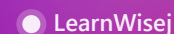


Finestre di dialogo, validazione e flussi di lavoro

Finestre di dialogo e flussi modali · Validazione e il pattern Salva/Annulla



Finestre di dialogo e flussi modali

Che cos'è una finestra di dialogo, quando usarla al posto di una pagina, il flusso modale in cinque passi e dove aiutano i message box.



Obiettivo della lezione

Capisci le **finestre di dialogo** e il flusso modale di aggiunta/modifica su cui si basano le app gestionali: apri una finestra mirata, lascia che l'utente completi un'unica attività, chiedi conferma per le azioni rischiose e restituisci un risultato chiaro.

1. Che cos'è una finestra di dialogo?

Una **finestra di dialogo** è una schermata più piccola che compare sopra la pagina principale, così che l'utente possa completare un'unica attività mirata. In questo corso le finestre di dialogo più comuni sono **Create Ticket** ed **Edit Ticket**. Una finestra di dialogo non dovrebbe cercare di essere l'intera applicazione — raccoglie i valori necessari, li valida e restituisce un risultato chiaro.



Idea chiave

Una finestra di dialogo è *modale*: l'utente la completa o la annulla prima di tornare alla pagina sottostante. È questa concentrazione a rendere prevedibili i flussi di aggiunta/modifica.

2. Finestra di dialogo e pagina a confronto

Tipo di schermata	Uso migliore	Esempio
Pagina / area contenuti principale	Grandi schermate dell'app in cui gli utenti passano tempo e navigano tra le funzionalità.	Pagina Tickets, pagina Customers, pagina Settings.
Finestra di dialogo / finestra modale	Un piccolo flusso mirato che va completato o annullato prima di tornare indietro.	Finestra Create Ticket, finestra Edit Ticket, messaggio Confirm Delete.

3. Il pattern comune del flusso modale

Quasi ogni funzionalità di aggiunta/modifica segue lo stesso flusso in cinque passi. Rispettarlo rende il flusso leggibile e prevedibile.

Click sul pulsante → l'utente sceglie New o Edit
Apri la finestra → mostra il form / la finestra modale
Valida → controlla i campi obbligatori
Salva → chiama il livello di servizio
Aggiorna la UI → aggiorna la griglia e mostra un feedback

L'utente fa clic su **New** o **Edit**, l'app apre una finestra di dialogo, la finestra valida i campi, il pulsante **Save** chiama un servizio e la pagina principale si aggiorna *dopo* che la finestra si è chiusa con successo.

4. Message box e conferme

Usa i message box quando l'utente ha bisogno di un feedback immediato o deve confermare un'azione rischiosa.

Situazione	Feedback consigliato
Salvataggio completato	Mostra un breve messaggio di conferma e aggiorna l'elenco dei ticket.
Campo obbligatorio mancante	Mostra un messaggio di validazione chiaro vicino al campo o in cima alla finestra di dialogo.
Eliminazione del ticket selezionato	Chiedi conferma prima di eliminare.
Annullamento dopo aver cambiato dei valori	Facoltativo: avvisa che le modifiche non salvate andranno perse.



Sei pronto per la prossima lettura

Sai descrivere che cos'è una finestra di dialogo, quando usarla al posto di una pagina, il flusso modale in cinque passi e dove aiutano i message box. Prossimo passo: la validazione dell'input e il pattern Salva/Annulla coerente.

Validazione e il pattern Salva/Annulla

Valida vicino all'interfaccia senza intrappolarci le regole di business, scrivi un gestore Save leggibile e applica un pattern Salva/Annulla coerente.



Obiettivo della lezione

Valida l'input vicino all'interfaccia senza intrappolarci le regole di business, scrivi un gestore Save leggibile e applica un pattern Salva/Annulla coerente, così gli utenti sanno sempre che cosa è successo.

1. Validazione: vicina all'interfaccia, ma non intrappolata lì

La validazione deve stare abbastanza vicina all'interfaccia da dare agli utenti un feedback immediato, ma le regole di business importanti devono comunque restare testabili fuori dal click del pulsante. La finestra di dialogo controlla i semplici problemi dei campi, poi chiama una classe di servizio per le regole riutilizzabili.

Tipo di validazione	Esempio	Posto giusto
Validazione semplice dell'interfaccia	La textbox del titolo è vuota; la priorità non è selezionata.	Un metodo della finestra di dialogo come <code>ValidateForm()</code> .
Regola di business	Un ticket chiuso non può essere modificato da un support agent.	<code>TicketService</code> o un servizio di validazione.
Regola di persistenza	L'ID del ticket esiste già nell'archivio.	Il livello repository / servizio.

2. Gestione sicura degli eventi

Gli eventi click dei pulsanti devono essere leggibili. Non nascondere chiamate lente al database o logica di business complicata dentro un banale evento dell'interfaccia, a meno che non sia intenzionale e con un nome chiaro. Un buon evento Save chiama piccoli helper come `ValidateForm()`, `BuildTicketFromFields()` e `ticketService.SaveTicket(ticket)`.

```
private void btnSave_Click(object sender, EventArgs e)
{
    if (!ValidateForm())
        return;

    Ticket ticket = BuildTicketFromFields();
    ticketService.SaveTicket(ticket);

    this.DialogResult = DialogResult.OK;
    this.Close();
}
```



Valida prima di salvare

Il gestore esce subito quando `ValidateForm()` fallisce — non viene salvato nulla e la finestra di dialogo resta aperta. Solo l'input valido arriva al servizio.

3. Un pattern Salva / Annulla coerente

- **Save** prima valida, poi chiama il servizio, poi chiude la finestra di dialogo *solo dopo il successo*.
- **Cancel** chiude la finestra di dialogo senza salvare.
- I messaggi di errore devono dire all'utente esattamente che cosa correggere.
- La pagina padre aggiorna la griglia solo quando la finestra di dialogo restituisce `OK`.
- Usa nomi chiari per i pulsanti: `btnSave`, `btnCancel`, `btnNew`, `btnEdit`.

4. Abitudini da sviluppatore per questo modulo

- Dai nomi chiari ai controlli, così il codice degli eventi è leggibile.
- Mantieni il layout della finestra di dialogo semplice e mirato.
- Separa lo stato dell'interfaccia, le regole di business e i dati salvati.
- Usa i servizi per le operazioni riutilizzabili sui ticket.
- Usa message box e messaggi di validazione in modo coerente.



Sei pronto per il lab

Prossimo passo: guarda la videoguia sulle finestre di dialogo, supera la verifica delle conoscenze, poi costruisci le finestre Create ed Edit Ticket riutilizzabili — prima valida, salva tramite il servizio, restituisci `DialogResult.OK` e aggiorna la griglia con un feedback chiaro.