualizzazione del contenuto del portale relativa a “Soluzioni Costruttive” - “Soluzioni Tipo” per un utente con credenziali di amministratore

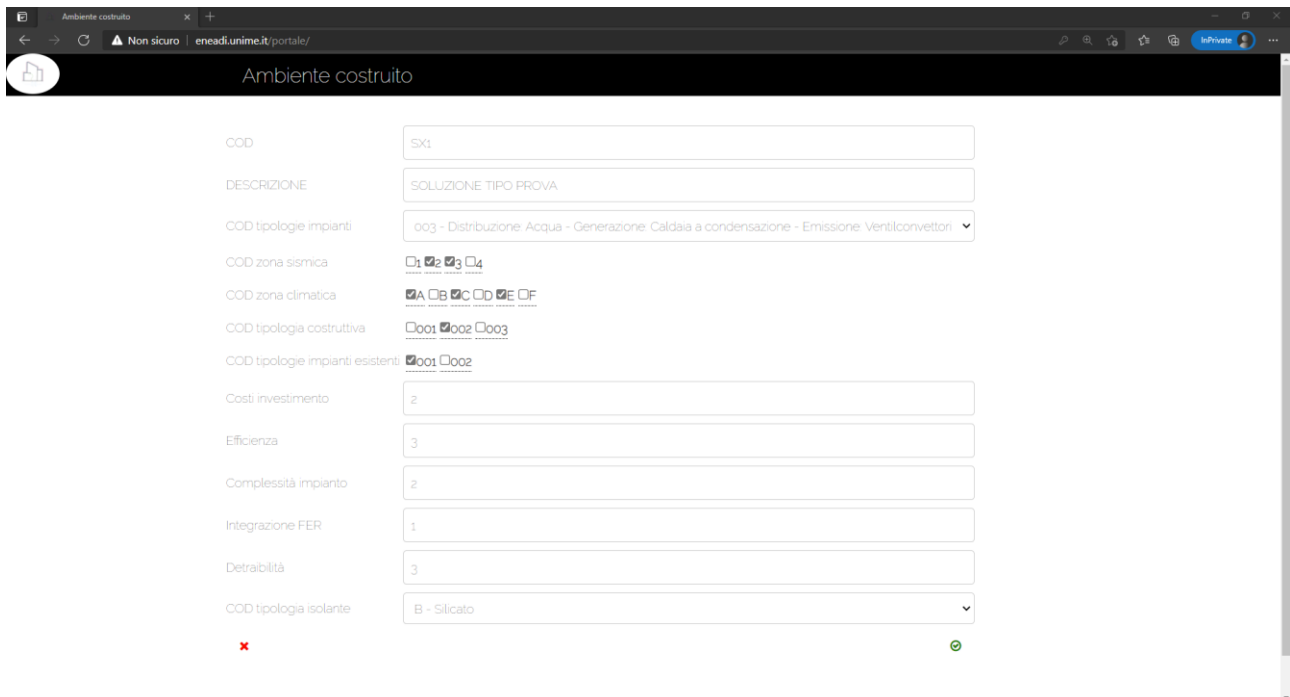
Come nel caso dell'Atlante, gli utenti amministratori avranno una visualizzazione che permette di inserire nuovi elementi per ciascuna categoria di soluzione costruttiva e anche di modificare e/o cancellare elementi già inseriti.

Effettuato il login, un amministratore accederà a una visualizzazione in cui saranno presenti le icone aggiuntive che permettono tali operazioni come illustrato in Figura 65.

Ci soffermeremo adesso sulla visualizzazione di inserimento/modifica relativa alle "soluzioni costruttive tipo" perché attraverso la stessa un utente amministratore potrà modificare le regole che permettono l'implementazione delle scelte della stessa attraverso la matrice dinamica. Come si può osservare in Figura 66, l'utente avrà la possibilità di inserire i campi "COD" relativo al codice della soluzione (che deve essere univoco) e "Descrizione" attraverso gli appositi input-box; sono poi presenti due elementi di tipo drop-down list attraverso i quali potrà selezionare la tipologia di impianto e il tipo di isolante della soluzione.

Sono inoltre presenti quattro diversi gruppi di check-box che permetteranno di descrivere le opzioni di selezione della "soluzione costruttiva tipo" in oggetto. Il primo gruppo riguarda le zone sismiche, il secondo gruppo le zone climatiche, il terzo gruppo la tipologia costruttiva e il quarto gruppo la tipologia di impianto esistente. Ciascun gruppo riporterà un numero di check-box corrispondenti alle possibili opzioni specificate nelle rispettive tabelle delle soluzioni costruttive che l'amministratore potrà ovviamente variare a suo piacimento. Se la "soluzione costruttiva tipo" inserita è selezionabile per una determinata opzione allora il corrispondente check-box dovrà essere marcato. Nell'esempio di prova illustrato in Figura 66 (che teniamo a sottolineare è totalmente inventato), la soluzione tipo inserita con il codice "SX1" prevede che sia selezionabile solo nel caso in cui la "zona sismica" è la "2-Zona con possibilità di forti terremoti" o la "3-Zona dove i forti terremoti sono meno probabili rispetto alla zona 1 e 2", la zona climatica è la "A-Comuni con gradi-giorno inferiori a 600", la "C-Comuni con gradi-giorno tra 901 e 1400" o la "E-Comuni con gradi-giorno tra 2101 e 3000", la tipologia costruttiva è esclusivamente la "002-CALCESTRUZZO" e la tipologia di impianto esistente è "002-Centralizzato". Nell'applicazione della matrice dinamica solo quando l'utente inserirà dei valori che soddisfano tutti i requisiti, la "soluzione costruttiva tipo" sarà scelta per la visualizzazione.

Sono presenti, inoltre, gli altri cinque campi attraverso i quali l'amministratore può inserire/modificare gli indici prestazionali relativi alla "soluzione costruttiva tipo": "Costi Investimento", "Efficienza", "Complessità impianto", "Integrazione FER", "Detraibilità". All'interno di questi campi andranno inseriti dei valori numerici prestazionali. La visualizzazione delle "soluzioni costruttive tipo" selezionate dalla matrice dinamica sarà effettuata ordinando le stesse secondo una funzione di tali parametri. Allo stato attuale la funzione utilizzata consiste semplicemente nella somma dei valori.



The screenshot shows a web browser window with the URL 'enead.unime.it/portale/'. The page title is 'Ambiente costruito'. The form contains the following fields and options:

- COD:** SX1
- DESCRIZIONE:** SOLUZIONE TIPO PROVA
- COD tipologie impianti:** 003 - Distribuzione: Acqua - Generazione: Caldaia a condensazione - Emissione: Ventilconvettori
- COD zona sismica:** C1, C2, C3, C4 (C2 is selected)
- COD zona climatica:** A, B, C, D, E, F (D is selected)
- COD tipologia costruttiva:** 0001, 0002, 0003 (0002 is selected)
- COD tipologie impianti esistenti:** 0001, 0002 (0002 is selected)
- Costi investimento:** 2
- Efficienza:** 3
- Complessità impianto:** 2
- Integrazione FER:** 1
- Detraibilità:** 3
- COD tipologia isolante:** B - Silicato

At the bottom of the form, there is a red 'x' icon on the left and a green checkmark icon on the right.

Figura 66: Visualizzazione per l’inserimento/modifica di una “Soluzione Tipo” e delle informazioni per implementare le regole della sua selezione attraverso la matrice dinamica

Concluse le operazioni di inserimento/modifica dei vari campi, l’utente potrà decidere se procedere con la memorizzazione facendo click sull’icona verde a forma di spunta presente in basso a destra della pagina o annullarla facendo click sulla icona rossa a forma di croce presente in basso a sinistra della pagina.

5.4 La sezione Matrice Dinamica

Attraverso questa sezione l’utente potrà accedere alla pagina che permette di individuare le soluzioni tipo che rispondono ai criteri inseriti dagli amministratori per implementare la matrice dinamica secondo quanto illustrato nel punto precedente.

La pagina presenta la struttura illustrata in Figura 67: nella parte alta saranno presente le drop-down list attraverso le quali l’utente potrà selezionare le condizioni della propria soluzione costruttiva. Attualmente i parametri selezionabili riguardano l’impostazione della “tipologia costruttiva”, della “tipologia di impianto esistente”, della “zona climatica” e della “zona sismica”. Effettuate tutte le selezioni, facendo click sull’icona verde a forma di lente di ingrandimento si visualizzeranno in una tabella che apparirà proprio sotto tale icona tutte le “soluzioni tipo” che rispondono ai criteri impostati ordinate secondo l’indice prestazionale. L’utente potrà poi ingrandire ciascuna riga per visualizzare i dettagli di una specifica soluzione proposta come illustrato in Figura 68. Inoltre, facendo click sull’icona di colore arancio rappresentante il simbolo dei file di tipo pdf potrà aprire, su una nuova finestra del browser, la scheda associata alla soluzione contenente tutte le informazioni tecniche (Figura 69). Utilizzando i comandi messi a disposizione dal proprio browser potrà quindi decidere di archiviare la scheda sul proprio sistema come file oppure stamparla.

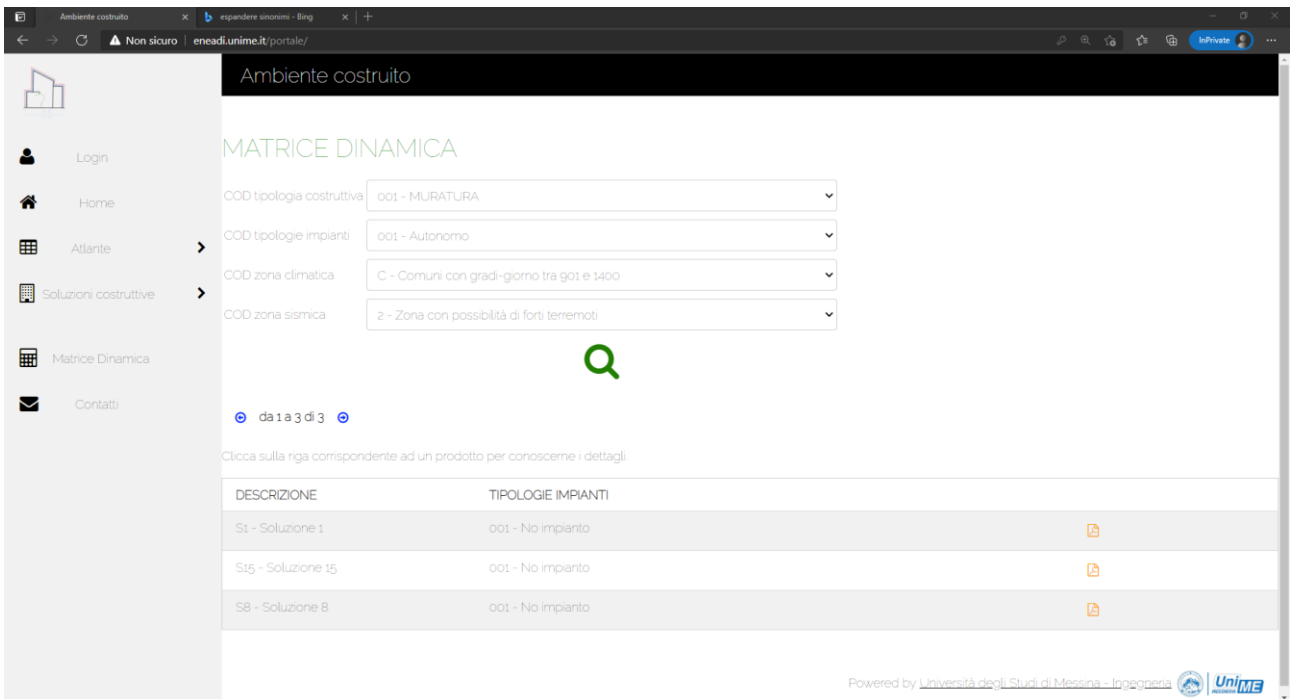


Figura 67: La pagina della Matrice Dinamica

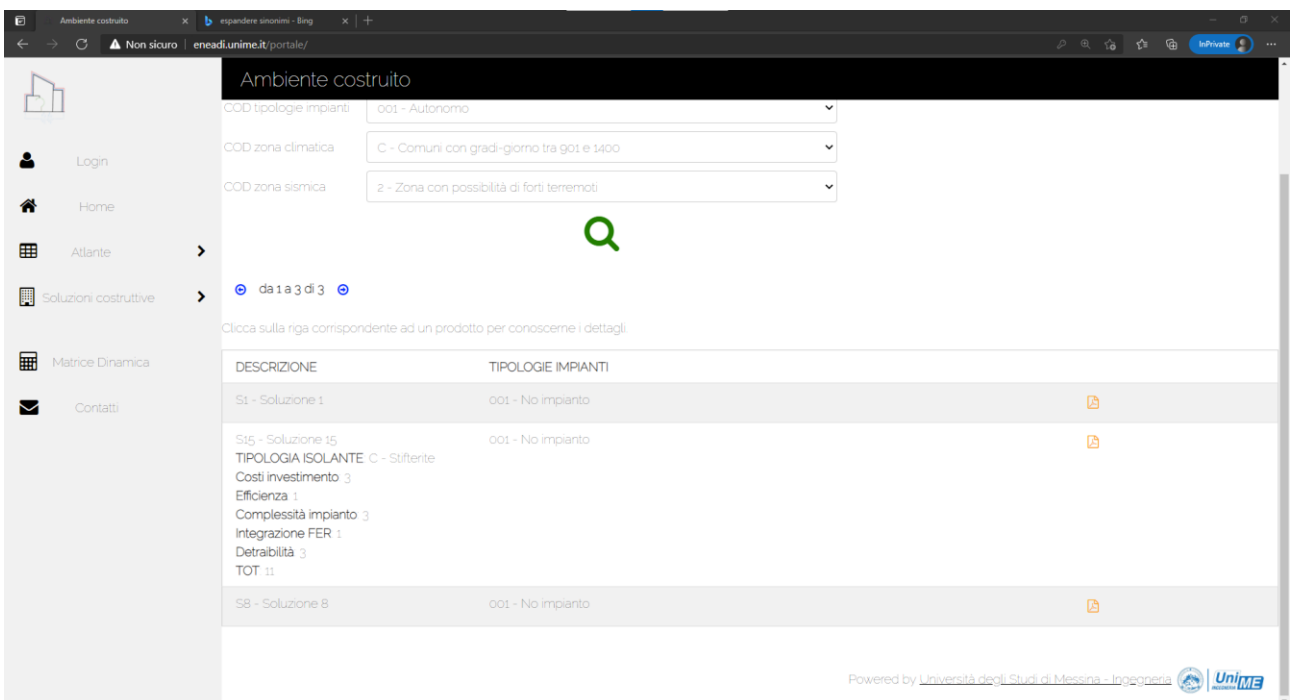
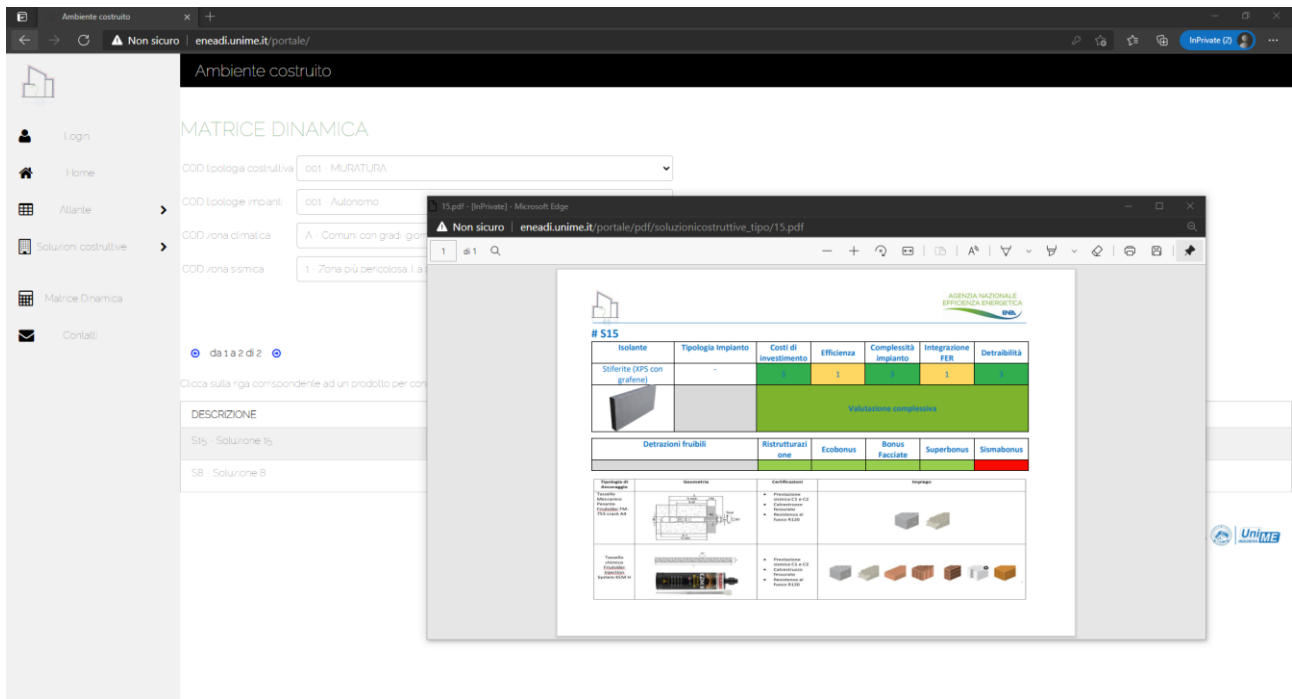


Figura 68: La pagina della Matrice Dinamica con ingrandita la riga relativa ad una specifica soluzione proposta



The screenshot shows a web application interface for 'Ambiente costruito'. The main area displays a 'MATRICE DINAMICA' with filters for construction type (MURATURA), plant type (Autonoma), climate zone (Comuni con gradi giorno), and seismic zone (Zona più pericolosa). A sidebar on the left contains navigation options like 'Login', 'Home', 'Atlante', 'Soluzioni costruttive', 'Matrice Dinamica', and 'Contatti'. A modal window is open, displaying a detailed view of a construction solution type (# S15) for 'Soliferite (XPS con grafene)'. This view includes a table with performance metrics and a section for incentives.

Isolante	Tipologia impianto	Costi di investimento	Efficienza	Complessità impianto	Integrazione FER	Detrattibilità
Soliferite (XPS con grafene)	-	-	1	1	1	1

Detrazioni fruibili	Ristrutturazioni	Ecobonus	Bonus Facciate	Superbonus	Sismabonus
10%	36%	36%	36%	36%	36%

The detailed view also includes sections for 'Tipologia di intervento', 'Caratteristiche tecniche', 'Certificazioni', and 'Impianti' with corresponding diagrams and images.

Figura 69: Apertura di una scheda relativa ad una “Soluzione costruttiva tipo” relativa ad una specifica soluzione proposta

6 Procedura di installazione

UniME fornisce l'insieme dei file necessari per l'installazione del portale. Il set si compone di due file:

- il file compresso `my_eneadi_portale_2021.tar.gz` che contiene tutti i file HTML, JavaScript e PHP che compongono il sito web;
- il file `my_eneadi_portale_2021.sql` che contiene tutti i dati relativi alle tabelle da importare nel database MySQL.

La seguente procedura di installazione fa riferimento ad un server con distribuzione Linux-Ubuntu (è stata utilizzata la versione 20.04.2 LTS), è necessario avere i privilegi di amministratore.

6.1 Installare Apache

Prima di installare il server Apache è necessario verificare che lo stesso non sia già installato sulla macchina in uso. Aprire un terminale e digitare il comando:

```
apache2 -v
```

se la versione ritornata è superiore alla 2.4, Apache è installato ed è possibile passare al paragrafo successivo. Se invece Apache non fosse installato, o se fosse presente una versione troppo obsoleta, sarà necessario re-installarlo. Digitare nel terminale i comandi:

```
sudo apt update  
sudo apt upgrade
```

per aggiornare i repository del server. Successivamente digitare:

```
sudo apt install apache2
```

per installare l'ultima versione del server apache disponibile. Verificare che il servizio sia correttamente funzionante attraverso il comando

```
sudo systemctl status apache2
```

l'output dovrebbe mostrare che il servizio sia attivo e abilitato.

6.2 Installare PHP

Prima di installare PHP è necessario verificare che lo stesso non sia già installato sulla macchina in uso. Aprire un terminale e digitare il comando

```
php --version
```

se la versione ritornata è superiore alla 7.4, PHP è installato ed è possibile passare al paragrafo successivo. Se invece PHP non fosse installato, o se fosse presente una versione troppo obsoleta, sarà necessario re-installarlo. Digitare nel terminale i comandi:

```
sudo apt update  
sudo apt upgrade
```

per aggiornare i repository del server. Successivamente digitare

```
sudo apt install php7.4 php7.4-mysql php-common php7.4-cli php7.4-  
json php7.4-common php7.4-opcache libapache2-mod-php7.4
```

per installare la versione 7.4 di PHP disponibile. Per verificare che PHP sia installato correttamente, riavviare il server apache con il comando

```
sudo systemctl restart apache2
```

quindi creare un file in formato php che visualizzi le informazioni di versione utilizzando, ad esempio, il comando

```
echo '<?php phpinfo(); ?>' | sudo tee -a /var/www/html/phpinfo.php > /dev/null
```

aprire quindi un browser e caricare la pagina creata (anche da un altro dispositivo) all'indirizzo del vostro server

```
http://<INDIRIZZO IP DEL SERVER>/phpinfo.php
```

Se l'installazione è andata a buon fine sarà visualizzata una pagina con la versione di PHP 7.4.3 e l'elenco di tutte le opzioni disponibili di PHP.

A questo punto, se volete, potete rimuovere la pagina PHP create per il test di funzionamento con il comando

```
sudo rm /var/www/html/phpinfo.php
```

6.3 Installare MySQL

Prima di installare *MySQL* è necessario verificare che lo stesso non sia già installato sulla macchina in uso. Aprire un terminale e digitare il comando

```
mysql --version
```

Il sistema mostrerà la versione del server MySQL installata. Se il numero di versione è superiore a 8.0, è possibile passare direttamente al paragrafo successivo. Al contrario, qualora la versione di MySQL fosse obsoleta oppure il server MySQL non fosse installato, si deve procedere ad una nuova installazione. Digitare nel terminale i comandi:

```
sudo apt update  
sudo apt upgrade
```

per aggiornare i repository del server. Successivamente digitare:

```
sudo apt-get install mysql-server
```

Una maniera alternativa per installare i tre servizi Apache-PHP-MySql quando si è sicuri che non sono presenti nel server è quella di installare prima il pacchetto *tasksel*:

```
sudo apt install tasksel
```

e quindi installare tutti i servizi LAMP (acronimo che identifica un server Linux in cui girano i servizi Apache, MySql e PHP/Perl/Python) attraverso il comando

```
sudo tasksel install lamp-server
```

una procedura automatica effettuerà l'installazione e la configurazione di Apache2, MySql e PHP.

Scelta una delle due modalità, comunque, al termine dell'installazione è raccomandato mettere in sicurezza l'installazione di MySQL lanciando uno script che viene installato assieme alla sua installazione. Lo script permetterà di rimuovere eventuali impostazioni non sicure di default e di fare in modo che l'accesso al database sia possibile solo immettendo delle credenziali valide. Per lanciare lo script, occorre digitare il comando

```
sudo mysql_secure_installation
```

Vi sarà richiesto se desiderate abilitare il plugin che verifica la robustezza delle password inserite, vi raccomandiamo di abilitare questa opzione. Potrete a questo punto scegliere il livello di verifica di robustezza della password che desiderate selezionando tre possibili valori (BASSO-MEDIO-ALTO). A questo punto potrete immettere la vostra *password di amministrazione di MySQL* e quindi impostare una serie di ulteriori parametri (potete scegliere le opzioni di default).

Se l'installazione è andata a buon fine, digitando il comando

```
sudo mysql -u root -p
```

potrete accedere al prompt del servizio una volta digitata la password impostata. Per uscire dal prompt si può usare il comando *exit*.

6.4 Creare e importare il database Ambiente Costruito

Per importare il database "ambiente_costruito" è necessario seguire i seguenti passi. Bisogna intanto accedere al prompt di MySQL utilizzando il comando:

```
sudo mysql -u root -p
```

Digitata la password precedentemente impostata in fase di installazione, è necessario digitare ed eseguire i seguenti comandi dal prompt di MySQL:

```
CREATE USER '<nomeutente>'@'localhost' IDENTIFIED BY '<passwordutente>';

CREATE DATABASE ambiente_costruito;

GRANT ALL PRIVILEGES ON ambiente_costruito.* TO
'<nomeutente>'@'localhost';

FLUSH PRIVILEGES;

exit
```

dove *<nomeutente>* e *<passwordutente>* possono essere scelti opportunamente ed arbitrariamente. I comandi precedenti creeranno nel server un account per un nuovo utente MySQL con username *<nomeutente>* e password *<passwordutente>*. Inoltre, creano il database *ambiente_costruito* ed assegnano all'utente *<nomeutente>* tutti i privilegi su questo database.

Effettuata tale operazione, si può provvedere ad importare il contenuto fornito da UniME del database *ambiente_costruito* nel file *my_eneadi_portale_2021.sql* attraverso il comando:

```
sudo mysql -u<nomeutente> -p -Dambiente_costruito <
my_eneadi_portale_2021.sql
```

Immettere la password SQL scelta per l'utente <nomeutente> ed attendere che l'operazione di importazione sia completata.

6.5 Installare *phpmyadmin*

È raccomandata anche l'installazione di *phpMyAdmin* per interagire in modalità grafica con il database MySQL. Se non dovesse essere installato sul server, è possibile farlo, digitando i comandi

```
sudo apt update  
sudo apt upgrade
```

per aggiornare i repository del server. Successivamente digitare:

```
sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl
```

Il sistema porrà delle domande. Il server web da riconfigurare è *Apache2*. Inoltre, rispondere NO quando il sistema chiederà se configurare il database tramite *dbconfig-common*.

Terminata l'installazione, *phpMyAdmin* deve essere attivato in modo da funzionare all'interno del server *Apache*. Digitare nel terminale

```
sudo phpenmod mbstring
```

Successivamente, provvedete ad aprire il file di configurazione di *Apache* con un *text-editor* ed aggiungete al termine del file la riga:

```
Include /etc/phpmyadmin/apache.conf
```

L'ultimo passo consiste nel chiedere al sistema di riavviare il server Apache utilizzando il comando:

```
sudo systemctl restart apache2
```

Se l'installazione è andata a buon fine, l'interfaccia web di *phpMyAdmin* sarà accessibile all'indirizzo <http://<INIDIRIZZO IP DEL SERVER>/phpmyadmin>. Chiedendo di aprire ad un browser tale indirizzo sarà visibile una schermata come quella visualizzata in Figura 70.



Figura 70: Schermata di accesso di PhpMyAdmin

Per accedere è possibile immettere username e password dell'utente root di MySQL o quelli impostati per gestire il database *ambiente_costruito*.

6.6 Installazione dei file del portale e creazione dell'Host Virtuale

I file del portale sono rilasciati da UniME per mezzo di un file compresso (**my_eneadi_portale_2021.tar.gz**). Tali file devono essere decompressi all'interno della cartella predefinita di Apache: */var/www/html*. L'operazione può essere eseguita da un terminale, copiando tale file nella cartella predefinita e poi eseguendo il comando:

```
sudo tar -xvfz my_eneadi_portale_2021.tar.gz
```

A questo punto è conveniente configurare un *Host Virtuale* per fare in modo che il server possa gestire più di un dominio. All'interno della cartella */etc/apache2/sites-available/* creare un file di nome *<nomedominio>.conf* (dove *<nomedominio>* è il nome del dominio che si deve assegnare al portale).

Copiare e incollare le seguenti righe all'interno del file:

```
<VirtualHost *:80>
  ServerAdmin webmaster@localhost
  ServerName <nomedominio>/portale

  DocumentRoot /var/www/html/portale

  <Directory /var/www/html/portale>
    Options FollowSymLinks
    AllowOverride All
    Require all granted
  </Directory>

  LogLevel warn
  ErrorLog ${APACHE_LOG_DIR}/error.log
  CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

e riavviare il server Apache con il comando

```
sudo service apache2 restart
```

L'ultimo step dell'installazione consiste nel modificare il file *myquery.php* del portale che si trova nella cartella *service*, immettendo il nome utente e la password scelti per l'utente SQL. Attraverso un editor di testo modificare i campi corrispondenti che si trovano nelle prime righe del file come mostrato in Figura 71

```
<?php
$dbhost = 'localhost';
$dbuser = '<nomeutente>';
$dbpass = '<passwordutente>';
$dbname = 'ambiente_costruito';

...
...
...

```

Figura 71: intestazione del file *myquery.php* dove inserire le credenziali di accesso a MySQL

Se tutto è andato a buon fine, sarà possibile accedere al portale usando un qualunque browser, collegandosi all'indirizzo web *http:<nomedominio>/portale*.

7 Riferimenti bibliografici

- [1] S. Krug, Don't make me think, revisited: a common sense approach to Web usability (Third ed.), San Francisco, California: New Riders, Peachpit, Pearson Education, 2014.
- [2] T. Berners-Lee e D. Connolly, «Hypertext Markup Language (HTML) Internet Draft,» IIR Working Group, June 1993. [Online]. Available: <https://www.w3.org/MarkUp/draft-ietf-iiir-html-01.txt>.
- [3] Ecma International, «ECMAScript® 2020 language specification,» Ecma International, [Online]. Available: <https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>.
- [4] W3Schools , «W3.CSS Tutorial,» W3Schools , [Online]. Available: <https://www.w3schools.com/w3css/default.asp>.
- [5] PHP Group, «PHP,» [Online]. Available: <https://www.php.net/>.
- [6] Ecma International, «ECMA-404 The JSON data interchange syntax,» Ecma International, [Online]. Available: <https://www.ecma-international.org/publications-and-standards/standards/ecma-404/>.
- [7] Oracle, «MySQL The world's most popular open source database,» Oracle, [Online]. Available: <https://www.mysql.com/>.
- [8] [Online]. Available: <https://github.com/h5p/font-awesome>.
- [9] [Online]. Available: <https://github.com/yougov/raleway-webfont> .
- [10] «GNU Bash,» [Online]. Available: <https://www.gnu.org/software/bash/>.
- [11] A. V. Aho, B. W. Kernighan e P. J. We, The AWK Programming Language, Addison-Wesley, 1988.
- [12] S. Steward, «PDFtk the pdf toolkit,» PDF Labs, [Online]. Available: <https://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/>.
- [13] «XpdfReader - pdftotext,» Glyph & Cog, LLC, [Online]. Available: <https://www.xpdfreader.com/pdftotext-man.html>.



Salvatore Serrano è nato a Catania (Italia) nel 1972. Ha conseguito rispettivamente la laurea quinquennale in Ingegneria Informatica nel 1999 e il dottorato di ricerca in Ingegneria Informatica e delle Telecomunicazioni nel 2003 presso l'Università degli Studi di Catania. Dal 2005 al 2007 ha lavorato come assegnista di ricerca presso il Dipartimento di Ingegneria Informatica e delle Telecomunicazioni dell'Università degli Studi di Catania occupandosi di riconoscimento dello stato emotivo degli individui attraverso l'analisi del segnale vocale inserito in un'attività di ricerca supportata da Telecom Italia Mobile (TIM). Attualmente è ricercatore a tempo indeterminato e professore aggregato del settore telecomunicazioni presso il Dipartimento di Ingegneria dell'Università degli Studi di Messina dove è anche responsabile del laboratorio "Telecomunicazioni". L'attività di ricerca riguarda l'area dell'elaborazione del segnale (codifica e riconoscimento della voce e dei segnali audio, elaborazione di segnali biomedicali, elaborazione del segnale per le telecomunicazioni) e le reti di telecomunicazioni (wireless mesh network, voice over IP e wireless sensor network, cognitive radio). È autore di oltre quaranta articoli scientifici apparsi su riviste internazionali o presentati a conferenze internazionali, revisore di diverse riviste internazionali e fa parte dell'Editorial Board della rivista International Journal of Distributed Sensor Networks (IJDSN).



Filippo Battaglia è nato a Messina nel 1976. Consegue la laurea quinquennale in Ingegneria Elettronica presso l'Università degli Studi di Messina nel 2008 con una tesi dal titolo "Progettazione e sviluppo di un SDK portabile per sistemi embedded, finalizzato alla realizzazione di software di visione artificiale, con applicazioni nel settore dell'assistenza ai non vedenti". Consegue il dottorato di ricerca in Ingegneria dell'Informazione presso l'Università degli Studi Mediterranea di Reggio Calabria nel 2012 con una tesi dal titolo "Autenticazione biometrica basata sull'utilizzo di tecnologia RFID e di algoritmi di tipo seriale per immagini a bassa risoluzione". Tra il 2013 ed il 2021 stato titolare di numerose borse di ricerca e contratti di ricerca presso le Università degli Studi di Messina e di Catania. Gli ambiti in cui ha svolto la propria attività di ricerca riguardano le comunicazioni LPWAN per reti di sensori wireless, l'Internet of Things e le comunicazioni M2M (Machine-to-Machine) e gli algoritmi di elaborazione e riconoscimento delle immagini.



Marco Scarpa ha conseguito la Laurea in Ingegneria Informatica nel 1994 e il Dottorato di Ricerca in Informatica presso l'Università degli Studi di Torino nel febbraio 2000. Attualmente è Professore Ordinario di Sistemi di Elaborazione presso l'Università degli Studi di Messina. È revisore stabile di numerose conferenze internazionali e riviste di informatica come: "IEEE Transactions on Dependable and Secure Computing", "IEEE Transactions on Parallel and Distributed Systems", "International Journal of System and Sciences", "Performance Evaluation", "Journal of Computational Science" e molti altri.

È stato guest editor di un numero speciale di Performance Evaluation (Elsevier); è attualmente membro del comitato editoriale del Journal of Distributed Sensor Network "e Area Editor di EAI Transactions on Cloud Systems. È stato coinvolto in numerosi Comitati di Programma Tecnico di conferenze internazionali. La sua attività di ricerca comprende l'analisi di affidabilità e disponibilità di sistemi distribuiti, reti di sensori wireless, algoritmi per la gestione dell'accesso opportunistico nei sistemi radio cognitivi, algoritmi per la soluzione di reti di Petri stocastiche non markoviane, distribuzioni "phase-type" e tecniche di valutazione delle prestazioni del software specialmente applicate a sistemi distribuiti. Recentemente, ha avviato una collaborazione industriale nel campo degli algoritmi di riconoscimento musicale efficiente. È stato attivamente coinvolto in diversi progetti di ricerca. Coordina gli sviluppi di WebSPN, uno strumento software in grado di gestire reti di Petri stocastiche con tempi di accensione generalmente distribuiti per lo studio delle prestazioni e dell'affidabilità del sistema, e MAGNET, una libreria software C / C ++ che fornisce una facile gestione e implementazione di complessi modelli Markovian Agent based .



Giuseppe Campobello nasce a Messina, Italia, nel 1975. Consegue la Laurea in Ingegneria Elettronica (summa cum laude) e il Dottorato di Ricerca (Ph.D.) in "Tecnologie Avanzate per l'Ingegneria dell'Informazione" presso l'Università degli Studi di Messina, rispettivamente nel 2000 e nel 2004. Dal dicembre del 2006 è Ricercatore Universitario del settore ING-INF/03 (Telecomunicazioni) presso l'Università degli Studi di Messina. Attualmente afferisce al Dipartimento di Ingegneria dell'Università degli Studi di Messina dove ricopre il ruolo di Professore Aggregato del settore Telecomunicazioni e responsabile del laboratorio di Comunicazioni Wireless. L'attività di ricerca, svolta sia in ambito universitario che in collaborazione con aziende ed enti di ricerca, si inquadra principalmente nell'ambito delle reti di telecomunicazioni e dell'elaborazione numerica dei segnali applicata alle telecomunicazioni e ai sistemi biomedicali. In particolare, l'attività di ricerca più recente è focalizzata sulle reti di sensori wireless e sull'Internet of Things. È autore di oltre cinquanta articoli scientifici apparsi su riviste internazionali o presentati a conferenze internazionali ed è revisore di diverse riviste internazionali della IEEE. È inoltre membro del Gruppo Telecomunicazioni e Tecnologie dell'Informazione (GTTI) e del Microwave Engineering Center for Space Applications (MECSA).