

## Ricerca Testuale Allarmi Storici

### Introduzione

Questa TN descrive le operazioni da seguire per fare la ricerca testuale degli allarmi storici in un'applicazione Intouch (Standalone e Managed).

### Versioni

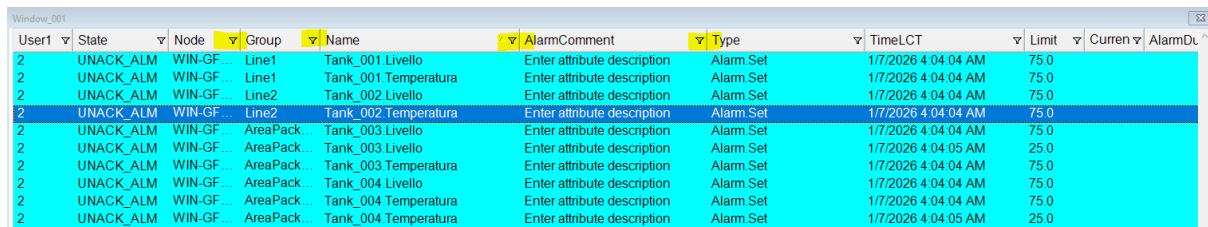
Quanto descritto in questa TN è stato verificato sulle seguenti versioni:

- Dalla 2023 in avanti

# Ricerca Testuale Allarmi Storici

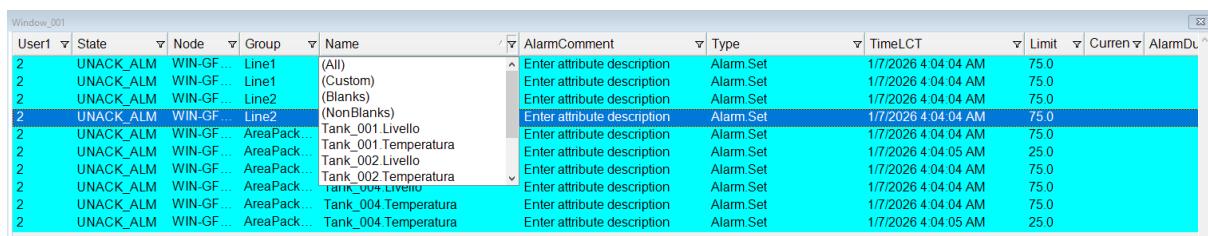
## Metodo 1 (nativo): Tramite Alarm Client

- Una volta inserito l'alarm client configurato per gli allarmi storici in un simbolo archestra e avviato il runtime, cliccare sul simbolo del filtro presente in tutte le colonne dell'alarm control

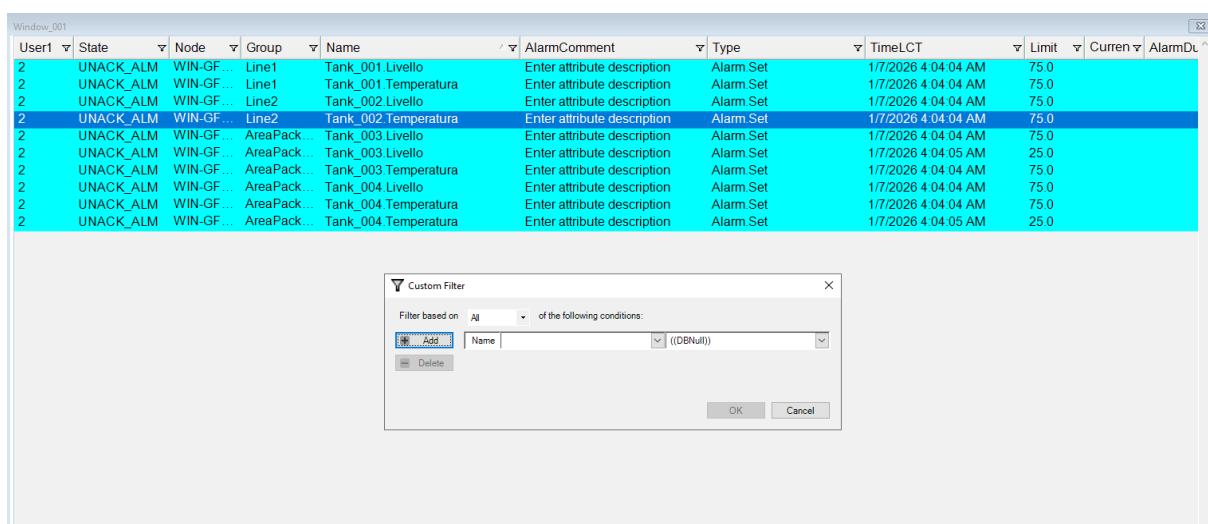


The screenshot shows a table titled "Window\_001" with columns: User1, State, Node, Group, Name, AlarmComment, Type, TimeLCT, Limit, Current, and AlarmDt. The data consists of multiple rows of historical alarm entries, such as "UNACK\_ALM" for various nodes like "WIN-GF..." and "AreaPack...", with names like "Line1", "Line2", "Tank\_001 Livello", etc. The "Type" column shows "Alarm Set" for most entries.

- Cliccando uscirà un menù a tendina



- Cliccare su (Custom)
- Apparirà una schermata di configurazione



## 5. Selezionare "Contains" e inserire la ricerca desiderata

The screenshot shows a software interface with a table titled 'Window\_001' displaying alarm history. The columns include User, State, Node, Group, Name, AlarmComment, Type, TimeLCT, Limit, Current, and AlarmD. Below the table is a 'Custom Filter' dialog box with the following settings:

- Filter based on: All
- Condition: Contains
- Value: Livello

### Metodo 2: Tramite Script

- Nel simbolo in cui è presente l'alarm client creare 2 custom property stringa vuote, denominate cpText e cpFullQuery

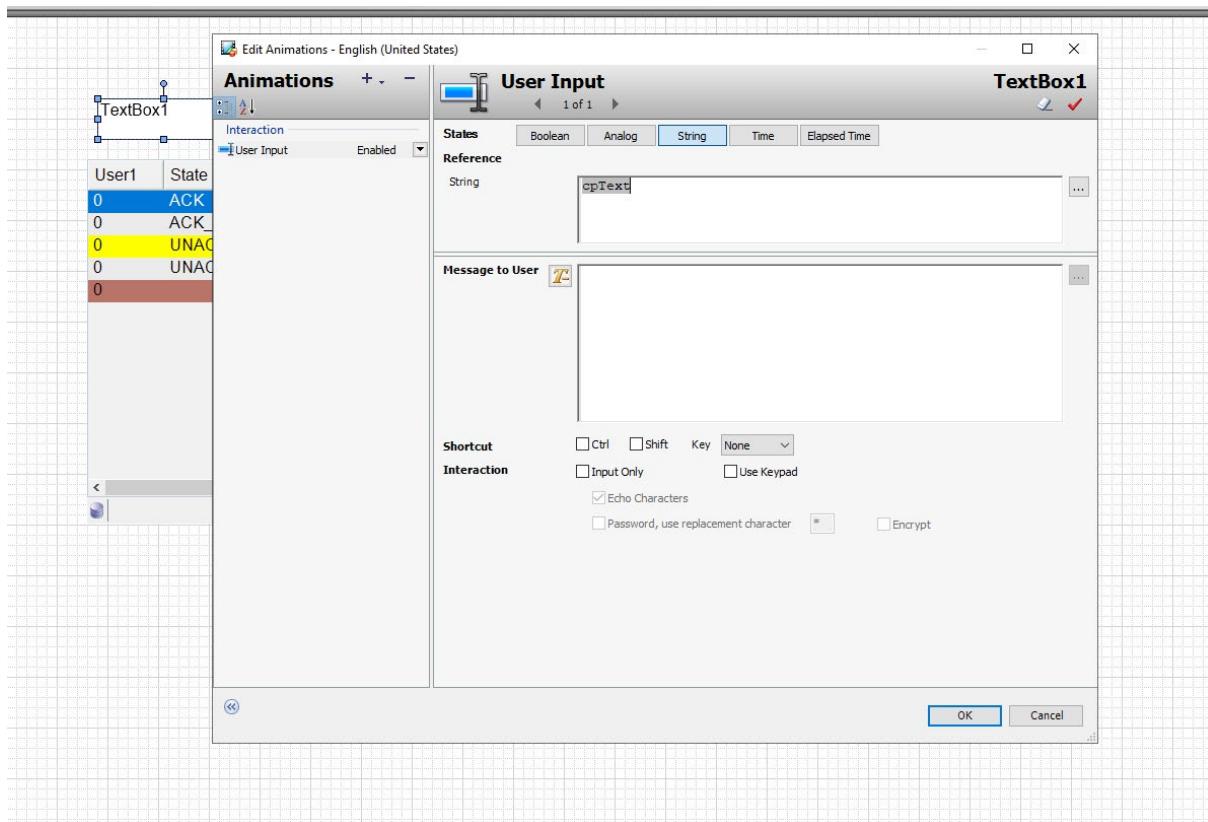
The screenshot shows the 'Edit Custom Properties - English (United States)' dialog box. On the left, there is a table of custom properties:

User1	State
0	ACK
0	ACK_R
0	UNACK
0	UNACK

A new property 'cpFullQuery' is being edited in the main pane. The properties are:

- Data Type: String
- Default Value: (empty)
- Visibility: Public (Property can be seen when symbol is embedded) (radio button selected)
- Description: (empty)

2. Creare una text box
3. Alla text box appena creata, applicare l'animazione user input e inserire come referenza la custom property "cpText"



4. Creare un bottone
5. Per filtrare ad esempio la colonna Name (ma vale per qualsiasi colonna), al bottone appena creato applicare l'animazione "Action Script" e incollare il seguente script:

```
dim baseQuery as string;
baseQuery = "";
cpFullQuery = baseQuery + "Name Like '%" + cpText + "%'";
AlarmClient1.QueryFilters.DefaultFilter.FilterCriteria = cpFullQuery;
'reset necessario per sicurezza solo se possono essere impostate
altre query favorite
AlarmClient1.Favorite = "Default";
'applica
AlarmClient1.Requery();
```

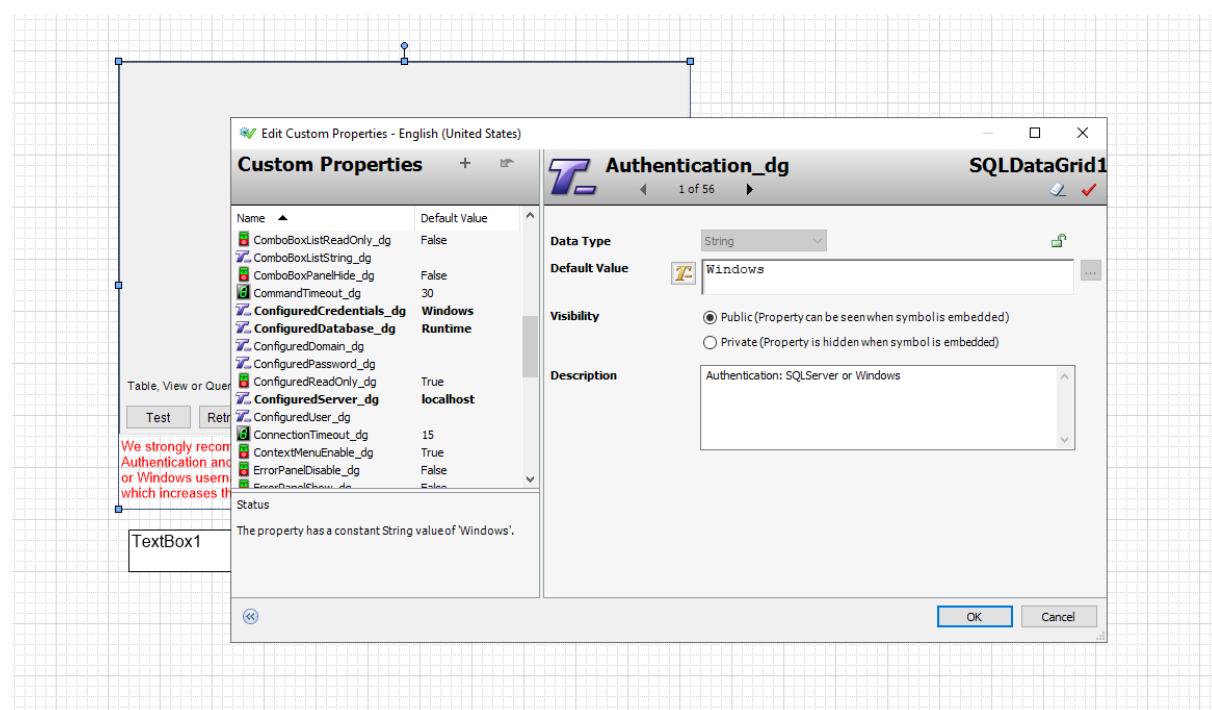
6. Salvare e Avviare il runtime
7. Scrivere nella text box il nome della tag da cercare o anche solo le iniziali
8. Cliccando sul button, verrà rieseguita una *requery* e l'alarm client verrà popolato filtrato per la ricerca effettuata.

### Metodo 3: Tramite SQLDataGrid

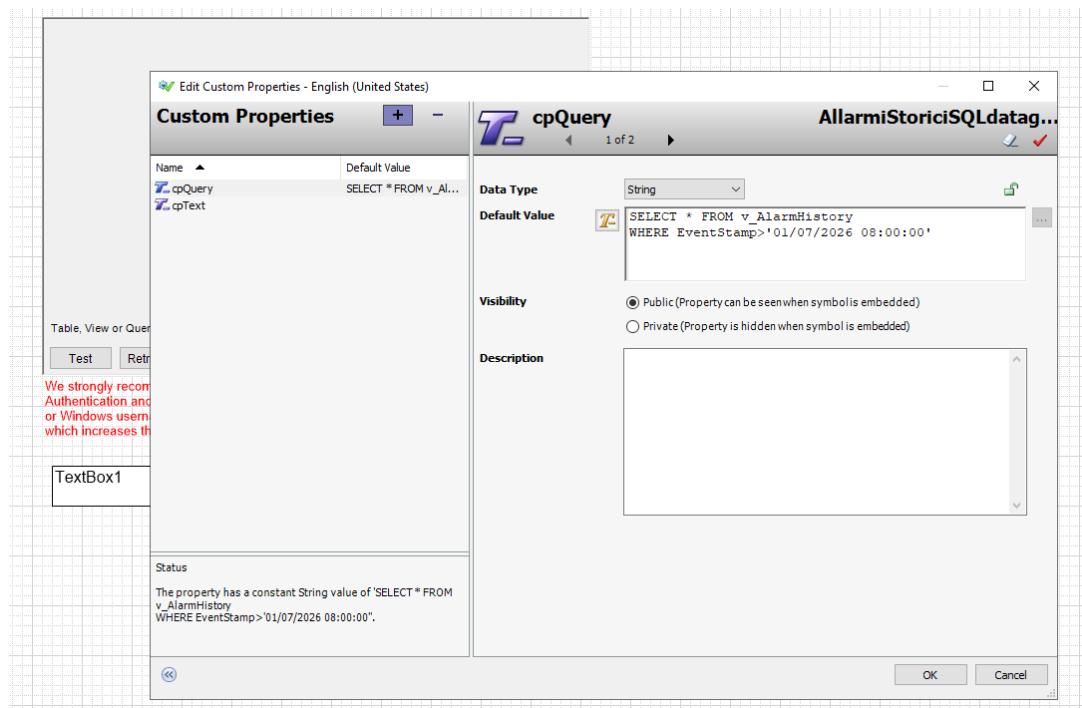
1. Creare un simbolo Archestra
2. Inserire all'interno del simbolo l'oggetto SQLDataGrid
3. Configurare la connessione al DB aprendo le custom property dall'oggetto.

Se si utilizza la windows authentication basta configurare "ConfiguredCredential\_dg" digitando direttamente la parola "Windows", poi nella parte di "ConfiguredDatabase\_dg" inserire il nome del database e nella parte di "ConfiguredServer\_Dg" in che server si trova inserendo il nome macchina (se in locale potete inserire "localhost").

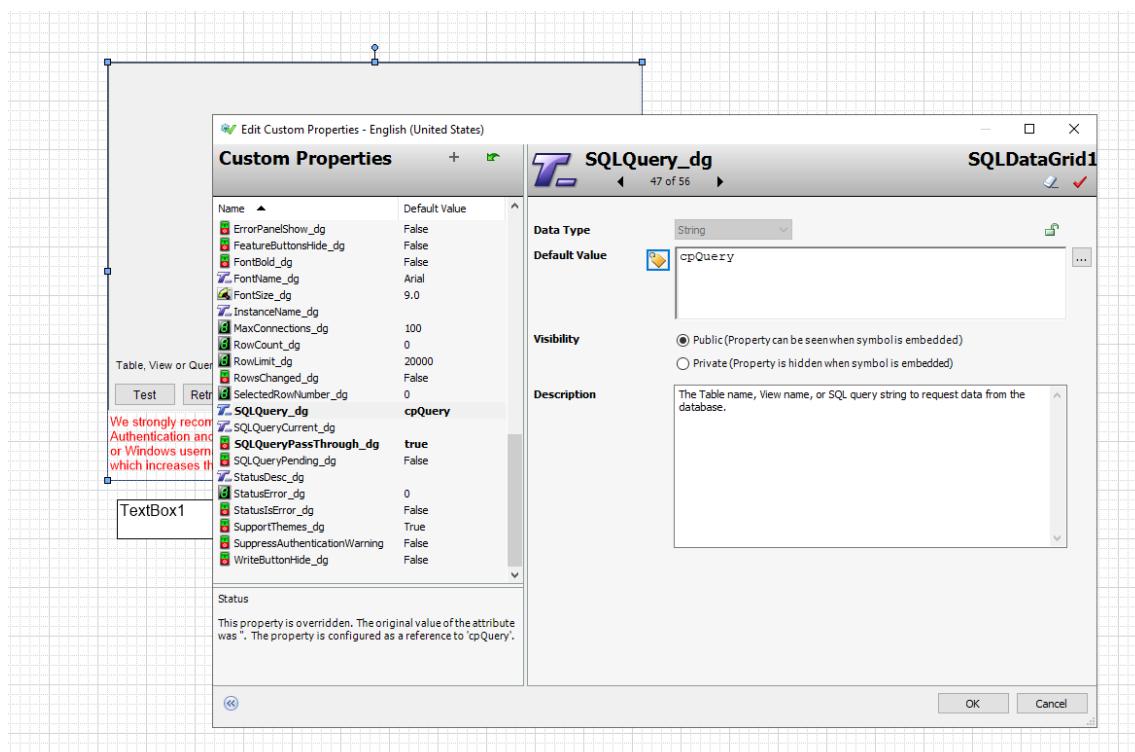
In caso non si utilizzasse la windows authentication, nella parte di "ConfiguredCredential\_dg" inserire il tipo di autenticazione e successivamente compilare i campi "ConfiguredDomain\_dg", "ConfiguredPassword\_dg", "ConfiguredUser\_dg".



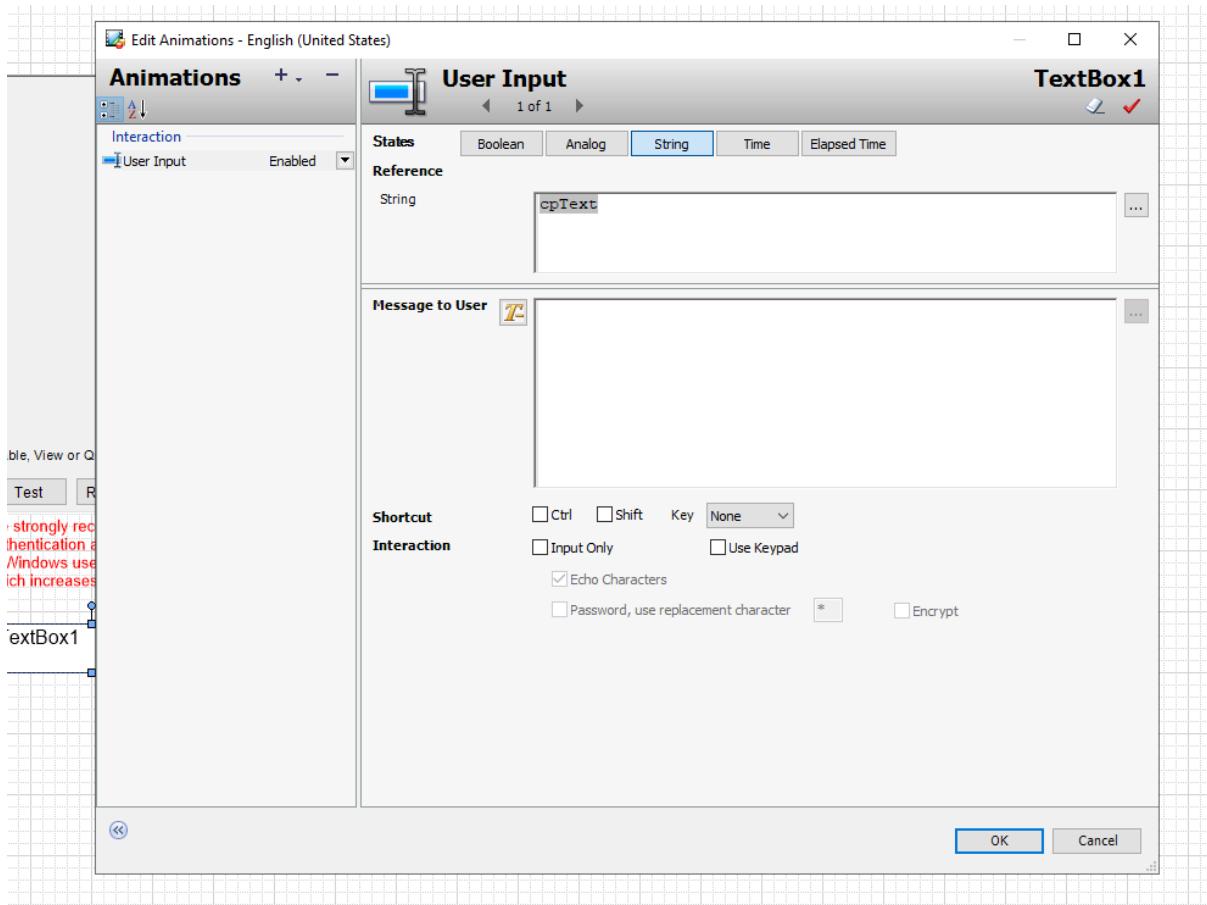
4. Creare due custom property stringa, la prima denominata cpText e si lascia vuota, la seconda denominata cpQuery e al suo interno scriviamo la query che desideriamo fare verso il nostro database.



5. Riaprire il custom property dell'oggetto SQLDataGrid.
6. Andare sulla proprietà "SQLQuery\_dg" e inserire cpQuery
7. Andare sulla proprietà "SQLQueryPassThrough\_dg" e scrivere true.



8. Creare una text box
9. Aprire le animazioni della text box e aggiungere l'animazione "User Input"
10. Cliccare su String e all'interno della referenza inserire cpText

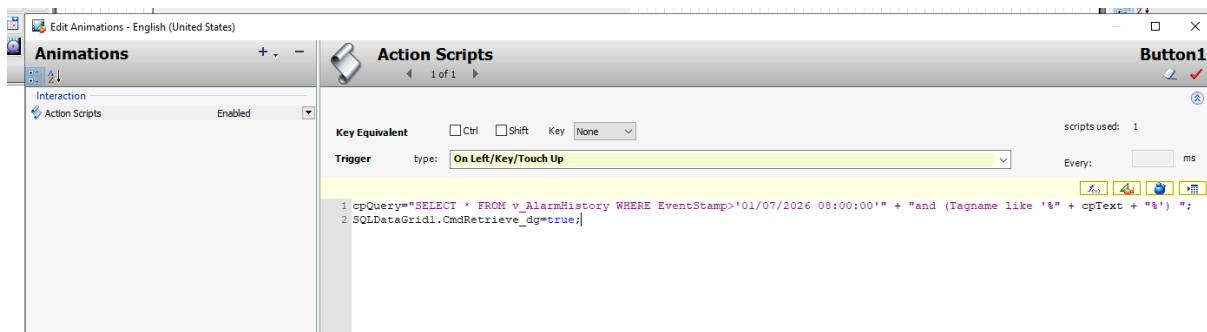


11. Creare un button

12. Inserire l'animazione Action Script e incollare le seguenti linee di codice:

```
cpQuery = "SELECT * FROM v_AlarmHistory2 WHERE EventStamp >= '01/07/2026 08:00:00'" + "and (Tagname like '%" + cpText + "%') ";
```

SQLDataGrid1.CmdRetrieve\_dg=true;



13. Salvare e chiudere il simbolo.
14. Avviare il runtime
15. Cliccando retrive usciranno tutti gli allarmi, mentre se si inserisce una chiave di ricerca all'interno della text box e si clicca sul button da noi creato precedentemente, verrà popolata la griglia con la ricerca desiderata.

## Referenze

AVEVA™ Alarm Client Control:

<https://docs.aveva.com/bundle/sp-alarm/page/1441216.html>

AVEVA™ SQLDataGrid Graphic:

<https://docs.aveva.com/bundle/sp-appserver/page/385863.html>

Il presente documento è fornito a scopo di esempio e non sostituisce la documentazione AVEVA. L'applicazione di quanto contenuto, in un preciso ambito applicativo, deve essere sempre validata da un tecnico Factory Software. La documentazione rilasciata da AVEVA resta il riferimento tecnico ufficiale da seguire: [AVEVA | Customer Support Portal](#).