

- [Introduzione](#)
- [Caratteristiche principali](#)
- [Requisiti software](#)
- [Struttura dell'applicazione](#)
- [Configurazione](#)
  - [campo tipo text](#)
  - [campo tipo select](#)
  - [campo tipo textarea](#)
  - [camp tipo file](#)
  - [campo tipo date](#)
  - [campo tipo booleano](#)
- [Avvio al termine della configurazione](#)
- [Note d'uso con screenshot esplicativi](#)
- [Condizioni legali di utilizzo](#)

## INTRODUZIONE

Ho sviluppato per uso personale e poi deciso di pubblicare una semplice applicazione che possiede una parte delle funzionalità di analoghe a pagamento. Prodotti come <https://phpgrid.com/>, <https://phpmaker.dev/> o <https://www.phpgrid.org>, offrono innumerevoli e pregevoli features; purtroppo una di esse molto utile è sempre offerta a pagamento, ovvero la visualizzazione master/detail che in una stessa pagina web mostra le tabelle in relazione uno a molti tipica dei database relazionali. Il pacchetto <https://datatables.net/> è gratuito ma a mio giudizio complesso per un principiante. Inoltre un manuale d'uso in italiano potrebbe inizialmente aiutare ad impraticarsi perché i migliori tutorial e video online per i suddetti pacchetti sono in lingua inglese. Avevo inoltre necessità di semplificare la pesantezza della procedura legata ai numerosi settaggi necessari prima di poter usare l'applicazione; il risultato è stato ciò che potete vedere in <http://wwwdisc.chimica.unipd.it/michele.furlan>.

Consiglio di leggere fino in fondo le istruzioni qui presenti perché aiutano ad avere una migliore visione d'insieme e di cogliere utili potenzialità. Nella demo visibile in <http://wwwdisc.chimica.unipd.it/michele.furlan> è inibito agire sui record, ciò al fine di mantenere l'integrità del database pubblicato. Per avere le funzionalità [CRUD](#) non presenti nella demo del sito bisogna installare l'applicazione su un proprio server.

## CARATTERISTICHE PRINCIPALI

Infinite tabelle annidate in relazione master/detail (uno a molti), tabelle multiple, inline editing di campi, operazioni di aggiunta, e cancellazione dei record, upload dei file in cartelle specifiche per ogni campo con indicazione percentuale upload sul server (progress bar), preview dei file immagine, visualizzazione in modalità tabulare e verticale, esportazione dei dati in formato excel xml, filtro di ricerca e ordinamento sui campi, paginazione, istanze multiple nella stessa pagina WEB, scroll automatico di pagina su espansione delle tabelle, codice sorgente commentato di breve lunghezza per una eventuale agevole comprensione e personalizzazione. Linguaggi adottati: php, sql, html5, javascript, css3, [libreria JQuery](#) e plugins ad essa correlato [JQuery.form](#).

## REQUISITI SOFTWARE

Un server web con il pacchetto AMP (Apache, PHP, MySQL,) sotto Windows o Linux con versioni non troppo datate, mentre lato client un browser anche non recente con supporto HTML5.

## STRUTTURA DELL'APPLICAZIONE

Scompattato il file **apptabelle.zip** troviamo i seguenti file:

**index.php**: è il primo file caricato nel browser, serve per il caricamento delle tabelle di esempio nel browser e va personalizzato;

**config.php**: contiene le variabili e settings di configurazione, tra cui la descrizione [del modello del database](#);

**function.php**: [le routine](#) php che gestiscono lato server i comandi impartiti dall'utente nel browser;

**fileupload.php**: insieme di routine l'eventuale presenza di campi designati all'upload e visualizzazione di singoli file da caricare sul server;

**crud.php**: riceve dal browser i comandi di cancellazione, creazione ecc... campi e record in tabella e smista le chiamate ricevute invocando in function.php la funzione appropriata;

**testapp.php**: contiene le routine di verifica della configurazione in config.php;

**myapptabelle.js**: che sta dentro la cartella **js** insieme a due file della libreria JQuery: **jquery-3.4.1.min.js** e **jquery.form.min.js**

## CONFIGURAZIONE

Fare riferimento al database di esempio visibile in <http://www.disc.chimica.unipd.it/michele.furlan>, le istruzioni SQL del DB sono contenute nel file **database-esempio.sql**. In esso sono anche contenuti i dati che volendo si possono importare, ma non sono presenti i file nelle cartelle che vedete nella demo online. L'esempio mostra un'ipotetica gestione d'inventario costituita da un insieme di edifici detti Dipartimenti registrati nella tabella master, da una tabella *stanze* in relazione con la tabella *dipartimenti* che raggruppa le stanze che afferiscono al medesimo dipartimento, e in ultima la tabella *oggetti* che rappresenta i singoli beni contenuti in una stanza.

Il file **config.php** è commentato per la maggior parte delle righe. In cima troviamo le variabili per la connessione al database mysql: **\$servername**, **\$dbname**, **\$username** e **\$password**. Seguono **\$massima\_dimensione\_file** e **\$quota\_massima** che fissano la massima dimensione caricabile sul server di un singolo file, e la quota massima di spazio disponibile per ogni cartella destinata a ricevere i file inviati; sono due impostazioni globali che hanno efficacia su tutti i campi di tipo file. La parte più impegnativa della configurazione riguarda la descrizione del modello del nostro DB (database) che è tutta contenuta nell'array associativo **\$tabella**. Osservate com'è costruito il **config.php** nell'esempio per comprendere come costruire il modello. Le tabelle nel DB che sono in relazione uno a molti devono essere tra loro collegate con alcuni appositi ed obbligatori campi chiave. In tutte le tabelle deve essere presente il campo **id** che funge da chiave primaria; normalmente un tipo di dato numerico autoincrement o temporizzato. Le tabelle "figlie", che sono in relazione con le prime nel classico schema uno a molti o master/detail deve essere indicato il nome della chiave esterna presente nella/nelle vostre tabelle details; quest'ultimo è un campo obbligatorio che serve a mettere in relazione le tabelle. Il file **config.php** nella variabile **\$tabella** descrive il modello del database che trovate nella cartella documentazione: **database-esempio.sql**. E' possibile testare l'esempio installando in proprio server il pacchetto PHP+MYSQL+WEB SERVER e importando nel database il file **database-esempio.sql** (es. con il comando `mysql -u root -p password <database-esempio.sql`).

Riassumo il tutto con la spiegazione sotto:

```
$tabella["dip"] =  
array(  
    "nometabella" => "dipartimenti",  
    "figlia" => "stanze",  
    "chiave_padre" => "",  
    "paginazione" => 6,  
    "pulsanti" => array(  
        "verticale" => TRUE, "add" => TRUE,  
        "delete" => TRUE, "filtro" => TRUE),  
    "campi" => VEDI SOTTO TIPO TEXT
```

- **"dip"**: è una chiave dell'array **\$tabella** che deve essere scelta univoca. Il nome della chiave deve contenere **solo caratteri alfanumerici**. Consiglio caldamente sia seguita tale convenzione per tutti i nomi delle chiavi degli array.
- **"dipartimenti"**: è il nome effettivo della tabella nel database (DB)
- **"figlia"**: è la chiave della tabella figlia nell'array **\$tabella**
- **"chiave\_padre"**: è il nome che nel DB è chiave esterna nella tabella "dipartimenti" che compie i match (JOIN) con la corrispondente chiave primaria id nella tabella padre. In esempio "chiave\_padre" è vuota perché non ci sono tabelle da cui dipartimenti dipende.
- **"paginazione"**: numero di righe massime in tabella. Inserire 0 per disattivare la paginazione.
- **"pulsanti"**: un array dove autorizzare le operazioni sui campi in tabella. **"verticale"** è un booleano che attiva se TRUE la visualizzazione in forma verticale della riga, nella quale include tutti i campi, non solo i campi esplicitamente definiti. **"add"**, **"delete"** e **"filtro"** se TRUE abilitano l'aggiunta, la cancellazione e il filtro dei dati.

I tipi di campo disponibili sono: **text**, **textarea**, **select**, **file**, **data**, **booleano** e **calcolato**; di seguito trovate la descrizione di ciascuno. Il mio consiglio è di osservare i campi presenti nel database di esempio perché contengono gli esempi di tutti i controlli di input usabili. E' obbligatorio specificare le chiavi degli array, la loro omissione o diversa denominazione implicherà un warning e la mancata inizializzazione dell'applicazione.

### Tipo: TEXT

```
"campi" => array("denominazione" =>  
array(  
    "tipo" => "text",  
    "etichetta" => "Denominazione",  
    "editable" => TRUE,  
    "size_filtro" => 20,  
    "attributi" => array(  
        "default_value" => "",  
        "size" => 20,  
        "maxlength" => 100,  
        "pattern" => "[A-Za-z]{3}",  
        "error_pattern" => "solo tre lettere")),
```

**"campi"**: apre l'elencazione dei campi in tabella.  
**"denominazione"**: è il nome del campo nella tabella del database, nel caso in esempio nella tabella dipartimenti. Il nome deve rispettare le convenzioni di MySQL, consiglio siano in lettere minuscole privi di caratteri speciali o di spaziatura e assolutamente privi del segno **"-"**;  
**"tipo" => "text"**: campo visivamente in tabella html tipo `<input type="text">`;  
**"etichetta" => "Denominazione"**: il nome che verrà visualizzato nell'intestazione della colonna in tabella;  
**"size\_filtro"**: larghezza in colonne del campo dove inserire il valore da filtrare  
**"attributi"**: è un array che precisa gli attributi specifici del campo di tipo **"text"**  
**"default\_value"**: il valore che verrà inserito nel campo quando si aggiunge un nuovo record in tabella;  
**"size"**: larghezza del campo di input in colonne;  
**"maxlength"**: massima lunghezza del testo immettabile dall'utente;  
**"pattern"**: contiene un' [espressione regolare](#) in formato html5 che serve a validare i dati immessi dall'utente (ovvero stabilisce dei vincoli), in esempio a lato è consentita l'immissione di tre lettere non case sensitive. L'uso delle espressioni regolari serve a forzare il tipo di dato, solo numeri, solo lettere, email ecc...  
**"error\_pattern"**: il messaggio di errore che l'utente visualizza in caso di mancata corrispondenza del dato immesso con il pattern.

## Tipo: SELECT

```
"indirizzo" => array(
"tipo" => "select",
"etichetta" => "Indirizzo",
"editable" => TRUE,
"size_filtro" => 15,
"attributi" => array(
"default_vuoto" => "-----",
"VV" => "Via Verdi", "VM" => "Via Monte
Rosso", "VV" => "Via Volta", "VT" =>
"Via Termoli")),
```

"indirizzo": come sopra per il campo di tipo "text", è il nome del campo in tabella nel DB;  
"tipo" => "select": indica la rappresentazione di un controllo di input di tipo select, in HTML  
<SELECT><OPTION>...</OPTION><OPTION>...</OPTION>...</OPTION>...</SELECT>  
"etichetta", "editable", "size\_filtro", "attributi": vale quanto già detto sopra per il campo di  
"tipo" => "text". Inizia ora la specifica vera e propria delle opzioni disponibili nell'array con  
chiave "attributi";  
"default\_vuoto": la chiave deve essere sempre presente nel array "attributi" e indica il  
valore stringa vuota "" che sarà visualizzato all'utente con la stringa "-----" che sarà  
selezionabile se si vorrà annullare il valore nel campo **o se il valore del campo non è  
presente nella lista successiva**. Di seguito nella forma "chiave" => "valore", chiave è il  
valore che verrà registrato nel campo del DB quando si seleziona valore. Se seleziono "Via  
Verdi", nel campo sarà registrato "VV". Chiave e valore possono coincidere.

## Tipo: TEXTAREA

```
"note" => array(
"tipo" => "textarea",
"etichetta" => "Annotazioni",
"editable" => TRUE,
"size_filtro" => 25,
"attributi" => array(
"default_value" => "",
"rows" => 2,
"cols" => 30,
"maxlength" => 100)),
```

"note": nome del campo in tabella (dipartimenti nel nostro caso);  
"tipo" => "textarea": indica di rappresentare un controllo di input di tipo textarea, in HTML  
<TEXTAREA rows="" cols=""></TEXTAREA>;  
"etichetta", "editable", "size\_filtro", "attributi": vale quanto già detto sopra per il campo di  
"tipo" => "text";  
"default\_value": valore che verrà inserito nel campo quando si aggiunge un nuovo record  
in tabella;  
"rows": numero di righe del controllo di input TEXTAREA;  
"cols": numero di colonne del controllo;  
"maxlength": numero massimo di caratteri immettibili dall'utente.

## Tipo: FILE

```
"foto" => array(
"tipo" => "file",
"etichetta" => "Foto",
"editable" => TRUE,
"size_filtro" => 15,
"attributi" => array(
"cartella_upload" => "upload",
"larghezza_max_minatura" => 70)),
```

"foto": nome del campo in tabella (dipartimenti nel nostro caso);  
"tipo" => "file": consente di fare l'upload nella cartella sotto designata nella chiave  
"cartella\_upload" di file di qualsiasi formato. La dimensione massima del file è stabilita nella  
variabile globale presente in cima a **config.php**: \$massima\_dimensioe\_file. La massima  
dimensione del file è inoltre gestita dalle variabili sul server, in genere in **php.ini**.  
"etichetta", "editable", "size\_filtro", "attributi": vale quanto già detto sopra per il campo di  
"tipo" => "text";  
"cartella\_upload" => "upload": specifica la cartella nel server dove deporre i file. Nel campo  
in tabella (di tipo varchar) viene memorizzato solo il nome del file. La cartella di upload va  
preventivamente creata sul server perché l'applicazione non lo fa per voi. Deve essere  
inoltre annidata dentro la cartella dell'applicazione medesima. Potreste creare per diversi  
campi di tipo file chiavi di "cartella\_upload" nel seguente modo "upload/foto",  
"upload/pdf", "upload/convegno2019" raggruppando tutti gli upload dei file sotto l'unica  
cartella upload. Esiste in config.php la variabile globale \$quota\_massima che indica la  
dimensione massima in MB consentita per "cartella\_upload".  
"larghezza\_max\_minatura": valore numerico in pixel che limita la larghezza dell'anteprima  
dei file di tipo immagine nella colonna in tabella associata al campo di tipo file.

## Tipo: DATE

```
"datainserimento" => array(
"tipo" => "date",
"etichetta" => "Data acquisizione",
"editable" => TRUE,
"size_filtro" => 20,
"attributi" => array(
"default_value" => "2019-11-11",
"max" => "2022-12-31",
"min" => "1979-01-01",
"pattern" => "\d{4}-\d{2}-\d{2}",
"error_pattern" => "immettere formato  
anno-mese-giorno (YYYY-MM-DD)"),
```

"datainserimento": nome del campo in tabella;  
"tipo" => "date": controllo di tipo date che in HTML5 permette di selezionare la data da una  
finestra grafica e salvare il valore in un stringa nel formato anno-mese-giorno; anno a  
quattro cifre, mentre mese e giorno con due;  
"etichetta", "editable", "size\_filtro", "attributi": vale quanto già detto sopra per il campo di  
"tipo" => "text";  
"default\_value": posiziona il cursore della finestra data alla data in formato YYYY-MM-GG  
corrispondente quando si crea un nuovo record;  
"max": valore in formato YYYY-MM-DD che indica il limite superiore della data  
selezionabile;  
"min": valore in formato YYYY-MM-DD che indica il limite inferiore della data selezionabile;  
"pattern": non alterare il valore perché è rappresenta l'espressione regolare che forza il  
formato data.  
"error\_pattern": messaggio di avviso quando il campo data è difforme al pattern.

## Tipo: BOOLEANO

```
"magazzino" => array(
"tipo" => "booleano",
"etichetta" => "Magazzino",
"editable" => TRUE,
"size_filtro" => 20,
"attributi" => array(
"default_value" => "0")),
```

**"magazzino"**: nome del campo in tabella (nell'esempio nella tabella stanze);  
**"tipo" => "booleano"**: controllo di tipo checkbox che memorizza 0 (==FALSE) o 1 (TRUE) in un campo sottostante nel DB mysql di tipo **tinyint(1)**;  
**"etichetta"**, **"editable"**, **"size\_filtro"**, **"attributi"**: vale quanto già detto sopra per il campo di **"tipo" => "text"**;  
**"default\_value"**: è d'obbligo indicare 0 (==FALSE) o 1 (==TRUE), non è previsto il tipo NULL in questo tipo di campo.

## Tipo: CALCOLATO

```
"calcolato1" => array(
"tipo" => "calcolato",
"etichetta" => "IF(magazzino, 'DEPOSITO',
'UFFICIO')",
"size_filtro" => 30,
"formula" => "IF(magazzino, 'DEPOSITO', 'UFFICIO')")
```

**"calcolato1"**: nome del campo, non ha alcuna corrispondenza in una tabella nel DB;  
**"tipo" => "calcolato"**: indica che il valore non è registrato in tabella ma è il risultato di un calcolo effettuato su altri campi;  
**"etichetta"**: nome del campo nelle intestazioni della tabella (qui ho messo la formula);  
**"size\_filtro"**: dimensione in pixel del campo di ricerca;  
**"formula"**: deve essere una valida espressione mysql che restituisce un valore a partire dai campi in essa contenuti. Nell'esempio se magazzino è TRUE restituisce la stringa DEPOSITO diversamente UFFICIO. *Il campo viene sempre aggiornato in seguito al cambiamento del valore/i contenuti nell'espressione.* Provare in esempio a modificare il valore booleano per verificarne l'effetto.

## AVVIO AL TERMINE DELLA CONFIGURAZIONE

Per rappresentare le tabelle in una pagina web, bisogna caricare in essa nell'ordine: la libreria **jQuery**, lo script **myscript.js**, inizializzare l'applicazione ed in ultima caricare le tabelle.

La prima e tutte le successive eventuali tabelle principali (le master) che desiderate visualizzare devono trovarsi all'interno della pagina web inserite in tag vuoti ad esse dedicati necessari a posizionarle dove preferite. E' consigliato un tag DIV dedicato con un proprio ID univoco e privo di contenuti (es. `<DIV id="div_tabella_dipartimenti"></DIV>`). Il caricamento delle tabelle avviene usando istruzioni javascript (vedere nel file di esempio **index.php**). Riassumendo, osservate il file **index.php** che carica la demo in <http://www.disc.chimica.unipd.it/michele.furlan> e vedrete in cima il seguente codice:

```
<SCRIPT type="text/javascript" src="./js/jquery-3.4.1.min.js"></SCRIPT> <!-- Carica libreria JQuery -->
<SCRIPT type="text/javascript" src="./js/myappatabelle.js"></SCRIPT> <!-- Carica lo script di gestione della app -->
<SCRIPT type="text/javascript">
// Solo dopo il caricamento completo della pagina si possono caricare le tabelle
$(document).ready(function() {
var objTabelle;
objTabelle = new AvviaAppTabelle(); // Crea l'oggetto AvviaAppTabelle
if(objTabelle.inizializzato) { // Solo se inizializzazione riuscita carico le tabelle
// Invocazione del rendering tabella "dip" dentro il DIV "corpo_tabella_dip"
objTabelle.CaricaTabella("corpo_tabella_dip", "dip");
// Invocazione del rendering tabella "oggettibis" dentro il DIV "corpo_tabella_oggettibis"
objTabelle.CaricaTabella("corpo_tabella_oggettibis", "oggettibis");
// Invocazione del rendering tabella "calcolata" dentro il DIV "corpo_tabella_calcolata"
objTabelle.CaricaTabella("corpo_tabella_calcolata", "calcolata");
}
}); // fine document ready
```

Con il **costruttore** **new AvviaAppTabelle()** si inizializza l'ambiente di rendering delle tabelle, che verifica la compatibilità HTML5 del browser e la correttezza dei settings in **config.php**. Gli errori rilevati verranno mostrati a video in una **finestra modale** all'interno del browser. Il metodo **CaricaTabella()** accetta due argomenti **obbligatori**, il primo l'id del tag html dove inserire la tabella, il secondo il nome della tabella master presente nell'array **\$tabella** in **config.php**. Per caricare ulteriori tabelle master nella medesima pagina si usa nuovamente il metodo **CaricaTabella** sull'oggetto **obj\_tabelle** già inizializzato.

Esistono due ulteriori metodi da poter applicare sull'oggetto di tipo **AvviaAppTabelle()** ovvero **objTabelle.ScaricaTabella("nometabella")** e **objTabelle.Reset()**. L'argomento **"nometabella"** nel metodo **ScaricaTabella()** deve essere il nome di una tabella master caricata nella pagina; ricordo che tale nome è quello indicato in **\$tabella["nometabella"]**; cioè l'etichetta che identifica la tabella master in **config.php** che non coincide necessariamente con il nome della tabella nel database. Il metodo **Reset()** applicato all'oggetto **AvviaAppTabelle** elimina tutte le tabelle master e detail caricate nella pagina.

## NOTE D'USO CON SCREENSHOOT ESPLICATIVI

Introduco qui alcuni dettagli sull'implementazione senza pretesa di essere esaustivo essendo numerosa la casistica dei messaggi di errori e ampia la personalizzazione della configurazione. Inizio con una panoramica sull'uso dei pulsanti per espandere le tabelle e per aggiungere o eliminare record.

The screenshot displays a web application interface with a table of records. The table has columns: VIEW, FULL, ID, Denominazione, Indirizzo, Telefono, Email, Foto, and An. A details popup is open, showing a table with columns: VIEW, FULL, ID, Codice stanza, Magazzino, Responsabile, and IF(magazzino, 'DEPOSITO', 'UFFICIO'). The popup also includes a 'NEW' button and a 'Tabella: STANZE' label. The main table shows records for 'Dipartimento di Beni Culturali' with various staff members and their roles.

VIEW	FULL	ID	Denominazione	Indirizzo	Telefono	Email	Foto	An
xml Excel	^ v	3	Dipartimento di Beni Culturali	Via Garibaldi	334444334	test@gmail.com		Note DBC
+	+	5	2233	<input type="checkbox"/>	Giorgio		UFFICIO	SUB: 1
+	+	7	644435	<input type="checkbox"/>	Giulio Verdi		UFFICIO	X
+	+	9	343434347	<input checked="" type="checkbox"/>	Simone Grigio		DEPOSITO	X

Nello screenshot sopra i pulsanti **+** e **-** nella colonna **VIEW** se cliccati servono a mostrare o chiudere la tabella details, non è necessario chiudere la tabella per aprirne altre, il clic del mouse su segno **+** chiude immediatamente le tabelle aperte non correlate per ridurre l'affollamento sullo schermo. I simboli **+** e **-** su sfondo giallo aprono la tabella verticale della riga desiderata. La tabella verticale mostra tutti i campi, non solo quelli indicati nel modello di tabella impostato in **config.php**, ma solo questi ultimi possono essere editati. L'aggiornamento dei campi avviene semplicemente inserendo/modificando i campi senza alcun bisogno di salvare il dato. L'aggiornamento è simultaneo sulle medesime tabelle, se modifichi un campo nella tabella verticale avverrà altrettanto nella corrispondente riga della stessa tabella tabulare e viceversa.

Per aggiungere nuovi record cliccare sul pulsante **NEW**, qualora per impostazione vi siano campi non editabili se consentita l'aggiunta il nuovo record permetterà comunque di modificarli.

L'**ordinamento** dei campi è disponibile con l'uso delle frecce **^** (A->Z) e inversa (Z->A). Il **filtro** sui campi opera cercando nel campo desiderato il valore contenuto nel campo filtro (tradotto in linguaggio SQL equivale a: WHERE campo like "%valorefiltro%"). La paginazione opera permettendo con i tasti **<** e **>** lo spostamento in successione pagina per pagina, e con i tasti **<<** e **>>** di andare nella prima e nell'ultima pagina, volendo si può inserire direttamente il valore numerico desiderato di pagina. Nell'ultima colonna di ciascuna tabella è possibile vedere se sono presenti righe corrispondenti nella tabella details (indicazione **SUB: NN**), qualora siano presenti non sarà possibile eliminare la riga se non si eliminano prima le righe nella tabella details.

La cancellazione dei record è notificata all'utente con un avviso tipo il sottostante:

The screenshot shows a confirmation dialog box titled 'CANCELLATO record:'. The dialog contains the following information: 'id: "10"', 'id\_dipartimento: "3"', 'piano: "0"', 'magazzino: "0"', and 'responsabile: ""'. There is an 'OK' button at the bottom right of the dialog. The background shows a table with records for 'Responsabile' and 'IF(magazzino, 'DEPOSITO', 'UFFICIO')'.

Una spiegazione ulteriore merita il campo di tipo **file** che come detto sopra permette l'upload di file di ogni tipo sul vostro server e di fare il preview delle immagini (qualunque formato supportato dal browser: gif, jpeg, png ecc...). Per gestire le operazioni di **upload/download/rimozione** dei file bisogna fare click sull'apposito pulsante nel campo, oppure sull'icona di preview in tabella dell'immagine che apre una nuova [finestra di tipo modale](#)



Il file sono salvati dentro una specifica cartella (tipo in esempio in **upload**), interna alla cartella d'installazione dell'applicazione. Nel caso si abbiano numerosi campi di tipo file è possibile riservare per ogni campo una cartella impostando la variabile **cartella\_upload** come detto sopra. Si potrebbero ad esempio creare cartelle per tipo di file (upload/pdf – upload/doc – upload/excel ecc... o per tipologia di campo). Non è possibile sovrascrivere file con lo stesso nome e si verrà avvisati nel caso si tentasse di farlo. Se i file sono presenti nella cartella designata ma non sono presenti nel campo del database (tipo text), oppure sono presenti nel campo del database ma non nel filesystem verranno eliminati dal campo e/o dal filesystem. Pertanto si raccomanda la dovuta attenzione nel caso di migrazione del database o di spostamento delle cartelle di upload di usare le seguenti precauzioni: aggiornare il nome della cartella di upload in **config.php** dopo ogni migrazione e il ripristino del database prima di avviare l'applicazione pena il rischio di perdita di dati e/o file (backup periodici e frequenti evitano tali problemi). Sotto un screenshot di esempio di avviso su quanto detto:

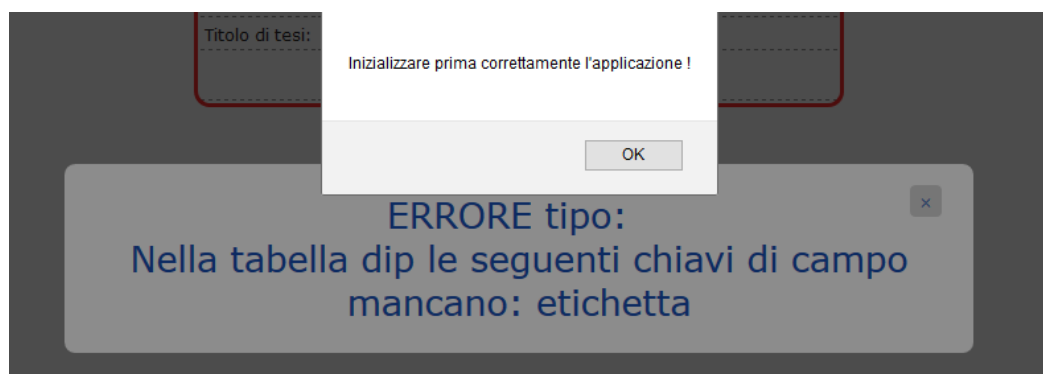


Di seguito sotto riporto uno screenshot che mostra come state implementate funzionalità quali il filtro, l'ordinamento ecc...:

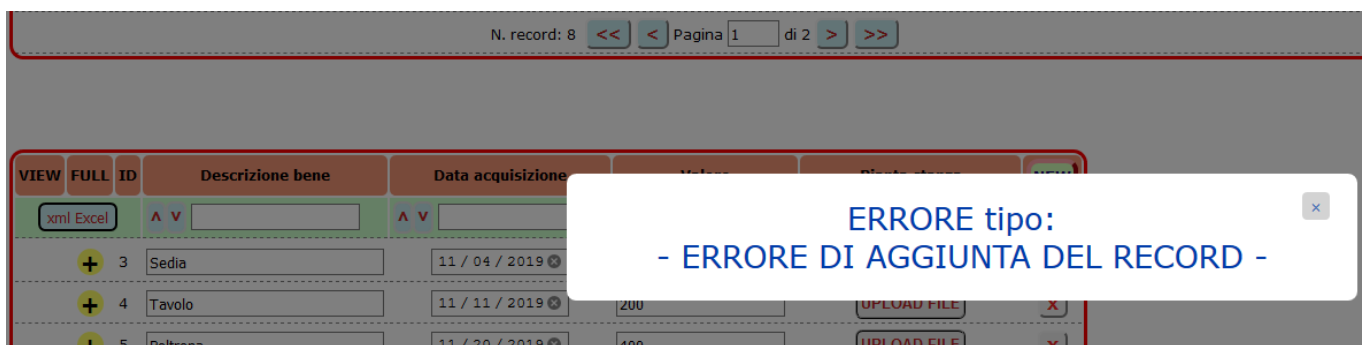


Il filtro di ricerca sul campo di tipo data è evidenziato dalla cerchiatura rossa sul campo medesimo e la data cercata è nel formato YYYY-MM-DD che trova le righe con la corrispondenza esatta. Si può immettere nel filtro anche solo una parte di essa, tipo solo l'anno (2019). Segnalo la differenza tra come è rappresentato il campo data nel controllo di input rispetto a come è realmente memorizzato nel database. La memorizzazione è conforme allo [standard ISO 8601](http://www.iso.org/standard/50067.html), mentre la rappresentazione nel browser dipende dalle impostazioni di localizzazione specifiche del browser che sono sotto il controllo delle preferenze dell'utente. Gli input nei controlli filtro, ordinamento e paginazione sono memorizzati e possono essere applicati simultaneamente; ovvero se ci si trova in una pagina specifica applicando il filtro o l'ordinamento si rimane nella stessa pagina.

Sotto uno screenshot che segnala la mancata inizializzazione dell'applicazione (costruttore **AvviaAppTabelle()** già visto sopra), compare nel messaggio la descrizione dell'errore, ovvero la mancata presenza (o storpiatura del nome) del campo etichetta nell'array \$tabella["dip"]. Tale tipo di messaggio compare ogniqualvolta l'uso dell'applicazione è inibito per un qualsiasi motivo, che va dal rilievo come in esempio di un'errata configurazione o la mancanza del collegamento di rete con il server, o la fase di attesa di aggiornamento dei record ecc... Lo scopo di avere una nutrita messaggistica è di non lasciare solo l'utente in balia di eventuali errori che possono manifestarsi.




Lo screenshot seguente mostra una condizione di errore che può verificarsi se non si presta la dovuta attenzione nel uso delle tabelle. Se si tenta di aggiungere un record nella tabella nel **database database-esempio.sql** definita **oggetti** nel caso usata nella demo in <http://wwwsic.chimica.unipd.it/michele.furlan> come tabella master si ha una [violazione dell'integrità referenziale](#); perché nel database ha una chiave esterna che non è stata valorizzata.



Sotto un esempio che mostra come sia stata implementata la possibilità di aggiungere record e di editarli anche quando i campi della tabella in **config.php** sono stati settati come non editabili. E' una possibilità che permette ad un utente di operare senza interferire sui record creati da altri utenti; un argomento più avanzato che mi riservo di sviscerare meglio insieme ad altri nel caso decidessi di ampliare questo manuale e/o l'applicazione.

La possibilità di creare tabelle multiple permette grazie ai campi calcolati di creare tabelle riepilogative usando formule e funzioni mysql. Sotto un esempio di tabella composta di soli campi calcolati che fa dei conteggi e con un campo somma valore che somma i valori contenuti nella colonna valore nella tabella oggetti del database di esempio:

FULL	ID	Descrizione bene	Data acquisizione	Valore	Pianta stanza	NEW
xls Excel						
+	2	Tavolo	14 / 11 / 2019	30	UPLOAD FILE	X
+	4	Armadio	11 / 11 / 2019	50	UPLOAD FILE	X
+	5	Cassettiera	11 / 11 / 2019	25		X
ID	Conta descrizione	Conta data	Somma valore	Conta foto	NO ADD	
2	3	3	105	3	SUB: NO SUB	

## CONDIZIONI LEGALI DI UTILIZZO

Il software **AppTabelle** è totalmente free, può essere liberamente modificato, copiato ed usato a fini personali o commerciali senza che nulla mi sia dovuto. Ringrazio in anticipo chi mi comunicherà il sito di eventuale utilizzo commerciale o vorrà segnalarmi la presenza di bachi e idee per correzioni ed integrazioni, al fine di migliorarlo.

Potete contattarmi via email scrivendo a [michele.furlan@unipd.it](mailto:michele.furlan@unipd.it)