JE3 - I form

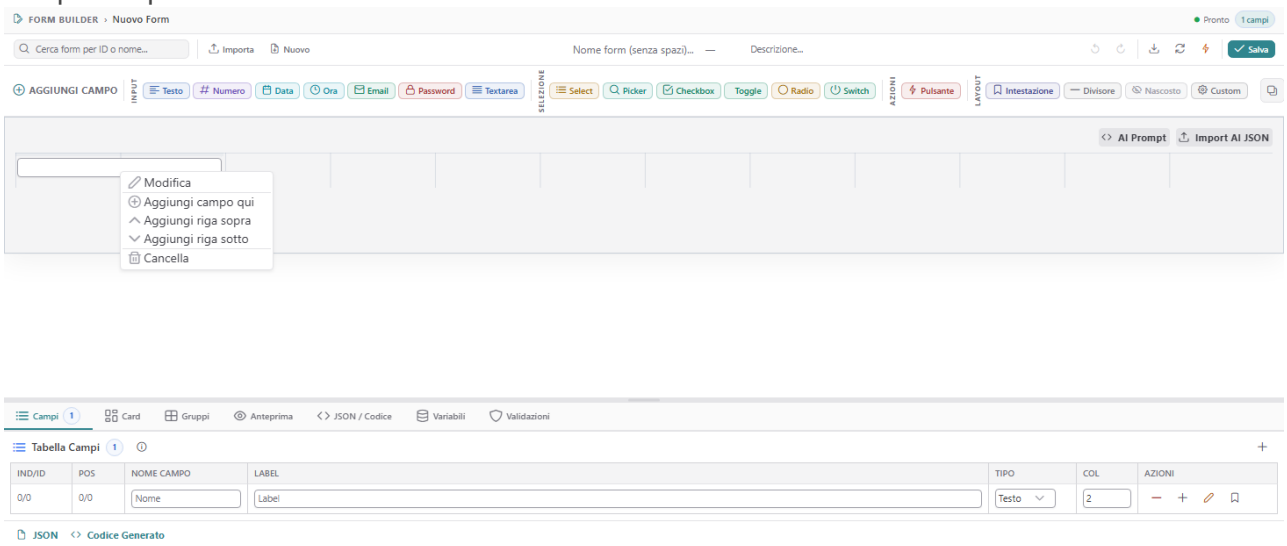
- JE3.1 - Disegnare un form
- JE3.2 - Regole e controlli
- JE3.3 - Scrivere valori e formule

JE3.1 - Disegnare un form

In una frase. Un form è la maschera dove l'utente scrive: la componi scegliendo i campi e mettendoli su una griglia, e la pagina vera esce uguale a come l'hai disegnata.

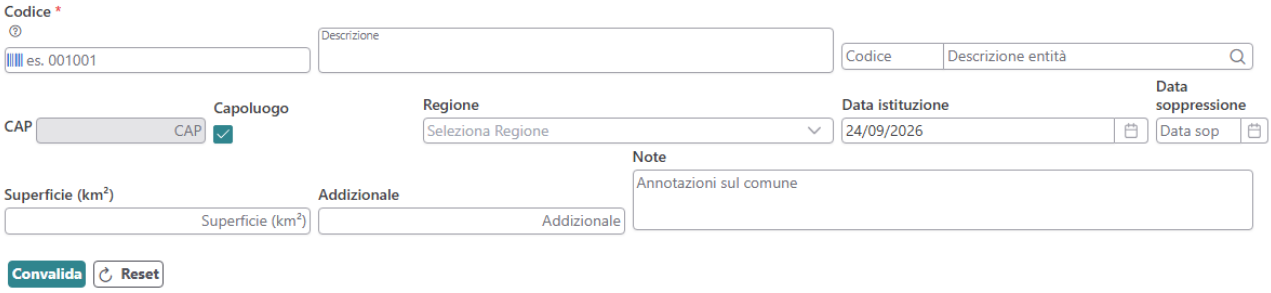
Un form disegnato non è un bozzetto da passare a un programmatore. È già la maschera che l'utente userà. Per questo il risultato si vede subito, e una modifica costa pochi minuti.

1. **Apri il Form Builder.** Dal menu: **Low-Code Platform › Dynamic Forms › Form Builder**. Puoi partire da un modello pronto (persona, azienda, veicolo, filtro di ricerca) o **Da zero**.
2. **Aggiungi i campi.** La barra in alto ha i tipi di campo, in quattro famiglie: testo, numero, data, scelta da un elenco, casella da spuntare, pulsante, titolo e altri. Un clic e il campo va nel primo posto libero.



Il Form Builder: in alto i tipi di campo, al centro la griglia, sotto l'elenco dei campi

3. **Mettili in ordine.** La griglia ha dodici colonne. Trascini un campo per spostarlo, tiri l'angolo per allargarlo. Con il tasto destro aggiungi una riga sopra o sotto. Ogni gesto si annulla con **Annulla**.
4. **Dai un nome a ogni campo.** Doppio clic sul campo: si apre il pannello a destra. Lì scrivi nome, etichetta, se è obbligatorio, il testo di aiuto, il valore con cui nasce.
5. **Raggruppa, se il form è lungo.** I campi si chiudono in riquadri con un titolo, oppure in schede o in passi con **Precedente** e **Successivo**.
6. **Guarda l'anteprima e salva.** La scheda **Anteprima** mostra il form vero. **Genera dati di test** lo riempie per provarlo. Poi **Salva**.



Lo stesso form nella pagina: campi, asterischi e aiuti sono quelli scelti nel disegno

Cosa ci guadagni. Per scegliere un valore collegato, come una provincia, c'è un tipo di campo già pronto: una finestra di ricerca con codice e descrizione. Non va costruita ogni volta.

Per chi sviluppa. Il nome del form non ammette spazi: è la chiave con cui viste e pagine lo cercano. Il disegno è in `frontend/ade-next/src/studio/form-builder`, il form a schermo in `frontend/ade-next/src/runtime/form`. Le funzioni che usano l'intelligenza artificiale e l'importazione di una maschera storica ci sono, ma la guida non le ha provate.

Per approfondire.

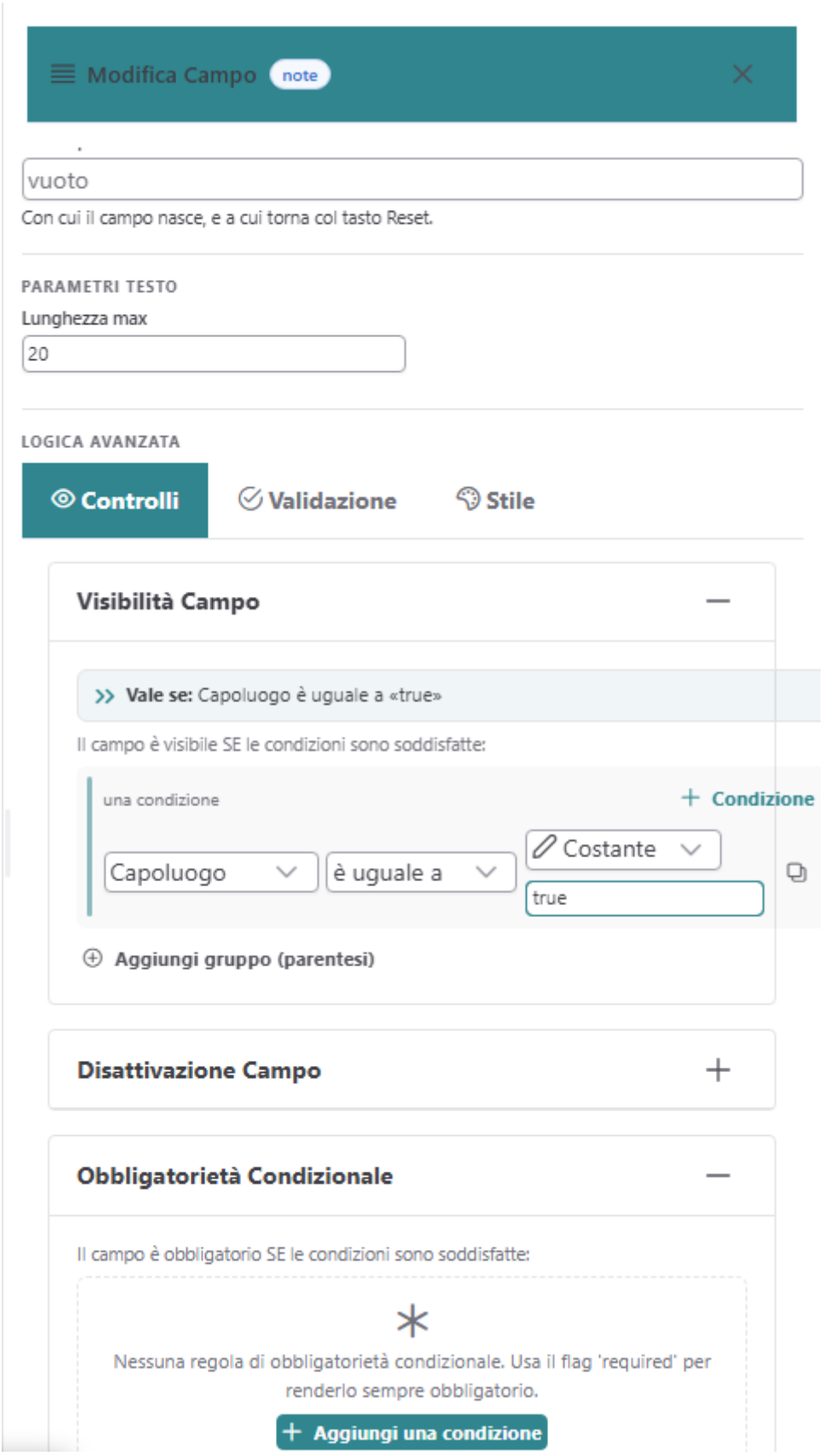
- Il Form Builder — disegnare un form e ritrovarlo uguale a runtime: `docs/guide/form/builder/README.md`

JE3.2 - Regole e controlli

In una frase. Le regole di un form si scelgono da un pannello, senza scrivere codice, e valgono due volte: mentre l'utente scrive e quando il server salva.

Un controllo che vive solo nella pagina si può aggirare, e uno che vive solo nel server si scopre troppo tardi. Qui la regola è una sola, scritta nel disegno del form. La leggono tutti e due, e danno lo stesso messaggio.

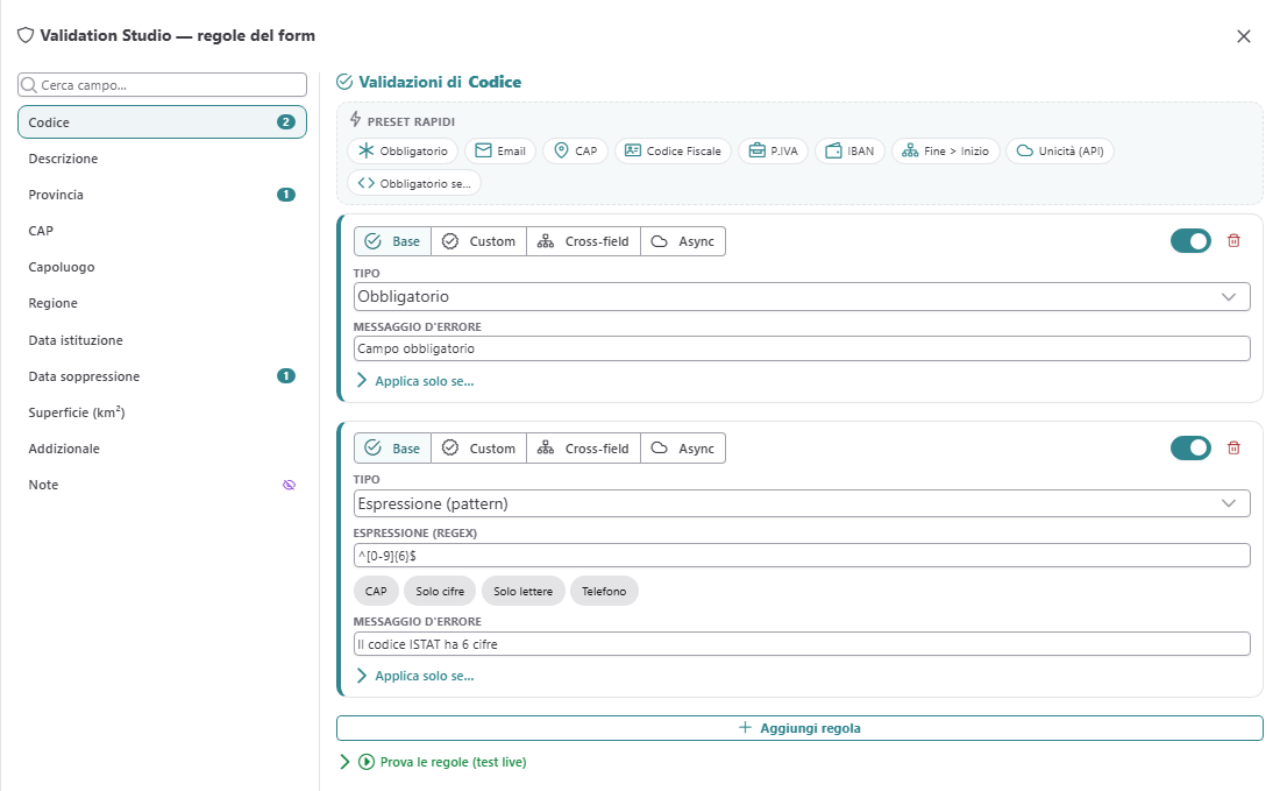
1. **Apri il campo.** Nel Form Builder, doppio clic sul campo. In fondo al pannello ci sono tre schede: **Controlli**, **Validazione**, **Stile**.
2. **Decidi quando il campo si vede.** Nella scheda **Controlli** componi la condizione scegliendo da tre tendine: un campo, un confronto, un valore. Sopra, la regola si rilegge in italiano. Allo stesso modo decidi quando il campo è bloccato e quando diventa obbligatorio.



La scheda Controlli: il campo Note si vede solo se Capoluogo è spuntato

3. **Aggiungi i controlli sul valore.** Nella scheda **Validazione** trovi quelli pronti: obbligatorio, email, CAP, codice fiscale, partita IVA, IBAN, «la fine dopo l'inizio», valore non già usato. Per ogni regola scrivi il messaggio che l'utente leggerà.
4. **Guardale tutte insieme.** La scheda **Validazioni** in fondo alla pagina apre l'elenco di tutti i campi con le loro regole. Ogni regola ha un interruttore: spenta, resta scritta ma non

vale.



Tutte le regole del form: a sinistra i campi, a destra le regole del campo scelto

5. **Prova.** Nella pagina, **Convalida** non è mai spento: se manca qualcosa dice cosa. Un campo nascosto non viene preteso.

> **Cosa ci guadagni.** Se un altro programma prova a salvare dati sbagliati saltando la pagina, il server li rifiuta con lo stesso messaggio che avrebbe visto l'utente.

> **Per chi sviluppa.** I due motori gemelli sono `frontend/ade-next/src/kernel/rule-engine` e, nel servizio, il pacchetto `validation`. Un file di casi in comune li tiene allineati. Un salvataggio rifiutato torna come errore 422.

> **Per approfondire.**
> • Il Form Builder — disegnare un form e ritrovarlo uguale a runtime: `docs/guide/form/builder/README.md`
> • Guida sviluppo — Validazioni, nascondimenti e picker delle relazioni: `docs/guide/sviluppo/validazioni-nascondimenti-e-picker.md`

JE3.3 - Scrivere valori e formule

In una frase. Ovunque serva un valore c'è la stessa casella con il pulsante **fx**: ti propone cosa puoi usare in quel punto e le funzioni pronte.

Quando disegni capita spesso di dover dire «prendi il valore da lì»: la provincia scritta nel form, il risultato del passo precedente, la data di oggi più trenta giorni. Scriverlo a memoria porta a errori che si vedono solo dopo. Il menu **fx** lo fa scegliere.

1. **Premi fx.** Lo trovi accanto alle caselle dei valori: nella condizione di un bivio, nei parametri di un endpoint, in «Esegui solo se».
2. **Scegli da dove viene il valore.** La prima parte del menu elenca quello che esiste in quel punto: i campi del form, i risultati dei passi precedenti, la riga corrente se sei dentro un ciclo. Quello che non esiste non compare.



Il menu fx: sopra i valori disponibili in quel punto, sotto le funzioni con un esempio

3. **Oppure cerca una funzione.** Le funzioni sono una trentina: testo, numeri, date, formato, logica. Cerchi quello che vuoi fare, non il nome: scrivendo «giorni» trovi la funzione che aggiunge giorni a una data. Ognuna mostra un esempio con il risultato.

4. **Continua a scrivere.** La scelta entra dove avevi il cursore. Puoi comporre: una funzione attorno a un valore, due valori confrontati.
5. **Leggi l'errore, se c'è.** La formula viene controllata mentre scrivi. Se non si può leggere, sotto la casella compare cosa non va e in che punto.

> **Cosa ci guadagni.** Chi sa già cosa scrivere scrive, chi non lo sa sceglie. In tutti e due i casi il controllo è lo stesso che userà il motore quando esegue.

> **Per chi sviluppa.** I valori si scrivono fra doppie graffe, per esempio `{{in.Provincia}}`. Il motore delle formule è proprio del sistema (`frontend/ade-next/src/kernel/expression-engine`): non esegue codice libero. Gli esempi del menu sono eseguiti da un test.

> **Per approfondire.**
> • Scrivere valori — il controllo unico e le funzioni: `docs/guide/variabili/value-editor/README.md`