



Ricerca di Sistema elettrico

Implementazione di un modulo che, all'interno del portale motori, consenta di effettuare studi e analisi di tipo statistico e l'archiviazione di tutta la documentazione di testing

G. Campobello, A. Segreto



IMPLEMENTAZIONE DI UN MODULO CHE, ALL'INTERNO DEL PORTALE MOTORI, CONSENTA DI EFFETTUARE STUDI E ANALISI DI TIPO STATISTICO E L'ARCHIVIAZIONE DI TUTTA LA DOCUMENTAZIONE DI TESTING"

G. Campobello, A. Segreto
Dipartimento di Ingegneria, Università degli Studi di Messina

Dicembre 2018

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dello Sviluppo Economico - ENEA

Piano Annuale di Realizzazione 2018

Area: "Efficienza energetica e risparmio di energia negli usi finali elettrici e interazione con altri vettori energetici"

Progetto: D.3 "Processi e Macchinari Industriali"

Obiettivo: a.1 "Facility per la sperimentazione e verifica di motori elettrici ad alta efficienza"

Responsabile del Progetto: ENEA

Il presente documento descrive le attività di ricerca svolte all'interno dell'Accordo di collaborazione " *Implementazione di un modulo che, all'interno del portale motori, consenta di effettuare studi e analisi di tipo statistico e l'archiviazione di tutta la documentazione di testing*"

Responsabile scientifico ENEA: Ing. Maria-Anna Segreto

Responsabile scientifico Dipartimento di Ingegneria: Prof. Ing. Giuseppe Campobello

Indice

SOMMARIO.....	1
1 FORM PER LA GESTIONE DEI FILE E MECCANISMI DI SICUREZZA.....	2
2 GRAFICI E FUNZIONI STATISTICHE.....	5
3 DESCRIZIONE DEI CODICI SORGENTE.....	8
APPENDICE.....	18

Sommario

Il presente report tecnico descrive le attività svolte nell'ambito dell'accordo di collaborazione tra ENEA e il Dipartimento di Ingegneria dell'Università di Messina per il PAR 2018 e che hanno avuto come obiettivo la definizione e lo sviluppo di un modulo per l'analisi statistica e l'archiviazione di dati inerenti motori elettrici per ambienti industriali.

In particolare le attività hanno permesso di estendere con nuove funzionalità il portale web dell'ENEA, disponibile al sito <http://motorielettrici.enea.it>, originariamente realizzato dalla stessa ENEA, in collaborazione con il Gruppo di Macchine Elettriche Rotanti di ANIE Energia, con l'obiettivo di raccogliere sul sito prodotti conformi al Regolamento della Commissione (CE) 640/2009 su efficienza energetica e classi di efficienza, oltre di segnalare le future evoluzioni normative sul tema.

Più precisamente, le attività oggetto del presente report hanno riguardato:

- l'estensione del portale web con la definizione e realizzazione di nuovi form atti a permettere l'upload e il download di tutta la documentazione di testing in maniera protetta. In particolare, le operazioni di upload e download sono possibili solo agli operatori autorizzati e in possesso di apposite credenziali personali. Il modulo realizzato, oltre a fornire gli strumenti per la gestione dei file, permette altresì il controllo di integrità degli stessi mediante checksum MD5.
- la realizzazione e integrazione di nuove funzionalità utili per studi statistici relativi ai motori verificati a seguito dei test sui banchi prova. In particolare, è stato modificato ed esteso il modulo per la gestione dei Check Report precedentemente sviluppato dalla stessa unità nell'ambito del PAR2017. Il nuovo modulo sviluppato fornisce una descrizione più dettagliata dello stato delle verifiche e permette di ottenere informazioni statistiche sulle verifiche effettuate.

Le funzionalità suddette sono descritte nelle successive Sezioni del presente report.

In particolare:

- nella Sezione 1 verranno descritte le form realizzate per l'upload e il download dei file oltre che i meccanismi di sicurezza introdotti;
- nella Sezione 2 sarà descritto il nuovo modulo per la gestione dei Check Report e le nuove funzionalità statistiche e grafiche sviluppate durante l'attività;
- nella Sezione 3 sarà descritto il codice sorgente dei moduli suddetti;
- infine in Appendice è riportata la procedura di installazione.

1 Form per la gestione dei file e meccanismi di sicurezza

Il portale web ENEA, disponibile al sito <http://motorielettrici.enea.it>, è stato realizzato con l'obiettivo di raccogliere sul sito dati inerenti motori elettrici e prodotti conformi al Regolamento della Commissione (CE) 640/2009 su efficienza energetica e classi di efficienza, oltre che di segnalare le future evoluzioni normative sul tema. Il portale fornisce informazioni testuali sui dati di targa dei motori elettrici, sulle relative classi di efficienza energetica oltre che la possibilità di gestire documenti forniti dalle stesse case costruttrici ("Test Report") o inerenti misure sperimentali e verifiche di funzionamento effettuate dall'ENEA ("Check Report"). La Figura 1 mostra il portale web ENEA, cui sono evidenziate in rosso le funzionalità aggiuntive sviluppate da UniME nell'ambito delle attività.

Per una descrizione più estesa del Portale si rimanda al "Report RdS 003" redatto dalla stessa unità nell'ambito delle attività relative al PAR2017.



Figura 1: Portale Web ENEA.

Scopo principale delle attività nell'ambito del PAR2018 è stato quello di estendere le funzionalità del portale al fine di introdurre meccanismi di sicurezza e permettere l'upload e il download di tutta la documentazione di testing in maniera protetta. In particolare il sistema realizzato permette le operazioni di upload e download solo agli operatori autorizzati e in possesso di apposite credenziali personali.

In Figura 2 è mostrata la form per l'inserimento delle credenziali di accesso.



Figura 2: Form di inserimento delle credenziali di accesso.

Il sistema prevede cinque tipologie di utenti:

- l'utente "amministratore" ("Administrator"), la cui funzione principale è quella di gestire gli account degli utenti e gli indirizzi IP delle macchine autorizzate all'accesso del server;
- l'utente "collaboratore" ("Contributor") che può inserire nuovi motori e i relativi dati di targa;
- l'utente "visitatore" ("Visitor") che può visualizzare i dati esposti dal portale ed effettuare il download dei documenti;
- l'utente "operatore" che, oltre ad inserire i dati di targa di nuovi motori e modificare quelli già esistenti, può effettuare operazioni sui documenti di Test e di Check;
- l'utente "comune" che, in assenza di un account, può solo visualizzare i dati e i documenti esposti dal portale e identificati come "documenti pubblici".

In Figura 3 è illustrata la form realizzata nell'ambito delle attività oggetto del presente report al fine di permettere agli operatori di introdurre i dati relativi alle verifiche effettuate sui motori e gestire l'upload dei documenti connessi.



Figura 3: Form per l'inserimento dei dati di check.

Come è possibile osservare in Figura 3, la form realizzata permette l'inserimento di dati relativi a tensione, corrente, potenza, frequenza e numero di giri oltre che delle temperature di carcassa e dell'aria di raffreddamento.

La form permette altresì di specificare lo stato delle verifiche tramite il menu a tendina identificato dalla label "Stato della verifica" posto al centro della form. Il pulsante "Altri Doc" permette l'upload della documentazione. Un checkbox "Download Pubblico" permette di specificare se i file potranno essere scaricati senza restrizioni anche da utenti comuni o, viceversa, se l'accesso ai file sarà possibile solo agli operatori in possesso di credenziali. Infine, premendo sul pulsante "Procedi", tutti i dati inseriti e i file importati saranno salvati sul server.

I dati e i file caricati potranno successivamente essere visualizzati tramite il portale nella sezione “Checks” associata ad ogni motore (si veda la Figura 1). Sempre dalla sezione “Checks” è possibile effettuare il download di un file spostando il mouse sull'icona del file desiderato e premendo il tasto sinistro del mouse. Premendo invece il tasto destro del mouse su uno dei file apparirà una finestra riportante il nome del file selezionato con accanto un codice alfanumerico, come illustrato nell'esempio di Figura 4.

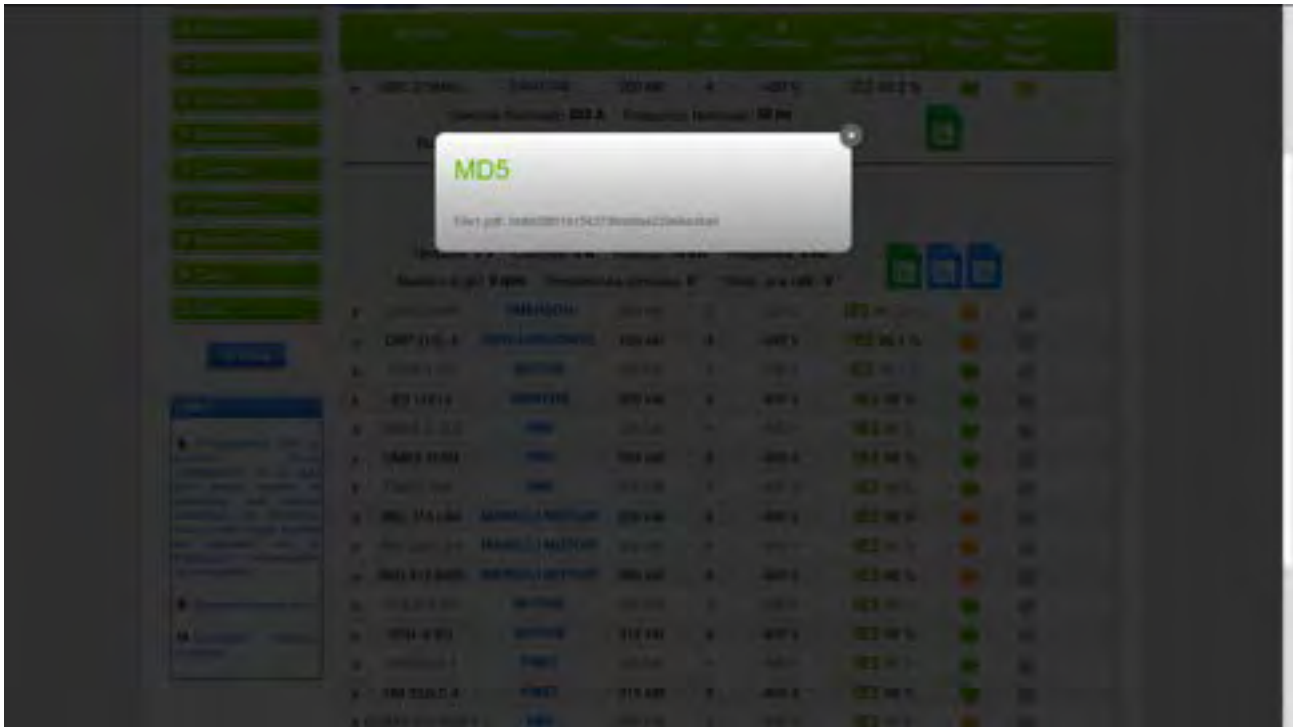


Figura 4: Visualizzazione del checksum MD5 relativo al file di interesse.

Il codice in questione corrisponde alla checksum MD5 del file selezionato. Una checksum MD5 consiste in una stringa di 128 bit, rappresentata da 32 caratteri esadecimali, ottenuta a partire dal file di input mediante una funzione hash standardizzata dalla IETF e descritta nel documento RFC1321¹. Tale funzione è stata progettata affinché la variazione di un singolo bit del file di input comporti una checksum generata completamente differente.

La checksum MD5 può essere efficacemente utilizzata per controlli di integrità dei file.

In particolare, la checksum può essere utilizzata da un utente per verificare che il file non sia stato danneggiato durante il download. Ciò è possibile semplicemente confrontando la checksum riportata dal portale con la checksum ricalcolata con un qualsiasi software compatibile con lo standard MD5. A tal fine sono disponibili diversi tool quali ad esempio *OpenSSL*². OpenSSL è disponibile per sistemi operativi Linux, Windows e MacOS e permette il calcolo della checksum MD5 mediante il comando *md5sum* a cui deve seguire il nome del file di cui si richiede il test di integrità. Ad esempio il comando

```
md5sum TestMotoreAX25.doc
```

effettua il calcolo della checksum sul file *TestMotoreAX25.doc*.

Se la checksum ottenuta mediante il comando suddetto coincide con la checksum riportata dal portale il file è stato scaricato correttamente.

Anche se ad oggi esistono strumenti più sofisticati e sicuri, la checksum MD5 può essere altresì utilizzata per verificare che i file non siano stati intenzionalmente manomessi ovvero che eventuali copie del file siano fedeli all'originale.

Si rimanda alla Sezione 3 del presente documento per una descrizione del codice sorgente sviluppato per la realizzazione delle funzionalità suddette.

¹ “The MD5 Message-Digest Algorithm”, <https://www.ietf.org/rfc/rfc1321.txt>.

² “OpenSSL”, <https://www.openssl.org/>.

2 Grafici e funzioni statistiche

Il portale permette di visualizzare i dati relativi ai motori anche mediante grafici a torta e a barre. Tali grafici forniscono utili informazioni statistiche, quali, ad esempio, il numero e la percentuale dei motori che rientra nelle diverse classi di efficienza (IEX), e possono essere ottenuti semplicemente selezionando l'“icona grafici” riportata in alto nella colonna relativa al campo di interesse (si veda la Figura 1).

Alcune delle funzionalità statistiche e grafiche implementate sono state già descritte nel “Report RdS 003” del PAR2017 redatto dalla stessa unità. A seguito delle attività del PAR2017 sono state però richieste da ENEA alcune modifiche. In questa sezione, ci si limiterà per motivi di brevità a descrivere solo le modifiche più rilevanti.

Una di tali modifiche ha riguardato il campo Check Report.

Nella versione precedente del modulo, selezionando l'icona grafici sulla colonna Check Report, il sistema riportava un grafico con la percentuale di motori che hanno o non hanno superato le verifiche (si veda ad esempio la Figura 5).

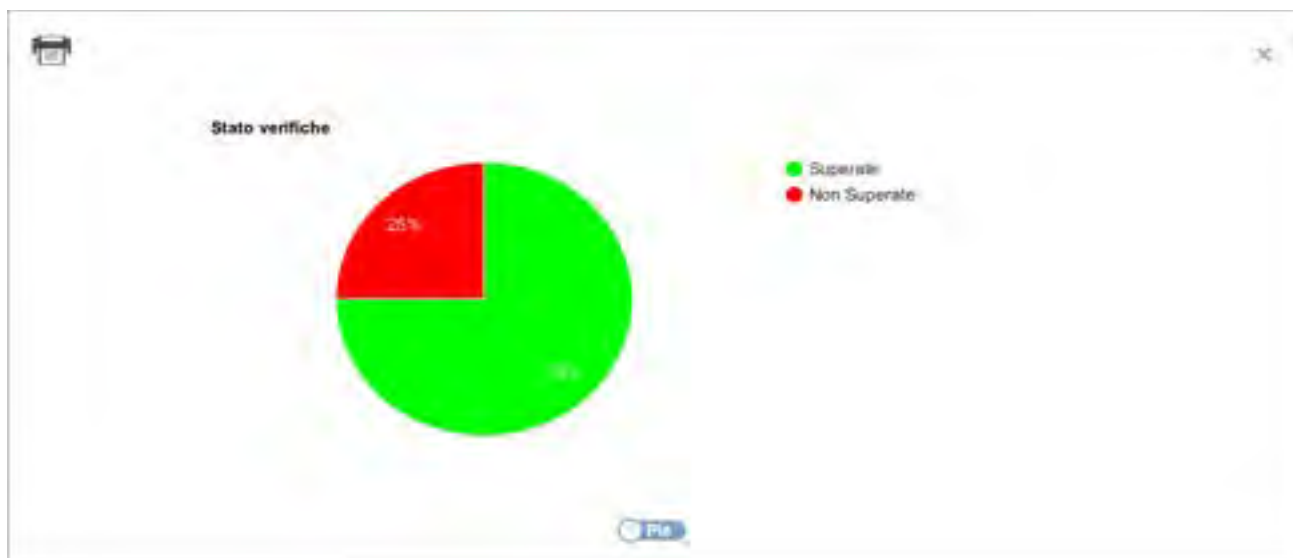


Figura 5: Stato delle verifiche e percentuale di motori che hanno superato le verifiche.

Il nuovo modulo sviluppato fornisce invece una descrizione più dettagliata dello stato delle verifiche e permette di ottenere un maggior numero di informazioni statistiche sulle verifiche effettuate, quali, ad esempio, quanti dei motori presi in carico ad una certa data siano stati ad oggi effettivamente verificati o la percentuale di motori nelle diverse fasi del processo di verifica.

In particolare, il nuovo modulo prevede per il processo di verifica le seguenti fasi:

- Verifica non ancora iniziata;
- Verifica ancora in corso;
- Verifica rimandata (in attesa di ulteriori informazioni da parte del produttore);
- Verifica ultimata (distinguendo tra verifiche ultimate con esito positivo e verifiche ultimate con esito negativo).

Ad ogni fase è associato un colore come riportato in Tabella 2.1.

Descrizione stato di check	Colore
Verifica ancora non iniziata	GRIGIO ●
Verifica ancora in corso	GIALLO ●
Verifica rimandata	ARANCIONE ●
Verifica ultimata con esito positivo	VERDE ●
Verifica ultimata con esito negativo	ROSSO ●

Tabella 2.1: Tabelle delle corrispondenze tra stati e colore associati.

La definizione dei nuovi stati ha richiesto alcune modifiche al database del portale. In particolare sono state modificate la tabella `mot_checks`, mediante la quale sono state specificate le possibili fasi del processo di verifica, e la tabella `mot_colors` che associa i colori alle differenti fasi.

In Figura 6 è riportato un esempio di grafico ottenuto dal nuovo modulo di gestione dei Check Report selezionando l'icona grafici della corrispondente colonna. Come è possibile osservare, il grafico fornisce la percentuale di motori nelle diverse fasi del processo di verifica.

Figura 6: Nuovo grafico relativo allo stato delle verifiche.

Si ritiene utile precisare che, a seguito di specifiche da parte di ENEA, nella colonna Check Report del portale gli stati "Verifica rimandata" e "Verifica in corso" sono entrambe identificate con il colore Giallo.

Un'altra delle modifiche introdotte riguarda la realizzazione di un nuovo modulo atto a fornire informazioni statistiche sul processo di verifica.

Il modulo può essere richiamato selezionando l'icona "orologio" presente in alto nella colonna Check Report. In tal caso apparirà sul portale il form in Figura 7.

Inserendo la data di inizio e di fine del periodo di interesse il modulo genera un grafico a barre che mostra il numero di motori sottoposti a verifica nel periodo in questione e il numero di motori per i quali la verifica risulta definitivamente conclusa o ancora in corso.



Figura 7: Informazioni statistiche sul processo di verifica relative ai motori presi in carico in un intervallo temporale desiderato

Tale modulo può essere efficacemente utilizzato per analizzare la produttività del processo di test o per schedulare i tempi di nuove verifiche.

3 Descrizione dei codici sorgente

In Figura 8, Figura 9 e Figura 10 è riportato parte del codice sorgente sviluppato nell'ambito delle attività oggetto del presente report tecnico.

In particolare, il codice in Figura 8 realizza la form di inserimento riportata in Figura 3 e precedentemente descritta nella Sezione 1

Come è possibile osservare la form è stata realizzata con una serie di comandi echo con codice html. La prima parte del codice, evidenziata in giallo, specifica i campi di input per le diverse grandezze fisiche oggetto di verifica (tensione, corrente, potenza, frequenza, temperatura della carcassa e temperatura dell'aria di raffreddamento). La parte evidenziata in verde implementa la combobox che consente di specificare lo stato della verifica. La parte evidenziata in azzurro gestisce l'upload e la visualizzazione dei file associati al motore.

La visualizzazione, l'inserimento o la cancellazione dei file è resa possibile attraverso la generazione dinamica della pagina web realizzata mediante un JavaScript denominato createUploader, di cui parte del codice è riportato in Figura 9. Si noti che il codice JavaScript è stato realizzato per supportare diverse tipologie di browser.

Lo script mette a disposizione tre metodi denominati loadFile, delFile e render.

Il metodo loadFile gestisce l'upload dei file, delFile permette la cancellazione di uno specifico file, mentre il metodo render determina e visualizza la lista dei file associati al motore.

Lo script interagisce con un servizio php, di cui si riporta parte del codice in Figura 10, il quale, interrogato dalla funzione creatUploader, fornisce la lista dei documenti relativi ad un motore, esegue l'upload dei file e l'eventuale cancellazione. Inoltre, il servizio php interagisce con il database mysql per valorizzare i campi relativi alle grandezze fisiche oggetto di verifica.

Come linguaggio di comunicazione tra i due script è stato utilizzato JSON (JavaScript Object Notation)³, un formato adatto all'interscambio di dati fra applicazioni client/server, basato sul linguaggio JavaScript Standard ECMA-262⁴.

Infine, il meccanismo della checksum MD5 viene realizzato dallo stesso servizio php utilizzando la funzione php md5_file di cui si riporta il prototipo nella Figura 11.

³ "Introducing JSON", <http://json.org/>.

⁴ "Final draft Standard ECMA-262 edition 5.1", March 2011 (Rev. 6) - Ecma 262, ECMA International, Marzo 2011.

```
$motore = $this->get_var('motore');
$checkdata = $this->get_var('checkdata');
echo '<h1>'. $lang["CHECK_MOT_TITLE"].' ('.$motore['modello'].')</h1>';
echo '<form action="'. $this->get_var('domain').'" method="post"
name="form-checkmot" target="_self" id="form-checkmot" data-ajax="'.
$this->get_var('domain').ACT_MOT.'/checkmod" onsubmit="return
checkForm($(this));">';
```

```
    echo '<input type="submit" value="ok" class="hidd" />';
    echo '<input type="hidden" id="id_mod" name="id_mod" value="'.
checkdata['id_motore'].'" />';
```

```
// TENSIONE
echo '<div class="div-input">';
    echo '<label>'. $lang["LIST_VOLT_CHECK"].'</label><br />';
    echo '<input id="check-tensione" name="check-tensione"
value="'. $checkdata['tensione'].'" data-min="0" data-type="9"
class="text_input small" type="text" /> <label>V</label>';
    echo '<div class="err_lbl"></div>';
echo '</div>';
```

```
// CORRENTE
echo '<div class="div-input">';
    echo '<label>'. $lang["LIST_CUR_CHECK"].'</label><br />';
    echo '<input id="check-corrente" name="check-corrente"
value="'. $checkdata['corrente'].'" data-min="0" data-type="9"
class="text_input small" type="text" /> <label>A</label>';
    echo '<div class="err_lbl"></div>';
echo '</div>';
```

```
// POTENZA
echo '<div class="div-input">';
    echo '<label>'. $lang["LIST_POW_CHECK"].'</label><br />';
    echo '<input id="check-potenza" name="check-potenza"
value="'. $checkdata['potenza'].'" data-min="0" data-type="9"
class="text_input small" type="text" /> <label>W</label>';
    echo '<div class="err_lbl"></div>';
echo '</div>';
```

```
// FREQUENZA
echo '<div class="div-input">';
    echo '<label>'. $lang["LIST_FREQ_CHECK"].'</label><br />';
    echo '<input id="check-frequenza" name="check-frequenza"
value="'. $checkdata['frequenza'].'" data-min="0" data-type="9"
class="text_input small" type="text" /> <label>Hz</label>';
    echo '<div class="err_lbl"></div>';
echo '</div>';
```

```
// DIV VUOTO
echo '<div class="div-input">';
echo '</div>';
```

```
// GIRI
echo '<div class="div-input">';
    echo '<label>'. $lang["LIST_RPM_CHECK"].'</label><br />';
```

```

        echo '<input id="check-giri" name="check-giri" value="'.
checkdata['giri'].'" data-min="0" data-type="9" class="text_input small"
type="text" /> <label>rpm</label>';
        echo '<div class="err_lbl"></div>';
echo '</div>';

// TEMPERATURA CARCASSA
echo '<div class="div-input">';
        echo '<label>'. $lang["LIST_TCASE_CHECK"].'</label><br />';
        echo '<input id="check-temp_case" name="check-temp_case"
value="'. $checkdata['temp_case'].'" data-min="0" data-type="9"
class="text_input small" type="text" /> <label>°C</label>';
        echo '<div class="err_lbl"></div>';
echo '</div>';

// DIV VUOTO
echo '<div class="div-input">';
echo '</div>';

// TEMPERATURA DELL'ARIA DI RAFFREDDAMENTO
echo '<div class="div-input">';
        echo '<label>'. $lang["LIST_TCOOLER_CHECK"].'</label><br />';
        echo '<input id="check-temp_cooler" name="check-temp_cooler"
value="'. $checkdata['temp_cooler'].'" data-min="0" data-type="9"
class="text_input small" type="text" /> <label>°C</label>';
        echo '<div class="err_lbl"></div>';
echo '</div>';

// DIV VUOTO
echo '<div class="div-input">';
echo '</div>';

// DIV COMBO STATO VERIFICA
echo '<div class="div-input">';
        echo '<label>'. $lang["LIST_OUTCOME_CHECK"].'</label><br />';
        echo '<select name="check-cod_check" id="check-cod_check"
class="text_input";">';
        foreach($this->get_var('checks') as $id_check)
                if($id_check["selected"]==1)
                        echo '<option value="'. $id_check["cod"].
'" selected>'. $lang[$id_check["desc"]].'</option>';
                else
                        echo '<option value="'. $id_check["cod"].' ">'.
$id_check["desc"].'</option>';
        echo "</select>";
echo '</div>';

// DIV VUOTO
echo '<div class="div-input">';
echo '</div>';

// UPLOAD FILE
echo '<div class="div-input">';
        echo '<div id="upcheck" data-bcolor="green"
data-bico="upload" data-bcap="'.
$lang["ADD_CONFORMITY"].'"></div>';
        if($motore['check']=="0")

```

```

                echo '<input type=checkbox id="mot-check"
name="mot-check" value="1" />';
            else
                echo '<input type=checkbox id="mot-check"
name="mot-check" value="1" checked /> ';
                echo '<label for="mot-check">'.
$lang["add-check"].'</label><br />';
            echo '</div>';

            echo '<div class="div-input doubled">';
            echo '<div id="upallcheck" data-bcolor="blue"
data-bico="paperclip" data-bcap="' . $lang["ADD_ATTACH"] . "'></div>';
            echo '</div>';

// DIV VUOTO
echo '<div class="div-input">';
echo '</div>';

// DIV SUBMIT
echo '<div class="div-input">';
echo '<div class="btn-but btn-blue"
onclick="$(this).parent().parent().submit()">
<i class="icon white ok"></i> ' . $lang["CONFIRM"] . '</div>';
echo '</div>';

// DIV VUOTO
echo '<div class="div-input">';
echo '</div>';

echo '</form>';

```

Figura 8: Codice sorgente relativo alla form di inserimento dei dati di check.

```

function createUploader(obj,cmd,maxfile,image){
    if(obj.length==0)return false;
    var strbut='<div class="btn-but btn-'+obj.data("bcolor")+
'" style="margin:5px 2px"><i class="icon white '+
obj.data("bico")+'"></i> '+obj.data("bcap")+</div>';
    var body=$( "<div></div>" ).appendTo(obj);
    var button,iload,ifile,iform;
    var fileList=[];
    var isIE=$( 'body' ).hasClass( 'ie' );
    var tthis=this;
    if(isIE){
        //----- hack for IE9 and IE8
        iform=$( '<form method="post" target="iframe-'+obj.attr('id')
+' " action="'+domain+cmd+
'_up?nojson=1" enctype="multipart/form-data"></form>' )
        .appendTo( $( 'body' ) ).submit( function( )
        {iload.show();button.remove();ifile.hide();});
        ifile=$( '<input type="file" name="file"
style=\'font-size:50px;cursor:pointer;position:absolute;
opacity:0.0;-moz-opacity:0.0;
-ms-filter:"progid:DXImageTransform.Microsoft.Alpha(Opacity=0)";\'/>' )
        .appendTo(iform).change( function( ){iform.submit();});
        $( '<iframe id="iframe-'+obj.attr('id')+'
" name="iframe-'+obj.attr('id')
+' "style="display:none"></iframe>' )
        .appendTo( $( 'body' ) )
        .on( 'load', function( ){
        if( $( this ).contents( )[0].location.href.match( domain+cmd+'_up' ) ){
        var resp=$( this ).contents( ).find( 'body' ).html( );
            if( resp[0]== '{' ){
                resJSON=JSON.parse( resp );
                if( resJSON.response!="OK" )
                    alert( resJSON.alert );
                else
                    fileList=resJSON.files;
            }else{
                alert( "Connection Error!!!" );
            }
            tthis.render( );
            var cfile=ifile.clone( true );
            ifile.replaceWith( cfile );
            ifile=cfile;
        }
        });
    }else{
        ifile=$( '<input type="file"
name="file" style="display:none" />' )
        .appendTo( 'body' ).change( function( ){tthis.loadFile();});
    }

    this.render=function(){
        body.empty();
        for( var i= 0; i < fileList.length; i++){
            $( '<i class="icon remove but" data-file="'+fileList[i
+' "></i> ' ).appendTo( body ).click( function( ){
                if( confirm( alert_form_msgs.go ) )
                    return tthis.delFile( $( this ).data( "file" ) );
            } );
        }
    }
}

```

```

    });
    if(image)
        body.append('<br />');
    else
        body.append('<a title="'+fileList[i]+' " href="'
+domain+cmd+'_tmp/0-'+fileList[i]+' " target="_blank">'
+fileList[i]+'</a><br />');
    }
    if(fileList.length>=maxfile)return;
    iload=$('').appendTo(body);
    button=$(strbut).appendTo(body);
    body.append('<div style="display:none;width:135px;
height:15px;border:1px solid #1A6DDE;
background:#FFFFFF;margin-left:2px">
<div style="width:0px;height:15px;background:#78ACF8"></div></div>');
    if(button.is(':visible')){
        if(isIE){
            //----- hack for IE9 and IE8
            ifile.show();
            ifile.width(button.outerWidth());
            ifile.height(button.outerHeight());
            ifile.offset(button.offset());
            button.onOffsetChanged(
function(lastOff,newOff){ifile.offset(newOff);});
        }else{
            button.click(function(){ifile.click();});
        }
    }
}

this.loadFile=function(){
    var file = ifile.get(0);
    if(file.files[0].size>maxfilesize){
        var strSize=maxfilesize+" Bytes";
        if(maxfilesize>1024)strSize=parseFloat((maxfilesize/1024).toFixed(2))
+" KB";
        if(maxfilesize>1048576)strSize=parseFloat((maxfilesize/1048576)
.toFixed(2))+ " MB";
        alert(alert_file.size.replace(/\\%s/g, strSize));
        ifile.val("");
        return;
    }
    var formData = new FormData();
    formData.append('file', file.files[0]);
    ifile.val("");
    button.hide();
    button.next().show();
    var pbar=button.next().find('> div');
    $.ajax({
        url: domain+cmd+"_up",

```

```

        type: 'POST',
        data: formData,
        dataType: 'json',
        cache: false,
        contentType: false,
        processData: false,
        success: function(resJSON)
        {
            if(resJSON.response!="OK")
                {alert(resJSON.alert);return;}
            fileList=resJSON.files;
        },
        error: function(){alert("Connection Error!!!");},
        complete: function(){tthis.render();},
        xhr: function(){
            var xhr = new window.XMLHttpRequest();

xhr.upload.addEventListener("progress",function(evt){
                if (evt.lengthComputable){
pbar.width(pbar.parent().width()*evt.loaded/evt.total);
                    }
                },false);
            return xhr;
        }
    });
}

this.delFile=function(fname){
    $.post(domain+cmd+'_del/0-'+fname,"",function(resJSON){
        if(resJSON.response!="OK"){alert(resJSON.alert);return}
        fileList=resJSON.files;
        tthis.render();
    })
    .fail(function(){alert("Connection Error!!!")})
    .always(function(){return;});
    return false;
}
this.delFile('');
}

```

Figura 9: Codice JavaScript per l'upload dei file.

```

...

//----- list of records
$fields=array("id_mot", "modello", "web", "description", "name",
"potenza", "poli", "tensione", "classe", "eta", "test", "corrente",
"frequenza", "giri", "fattore", "peso", "color_check",
"descrizione_check", "tensione_check", "corrente_check",
"potenza_check", "frequenza_check", "giri_check", "descrizione_check",
"Tcase_check", "Tcooler_check", "data_crea_check", "user_crea_check",
"data_mod_check", "user_mod_check");
while($rmot=$mot->fetch_assoc()){
    $record=array();
    foreach($fields as $ff)
        $record[$ff]=$rmot[$ff];

    $record["write"]=false;          //----- privileges
    $record["verify"]=false;
    // SS VERIFICA SOLO DA UN AMMINISTRATORE
    if($_SESSION['logged_in'] &&
(($rmot["id_azienza"]==$_SESSION['param']['azienda'] &&
$_SESSION['param']['motori']=='1') || $_SESSION['param']['admin']=='1'))
    {
        $record["write"]=true;
        if ($_SESSION['param']['admin']=='1')
            $record["verify"]=true;
    }
    //----- test report file
    $record["report"]["title"]=$lang["LIST_TEST_LINK"];
    $record["report"]["path"]="NOFILE";
    if(listFiles('motori/'.$record["id_mot"].'.report')){
        $record["report"]["path"]=$domain.ACT_MOT."/report/".
$record["id_mot"]."-Test_Report.pdf";
        if($record['test']==0 && !$record["write"])
            $record["report"]["alert"]=$lang["ALERT_LINK_PRIVATE"];
        if(!$_SESSION['logged_in'])
            $record["report"]["alert"]=$lang["ALERT_LINK"];
    }else{
        $record["report"]["alert"]=$lang["ALERT_LINK_NOFILE"];
    }
    //----- doc files
    $files=listFiles('motori/'.$record["id_mot"].'.doc');
    foreach($files as $file){
        $doc=array();
        $doc["path"]=$domain.ACT_MOT."/doc/".$record["id_mot"]."-".
$file;
        $doc["title"]=$file;
        if(!$_SESSION['logged_in'])
            $doc["alert"]=$lang["ALERT_LINK"];
        $record["docs"][]=$doc;
    }
    //----- check report file
    $record["check"]["title"]=$lang["LIST_CHECK_LINK"];
    $record["check"]["path"]="NOFILE";
    if(listFiles('motori/'.$record["id_mot"].'.check')){
        $record["check"]["path"]=$domain.ACT_MOT."/check/".

```

```

$record["id_mot"]."-Check_Report.pdf";
    if($record['test']==0 && !$record["write"])
        $record["check"]["alert"]=$lang["ALERT_LINK_PRIVATE"];
    if(!$SESSION['logged_in'])
        $record["check"]["alert"]=$lang["ALERT_LINK"];
}else{
    $record["check"]["alert"]=$lang["ALERT_LINK_NOFILE"];
}
//----- check doc files
$files=listFiles('motori/'. $record["id_mot"].'.checkdoc');
foreach($files as $file){
    $doc=array();
    $doc["path"]=$domain.ACT_MOT."/checkdoc/".
$record["id_mot"]."-".$file;
    $doc["title"]=$file;
    if(!$SESSION['logged_in'])
        $doc["alert"]=$lang["ALERT_LINK"];
    $record["docs_check"][]=$doc;
}
//----- admin details
if($SESSION['logged_in'] && $SESSION['param']['admin']=='1'){
    $record["admin"]["user_crea"]=$rmot['id_crea'];
    $record["admin"]["data_crea"]=$rmot['data_crea'];
    $tmp=$mysqli->query("SELECT CONCAT(name,' ',surname) AS name
FROM `".$prefixdb."users` WHERE `id`='".$rmot['id_crea']."'");
    if($tmp->num_rows > 0){
        $rtmp=$tmp->fetch_assoc();
        $record["admin"]["user_crea"]=$rtmp['name'];
    }
    $record["admin"]["user_mod"]=$rmot['id_mod'];
    $record["admin"]["data_mod"]=$rmot['data_mod'];
    $tmp=$mysqli->query("SELECT CONCAT(name,' ',surname) AS name
FROM `".$prefixdb."users` WHERE `id`='".$rmot['id_mod']."'");
    if($tmp->num_rows > 0){
        $rtmp=$tmp->fetch_assoc();
        $record["admin"]["user_mod"]=$rtmp['name'];
    }
}
$jsonvar["records"][]=$record;
}

...

if(!$SESSION['logged_in'] || $SESSION['param']['motori']!='1' ||
$id=="")
    goIndex(ACT_MOT);

$qr=$mysqli->query("SELECT * FROM ".$premotdb."motori WHERE id='".$id.'"
AND '".$SESSION['param']['admin']."'");

if($qr->num_rows==0)
    goIndex(ACT_MOT);

$motore=$qr->fetch_assoc();
$mytemp->set_var("motore",$motore);
/* CAMPI DELLA TABELLA MOT_CHECKDATA */

```

```

$qqr=$mysqli->query("SELECT * FROM ".$premotdb."checkdata WHERE
id_motore='".$id."'");
/* SE LA IL RESULT SET DELLA QUERY E' VUOTO INSERISCO VALORI NULLI IN
TUTTI I CAMPI */
if($qqr->num_rows==0) {
    $fields=array("id_motore", "tensione", "corrente", "potenza",
"frequenza", "giri", "cod_check", "temp_case", "temp_cooler");
    // CREATE EMPTY CHECK DATA ENTRY
    $sql="INSERT INTO `".$premotdb."checkdata` (";

    foreach($fields as $vv)
        $sql.="`".$vv.``, ";
    $sql.="`id_crea`, `data_crea`, `id_mod`, `data_mod`) VALUES ('" .
$id . "', '0', '0', '0', '0', '0', 'UN', '0', '0',";
    $sql.="'" . $_SESSION['param'][$id] . "', '" . date('Y-m-d H:i:s') . "',
'" . $_SESSION['param'][$id] . "', '" . date('Y-m-d H:i:s') . "')";

    //echo $sql . "<br>";

    $mysqli->query($sql);
    $lastid=$mysqli->insert_id;
    /* RICERCA CAMPI AGGIUNTIVI NELLA TABELLA MOT_CHECKDATA */
    $qqr=$mysqli->query("SELECT * FROM ".$premotdb."checkdata WHERE
id_motore='".$id."'");
}
$checkdata=$qqr->fetch_assoc();
$mytemp->set_var("checkdata",$checkdata);

// COMBO CHECK
$qqr = $mysqli->query("SELECT *,IF(cod='".$checkdata['cod_check'].'",1,0)
AS selected FROM ".$premotdb."checks ORDER BY cod");
$item=array();
while($item=$qqr->fetch_assoc())
    $items[]=$item;
$mytemp->set_var("checks",$items);

...

```

Figura 10: Script php per la gestione dei file.

```

md5_file ( string $filename [, bool $raw_output = FALSE ] ) : string

```

Figura 11: Funzione php per il calcolo della checksum MD5.

Appendice

Procedura di installazione

Preventivamente alla installazione dei codici occorre verificare che mysql e php siano installati sulla macchina che ospiterà il portale e che le relative versioni siano almeno pari a quelle minime di seguito indicate:

- mysql >= 5.4
- php >= 5

A tal fine è sufficiente digitare i comandi

```
mysql --version
php --version
```

Successivamente occorre copiare l'intero progetto nella cartella desiderata (es. /var/www/enea/macchine_elettriche/) avendo cura di specificare il nome della cartella nei file di configurazione di Apache. I file di configurazione di Apache hanno estensione .conf e nel caso di macchine Unix/Linux si trovano tipicamente nella cartella /etc/apache2/ o, nel caso di Virtual Host, nella cartella /etc/apache2/sites-available/.

Si riporta di seguito un possibile file di configurazione fornito con i codici sorgente (www.motorielettrici.enea.it.conf):

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    ServerName motorielettrici.enea.it
    ServerAlias www.motorielettrici.unime.it
    ServerAlias macchineelettriche.unime.it
    ServerAlias macchineelettriche.enea.it
    ServerAlias trasformatori.unime.it
    ServerAlias trasformatori.unime.it

    DocumentRoot /var/www/enea/macchine_elettriche/public

    <Directory /var/www/enea/macchine_elettriche>
        Options FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

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

Caricato il file di configurazione nella cartella /etc/apache2/sites-available/ è necessario caricare il modulo rewrite di apache:

```
sudo a2enmod rewrite
```

e riavviare il server Apache:

```
sudo service apache2 restart
```

Infine per l'aggiornamento del database motori, occorre eseguire il seguente comando

```
mysql -u[USER] -p[PASSWORD] -Dmotori < motori_extension.sql
```

dove [USER] e [PASSWORD] vanno sostituiti con le credenziali di accesso al database e

motori_extension.sql è un file fornito con i codici sorgente contenente le seguenti query:

```
CREATE TABLE IF NOT EXISTS `mot_checkdata` (  
  `id` INT(11) NOT NULL AUTO_INCREMENT,  
  `id_motore` INT(11) NOT NULL UNIQUE,  
  `tensione` DOUBLE NOT NULL,  
  `corrente` DOUBLE NOT NULL,  
  `potenza` DOUBLE NOT NULL;  
  `frequenza` INT(11) NOT NULL,  
  `giri` INT(11) NOT NULL,  
  `cod_check` VARCHAR(10) NOT NULL,  
  `temp_case` DOUBLE NOT NULL,  
  `temp_cooler` DOUBLE NOT NULL,  
  `data_crea` DATETIME NOT NULL,  
  `id_crea` INT(11) NOT NULL,  
  `id_mod` INT(11) NOT NULL,  
  `data_mod` DATETIME NOT NULL,  
  PRIMARY KEY (`id`),  
  FOREIGN KEY (`cod_check`) REFERENCES `mot_checks` (cod))  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8;  
  
INSERT INTO `eff_colors` (`cod`, `desc`) VALUES ('GRE', 'Green');  
INSERT INTO `eff_colors` (`cod`, `desc`) VALUES ('RED', 'Red');  
INSERT INTO `eff_colors` (`cod`, `desc`) VALUES ('YEL', 'Yellow');  
INSERT INTO `eff_colors` (`cod`, `desc`) VALUES ('GRA', 'Gray');  
  
INSERT INTO `mot_checks` (`cod`, `desc`, `cod_color`) VALUES ('UN',  
'Unevaluated', 'GRA');  
INSERT INTO `mot_checks` (`cod`, `desc`, `cod_color`) VALUES ('OK',  
'Approved', 'GRE');  
INSERT INTO `mot_checks` (`cod`, `desc`, `cod_color`) VALUES ('ERR',  
'Rejected', 'RED');  
INSERT INTO `mot_checks` (`cod`, `desc`, `cod_color`) VALUES ('RAT',  
'Rating', 'YEL');  
INSERT INTO `mot_checks` (`cod`, `desc`, `cod_color`) VALUES ('INT',  
'Integrations', 'YEL');
```



Giuseppe Campobello nasce a Messina, Italia, nel 1975.

Riceve 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 2004 al 2006 è stato titolare di contratti finalizzati ad attività di ricerca e professore a contratto della Facoltà di Ingegneria e della Facoltà di Scienze MM.FF.NN. dell’Università di Messina.

Nel 2006 è stato altresì assegnista di ricerca presso il Dipartimento di Matematica e Informatica dell’Università di Catania.

Nel dicembre del 2006 vince il concorso per Ricercatore Universitario del settore ING-INF/03 (Telecomunicazioni) presso l’Università degli Studi di Messina.

Attualmente afferisce al Dipartimento di Ingegneria dell’Università di Messina dove è Ricercatore Confermato a tempo pieno oltre che 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, dell’elaborazione numerica dei segnali e dell’elettronica digitale applicata alle telecomunicazioni. In particolare l’attività di ricerca più recente è focalizzata sulle reti di sensori wireless e sulle tecniche di compressione.

È autore di oltre trenta articoli scientifici apparsi su riviste internazionali o presentati a conferenze internazionali e revisore di diverse riviste internazionali della IEEE e della Elsevier.

È inoltre membro del Consiglio Scientifico del Gruppo Telecomunicazioni e Tecnologie dell’Informazione (GTTI) e del Microwave Engineering Center for Space Applications (MECSA).



Antonino Segreto nasce a Messina, Italia nel 1979.

Consegue la Laurea quinquennale in Ingegneria Elettronica nel 2011 e il Dottorato di Ricerca (Ph.D) in “Tecnologie avanzate per l’Optoelettronica, la Fotonica e Modellizzazione Elettromagnetica” nel 2015, entrambe presso l’Università degli Studi di Messina.

Dal 2015 ad oggi è collaboratore di ricerca presso il Dipartimento di Ingegneria dell’Università degli Studi di Messina.

L’attività di ricerca riguarda principalmente le reti di telecomunicazioni, l’elaborazione dei segnali e le reti di sensori wireless.