

REPORT TECNICO

Banca Dati Complessa "CSR Regionali" — Complementi per lo Sviluppo Rurale 2023-2027

Autore e responsabile: Mario Cariello

Coautori: Pietro Manzoni, Clelia Staibano, Laura Starna

Versione documento: 2.0 (estesa)

Data emissione: gennaio 2025

Stato: Documento tecnico ufficiale

1. Premessa e inquadramento normativo

1.1 La nuova PAC 2023-2027

La Politica Agricola Comune per il periodo 2023-2027 è stata riformata in modo sostanziale rispetto al ciclo precedente. Il nuovo quadro normativo è basato su:

- **Regolamento (UE) 2021/2115** — piani strategici della PAC
- **Regolamento (UE) 2021/2116** — finanziamento, gestione e monitoraggio della PAC
- **Regolamento (UE) 2021/2117** — organizzazione comune dei mercati (OCM)

La novità principale è il **Piano Strategico Nazionale (PSN)**, un unico documento che sostituisce i precedenti Programmi di Sviluppo Rurale e i pagamenti diretti.

1.2 I CSR come strumento di attuazione

In Italia, il PSN viene attuato a livello regionale attraverso i **Complementi per lo Sviluppo Rurale (CSR)**. Ogni Regione/PA definisce il proprio CSR, che declina:

- **Interventi** (ex misure) finanziati
- **Dotazioni finanziarie** per ciascun intervento
- **Criteri di ammissibilità**
- **Target e obiettivi specifici**
- **Modalità di attuazione**

I CSR sono **negoziati con la Commissione Europea** e approvati con Decisione di esecuzione. Ogni modifica sostanziale richiede una nuova versione.

1.3 La fase di transizione

Il passaggio da PSR a CSR ha comportato:

- **Nuova architettura** — gli interventi sostituiscono le misure
- **Nuova numerazione** — le versioni seguono logiche diverse per ogni Regione

- **Nuove tempistiche** — ogni Regione procede con ritmi propri
- **Nuove priorità** — il Green Deal europeo e la strategia Farm to Fork

Questa transizione rende la banca dati CSR **particolarmente dinamica e complessa** da mantenere aggiornata.

2. Architettura generale del sistema

2.1 Modello dati

Il sistema è basato su un **indice strutturato in JavaScript** (oggetto CSR_INDEX) organizzato per **codici regionali normalizzati**, identici a quelli della banca dati PSR per garantire **coerenza e interoperabilità tra i due archivi**.

2.2 Struttura dei record

Ogni record contiene:

- Campo: `version` — Tipo: `number/string` — Descrizione: Numero di versione (es. 1, 2.1, 5, 8)
- Campo: `date` — Tipo: `string (ISO 8601)` — Descrizione: Data di pubblicazione (YYYY-MM-DD)
- Campo: `url` — Tipo: `string` — Descrizione: Collegamento diretto al PDF
- Campo: `title` — Tipo: `string` — Descrizione: Denominazione del Complemento
- Campo: `period` — Tipo: `string` — Descrizione: Periodo di programmazione (2023-2027)

2.3 Volume di dati

La banca dati complessa gestisce attualmente **oltre 100 record documentali**, distribuiti su 21 entità territoriali:

- Regione/PA: Abruzzo — Codice: ABR — Numero versioni: 4 — Prima versione: 2022-12-29 — Ultima versione: 2025-10-21
- Regione/PA: Basilicata — Codice: BAS — Numero versioni: 6 — Prima versione: 2022-12-30 — Ultima versione: 2026-02-11
- Regione/PA: Calabria — Codice: CAL — Numero versioni: 1 — Prima versione: 2022-12-28 — Ultima versione: 2022-12-28
- Regione/PA: Campania — Codice: CAM — Numero versioni: 8 — Prima versione: 2022-12-02 — Ultima versione: 2026-05-15
- Regione/PA: Emilia-Romagna — Codice: EMR — Numero versioni: 2 — Prima versione: 2022-09-28 — Ultima versione: 2025-12-15
- Regione/PA: Friuli V.G. — Codice: FVG — Numero versioni: 8 — Prima versione: 2023-01-13 — Ultima versione: 2026-08-07
- Regione/PA: Lazio — Codice: LAZ — Numero versioni: 4 — Prima versione: 2023-01-12 — Ultima versione: 2026-01-15

- Regione/PA: Liguria — Codice: LIG — Numero versioni: 2 — Prima versione: 2023-06-01 — Ultima versione: 2023-10-31
- Regione/PA: Lombardia — Codice: LOM — Numero versioni: 6 — Prima versione: 2022-11-21 — Ultima versione: 2026-05-04
- Regione/PA: Marche — Codice: MAR — Numero versioni: 5 — Prima versione: 2023-08-01 — Ultima versione: 2026-02-11
- Regione/PA: Molise — Codice: MOL — Numero versioni: 3 — Prima versione: 2023-03-06 — Ultima versione: 2026-03-06
- Regione/PA: Piemonte — Codice: PIE — Numero versioni: 5 — Prima versione: 2023-02-16 — Ultima versione: 2026-05-04
- Regione/PA: P.A. Bolzano — Codice: PABZ — Numero versioni: 7 — Prima versione: 2023-01-16 — Ultima versione: 2026-03-06
- Regione/PA: P.A. Trento — Codice: PATN — Numero versioni: 5 — Prima versione: 2023-02-17 — Ultima versione: 2026-01-30
- Regione/PA: Puglia — Codice: PUG — Numero versioni: 7 — Prima versione: 2022-11-29 — Ultima versione: 2026-07-06
- Regione/PA: Sardegna — Codice: SAR — Numero versioni: 8 — Prima versione: 2023-04-13 — Ultima versione: 2026-07-22
- Regione/PA: Sicilia — Codice: SIC — Numero versioni: 2 — Prima versione: 2023-03-30 — Ultima versione: 2024-09-02
- Regione/PA: Toscana — Codice: TOS — Numero versioni: 6 — Prima versione: 2022-12-27 — Ultima versione: 2025-07-28
- Regione/PA: Umbria — Codice: UMB — Numero versioni: 8 — Prima versione: 2023-12-20 — Ultima versione: 2026-07-29
- Regione/PA: Valle d'Aosta — Codice: VDA — Numero versioni: 3 — Prima versione: 2023-03-22 — Ultima versione: 2026-04-16
- Regione/PA: Veneto — Codice: VEN — Numero versioni: 4 — Prima versione: 2023-03-15 — Ultima versione: 2025-01-21

Totale: oltre 100 record.

Questa **forte disomogeneità** (da 1 a 8 versioni per regione) riflette:

- **Diversi stadi di avanzamento** del processo di approvazione
- **Diversi ritmi di aggiornamento** delle Regioni
- **Complessità amministrativa** variabile
- **Eventi contingenti** — es. Calabria ferma alla versione 1 dal 2022

3. Componenti tecnologiche

3.1 Front-end

3.1.1 HTML5

Struttura semantica con:

- `<header>` — intestazione
- `<main>` — contenuto principale
- `<div class="panel">` — contenitori
- `<select>` — menu a tendina
- `<input type="search">` — ricerca
- `<ul class="list">` — lista risultati

Attributi ARIA per l'accessibilità:

- `aria-live="polite"` — annuncio automatico dei cambiamenti
- `role` e `aria-label` — etichette esplicite

3.1.2 CSS3

Design system coerente con la banca dati PSR:

- **Palette:** blu #1e3a8a, #2563eb, grigi #4b5563, #6b7280
- **Tipografia:** Inter, system-ui
- **Layout:** flexbox, grid
- **Componenti:** badge, bottoni, card, tooltip, legenda

3.1.3 JavaScript (ES6+)

Logica applicativa moderna:

- `const, let`
- Arrow functions
- Template literals
- `async/await`
- `Object.entries()`
- `Array.prototype.sort(), filter(), map()`

3.2 Librerie esterne — Analisi approfondita

La banca dati complessa "CSR Regionali" si avvale delle **stesse quattro librerie JavaScript esterne** della banca dati PSR, garantendo coerenza tecnologica e manutenibilità. Di seguito un'analisi dettagliata di ciascuna.

3.2.1 JSZip (versione 3.10.1)

Fornitore: CDNJS (cdnjs.cloudflare.com) **Licenza:** MIT (permissiva, consente uso commerciale e modifiche) **Peso:** ~100 KB (minificato)

Funzione principale: generazione di archivi ZIP lato client, senza necessità di un server intermedio.

Dettagli tecnici:

- Consente di creare archivi compressi direttamente nel browser
- Supporta la compressione DEFLATE (il metodo standard per i file ZIP)
- Gestisce cartelle, file annidati e metadati
- API asincrona (`generateAsync()`) che non blocca il thread principale
- Compatibile con tutti i browser moderni (Chrome, Firefox, Safari, Edge)

Utilizzo nella banca dati:

- Creazione di un oggetto JSZip per ogni download
- Creazione di una cartella con nome CSR_{Regione} (spazi sostituiti con underscore)
- Aggiunta di ogni PDF scaricato come file binario (`folder.file(fileName, blob)`)
- Aggiunta del file README.txt con l'elenco completo dei documenti
- Generazione dell'archivio con `zip.generateAsync({ type: "blob" })`
- Avvio del download tramite un link temporaneo (`URL.createObjectURL`)

Perché è stata scelta:

- **Nessuna dipendenza server** — la compressione avviene interamente nel browser, riducendo il carico sul server
- **Standard de facto** — è la libreria più utilizzata per la manipolazione ZIP in JavaScript
- **Manutenzione attiva** — aggiornamenti regolari e community ampia

3.2.2 SheetJS (xlsx.full.min.js, versione 0.18.5)

Fornitore: CDNJS (cdnjs.cloudflare.com) **Licenza:** Apache 2.0 (permissiva) **Peso:** ~800 KB (minificato)

Funzione principale: lettura, manipolazione e scrittura di fogli di calcolo in formato Excel (XLSX) direttamente nel browser.

Dettagli tecnici:

- Supporta i formati XLSX, XLS, CSV, ODS e altri
- Consente la creazione di fogli di lavoro da array di oggetti JavaScript
- Gestisce stili, formule, grafici e tabelle pivot

- API sincrona e asincrona
- Compatibile con browser e Node.js

Utilizzo nella banca dati:

- Conversione dei dati piatti (flattenCSRData()) in un foglio di lavoro
- Creazione di un nuovo workbook (XLSX.utils.book_new())
- Aggiunta del foglio al workbook (XLSX.utils.book_append_sheet())
- Scrittura del file Excel (XLSX.writeFile())
- Download automatico con nome CSR_OpenData_{data}.xlsx

Perché è stata scelta:

- **Standard industriale** — è la libreria di riferimento per Excel in JavaScript
- **Formati multipli** — consente di esportare anche in CSV e JSON con la stessa libreria
- **Affidabilità** — utilizzata da migliaia di applicazioni enterprise

3.2.3 Leaflet (versione 1.x)

Fornitore: caricata dal server istituzionale (/downloads/leaflet.js) **Licenza:** BSD-2-Clause (permissiva) **Peso:** ~145 KB (minificato)

Funzione principale: rendering di mappe interattive georeferenziate nel browser.

Dettagli tecnici:

- Basata su HTML5 Canvas e SVG
- Supporta GeoJSON, KML, GPX e altri formati geografici
- Gestisce layer, marker, poligoni, tooltip e popup
- Ottimizzata per performance su dispositivi mobili
- Plugin ecosystem molto ampio

Utilizzo nella banca dati:

- Inizializzazione della mappa con centro su Italia ([42.5, 12.5], zoom 5)
- Caricamento dei GeoJSON (regioni.geojson e province.geojson)
- Creazione di layer geografici con L.geoJSON()
- Applicazione di stili personalizzati (colori, spessori, opacità)
- Gestione degli eventi di click e hover
- Binding dei tooltip con layer.bindTooltip()
- Adattamento automatico della vista con map.fitBounds()

Perché è stata scelta:

- **Leggerezza** — molto più leggera di alternative come OpenLayers o Mapbox GL
- **Semplicità** — API intuitiva e ben documentata

- **Open source** — licenza BSD, nessun costo di licenza
- **Affidabilità** — utilizzata da NASA, GitHub, e migliaia di siti istituzionali

3.2.4 Leaflet CSS (versione 1.x)

Fornitore: caricata dal server istituzionale (/downloads/leaflet.css) **Licenza:** BSD-2-Clause (permissiva)

Funzione principale: stile e controlli della mappa Leaflet.

Dettagli tecnici:

- Definisce lo stile dei controlli (zoom, attribuzione, scale)
- Gestisce il layout dei tooltip e dei popup
- Include i marker e le icone di default
- Compatibile con il sistema di design custom della banca dati

Utilizzo nella banca dati:

- Stile dei controlli di zoom (posizione, dimensione, colore)
- Stile dei tooltip (sfondo, bordo, font)
- Stile dei popup (se utilizzati)
- Integrazione con il CSS custom (`#map.leaflet-container { font-family: "Inter", system-ui, sans-serif; }`)

Perché è stata scelta:

- **Coerenza visiva** — garantisce che la mappa si integri perfettamente con il design della pagina
- **Performance** — CSS leggero e ottimizzato
- **Compatibilità** — funziona con tutte le versioni di Leaflet

3.3 Dati geografici

Identici alla banca dati PSR:

- `regioni.geojson` — confini amministrativi regionali
- `province.geojson` — confini provinciali (per le PA di Trento e Bolzano)

Fonte: openpolis/geojson-italy (CC-BY 4.0), basato su dati ISTAT.

4. Funzionalità implementate

4.1 Mappa di calore interattiva cliccabile

4.1.1 Logica di colorazione

La mappa rappresenta **cromaticamente** il numero di versioni CSR pubblicate:

- Numero versioni: 1 — Colore: Azzurro chiaro — Codice esadecimale: #dbe4ff
- Numero versioni: 2 — Colore: Azzurro medio — Codice esadecimale: #bac8ff
- Numero versioni: 3 — Colore: Azzurro intenso — Codice esadecimale: #91a7ff
- Numero versioni: 4 — Colore: Blu medio — Codice esadecimale: #5c7cfa
- Numero versioni: 5 — Colore: Blu intenso — Codice esadecimale: #3b5bdb
- Numero versioni: 6+ — Colore: Blu scuro — Codice esadecimale: #1e3a8a

4.1.2 Mappatura ISTAT

Stessa logica della banca dati PSR:

- `getIstatCode()` estrae il codice ISTAT
- `ISTAT_TO_CODES` mappa i codici alle entità
- Gestione Province Autonome separate

4.1.3 Interazioni disponibili

Hover:

- Tooltip sticky con nome entità, numero versioni, data ultima modifica

Click:

- Selezione regione
- Sincronizzazione col menu
- Evidenziazione confine

Legenda dinamica:

- Generata automaticamente

4.2 Motore di ricerca

Filtri testuali su anno, versione, titolo, data. Ricerca case-insensitive, aggiornamento in tempo reale.

4.3 Ordinamento cronologico

Versioni presentate in ordine decrescente di data.

4.4 Download ZIP con README — Analisi approfondita

La funzione `downloadAllPdfs()` è una delle **funzionalità più complesse e utili** della banca dati. Consente all'utente di scaricare **tutte le versioni CSR di una Regione in un unico archivio ZIP**, corredato da un file README che documenta l'intero contenuto.

4.4.1 Flusso operativo dettagliato

Fase 1 — Verifica dei dati:

```
const files = CSR_INDEX[regionCode];  
  
if (!files || !files.length) return alert("Nessun documento disponibile per questa Regione.");
```

- Controlla che la Regione selezionata abbia almeno un documento
- In caso negativo, mostra un alert all'utente

Fase 2 — Creazione dell'archivio ZIP:

```
const zip = new JSZip();  
  
const folder = zip.folder(`CSR_${regionName.replace(/\s+/g, "_")}`);
```

- Istanza un nuovo oggetto JSZip
- Crea una cartella interna con nome CSR_{Regione} (spazi → underscore)

Fase 3 — Inizializzazione del README:

```
let readme = `Elenco versioni CSR ${regionName}\n-----\n`;
```

- Il README inizia con un'intestazione che identifica la Regione

Fase 4 — Aggiornamento dello stato del pulsante:

```
const btn = document.querySelector(".btn-zip");  
  
btn.textContent = "Download in corso...";
```

- Il pulsante diventa "Download in corso..." per informare l'utente

Fase 5 — Download e aggiunta dei PDF:

```
for (const file of files) {  
  const fixedUrl = ensureHttps(file.url);  
  try {  
    const response = await fetch(fixedUrl);  
    const blob = await response.blob();  
    const fileName = `v${fmtVersion(file.version)}_${regionName}.pdf`;  
    folder.file(fileName, blob);  
    readme += `Versione ${fmtVersion(file.version)} - Data  
${fmtDate(file.date)}\n${fixedUrl}\n\n`;  
  } catch (err) {  
    readme += `⚠ Errore download: ${fixedUrl}\n`;  
  }  
}
```

- Itera su tutte le versioni della Regione
- Normalizza l'URL con ensureHttps()
- Scarica il PDF con fetch() e lo converte in blob
- Aggiunge il file alla cartella ZIP con nome v{versione}_{Regione}.pdf

- Aggiunge la riga corrispondente al README
- In caso di errore, registra l'errore nel README senza interrompere il processo

Fase 6 — Aggiunta del README all'archivio:

```
folder.file("README.txt", readme);
```

Fase 7 — Generazione dell'archivio ZIP:

```
const content = await zip.generateAsync({ type: "blob" });
```

- Genera l'archivio compresso in formato blob

Fase 8 — Avvio del download:

```
const link = document.createElement("a");
link.href = URL.createObjectURL(content);
link.download = `CSR_${regionName}.zip`;
link.click();
```

- Crea un link temporaneo
- Imposta l'URL del blob
- Imposta il nome del file (CSR_{Regione}.zip)
- Simula il click per avviare il download

Fase 9 — Ripristino dello stato del pulsante:

```
btn.textContent = "📦 Scarica tutte le versioni in ZIP";
```

4.4.2 Esempio di README generato

Elenco versioni CSR Abruzzo

Versione 4 - Data 21/10/2025

https://www.reterurale.it/downloads/CSR_Abruzzo_V4_21102025.pdf

Versione 3 - Data 27/02/2025

https://www.reterurale.it/downloads/CSR_Abruzzo_V3_27022025.pdf

Versione 2 - Data 15/02/2024

https://www.reterurale.it/downloads/CSR_Abruzzo_V2_15022024.pdf

Versione 1 - Data 29/12/2022

https://www.reterurale.it/downloads/CSR_Abruzzo_V1_29122022.pdf

4.4.3 Vantaggi per l'utente

- **Risparmio di tempo** — scarica tutte le versioni in un unico click
- **Organizzazione** — file rinominati in modo coerente (v4_Abruzzo.pdf)

- **Documentazione** — README con elenco completo e link
- **Affidabilità** — gestione degli errori senza interrompere il processo
- **Portabilità** — archivio ZIP compatibile con tutti i sistemi operativi

4.5 Copia link negli appunti — Analisi approfondita

La funzione `copyAllLinks()` consente di **copiare l'elenco completo dei link** delle versioni CSR di una Regione in un unico testo formattato, pronto per essere incollato in documenti, email o fogli di calcolo.

4.5.1 Flusso operativo dettagliato

Fase 1 — Verifica dei dati:

```
const files = CSR_INDEX[regionCode];

if (!files || !files.length) return alert("Nessun documento disponibile per questa Regione.");
```

Fase 2 — Generazione del testo formattato:

```
let text = `Elenco link CSR ${regionName}\n-----\n`;

files.forEach(f => {
  const fixedUrl = ensureHttps(f.url);
  text += `Versione ${fmtVersion(f.version)} (${fmtDate(f.date)}) → ${fixedUrl}\n`;
});
```

- Intestazione con nome della Regione
- Per ogni versione: numero, data, URL normalizzato

Fase 3 — Copia nella clipboard:

```
navigator.clipboard.writeText(text)

.then(() => alert("☑ Elenco link copiato negli appunti"))

.catch(() => alert("⚠ Errore nella copia dei link"));
```

- Utilizza l'API `navigator.clipboard` (standard moderno)
- Gestisce sia il successo che l'errore

4.5.2 Esempio di testo copiato

Elenco link CSR Abruzzo

Versione 4 (21/10/2025) → https://www.reterurale.it/downloads/CSR_Abruzzo_V4_21102025.pdf
 Versione 3 (27/02/2025) → https://www.reterurale.it/downloads/CSR_Abruzzo_V3_27022025.pdf
 Versione 2 (15/02/2024) → https://www.reterurale.it/downloads/CSR_Abruzzo_V2_15022024.pdf
 Versione 1 (29/12/2022) → https://www.reterurale.it/downloads/CSR_Abruzzo_V1_29122022.pdf

4.5.3 Vantaggi per l'utente

- **Velocità** — copia tutti i link in un unico click
- **Formato uniforme** — testo ordinato e leggibile

- **Compatibilità** — può essere incollato in qualsiasi applicazione
- **Tracciabilità** — ogni link è associato alla versione e alla data

4.6 Esportazione Open Data — La banca dati è completamente Open Data

La banca dati complessa "CSR Regionali" è **completamente open data**, nel pieno rispetto dei principi di trasparenza, riusabilità e interoperabilità promossi dalla normativa europea e italiana (direttiva PSI, CAD — Codice dell'Amministrazione Digitale).

4.6.1 Principi open data applicati

- **Accessibilità totale** — tutti i dati sono liberamente accessibili senza autenticazione, registrazione o pagamento
- **Formati aperti** — i dati sono disponibili in formati standard e non proprietari (JSON, CSV, XLSX)
- **Riusabilità** — i dati possono essere utilizzati, modificati e ridistribuiti da chiunque, anche per scopi commerciali
- **Granularità** — i dati sono disponibili a livello di singolo record (versione CSR per Regione)
- **Aggiornamento** — i dati vengono aggiornati regolarmente man mano che le Regioni pubblicano nuove versioni

4.6.2 Pannello "Scarica database CSR (Open Data)"

Il pannello dedicato consente di esportare l'intero archivio in **tre formati aperti**:

JSON (JavaScript Object Notation):

- Formato standard per lo scambio di dati strutturati
- Leggibile da qualsiasi linguaggio di programmazione
- Struttura gerarchica che preserva le relazioni tra i dati
- File di esempio: CSR_OpenData_2026-09-01.json

CSV (Comma Separated Values):

- Formato tabellare universale
- Compatibile con Excel, LibreOffice, Google Sheets
- Importabile in qualsiasi database relazionale
- File di esempio: CSR_OpenData_2026-09-01.csv

XLSX (Excel Open XML):

- Formato standard per fogli di calcolo
- Include intestazioni di colonna e formattazione
- Compatibile con Microsoft Excel e alternative open source
- File di esempio: CSR_OpenData_2026-09-01.xlsx

4.6.3 Struttura dei dati esportati

Ogni file esportato include i seguenti campi per ogni record:

- Campo: `codice` — Descrizione: Codice regionale normalizzato (es. ABR)
- Campo: `regione` — Descrizione: Nome completo della Regione/PA
- Campo: `versione` — Descrizione: Numero di versione (es. 4)
- Campo: `data` — Descrizione: Data di pubblicazione (formato italiano DD/MM/YYYY)
- Campo: `titolo` — Descrizione: Denominazione del Complemento (es. CSR Abruzzo)
- Campo: `periodo` — Descrizione: Periodo di programmazione (2023-2027)
- Campo: `url` — Descrizione: Collegamento diretto al PDF

4.6.4 Vantaggi dell'approccio open data

- **Trasparenza amministrativa** — i cittadini possono verificare l'uso dei fondi europei
- **Ricerca scientifica** — i ricercatori possono analizzare i dati con strumenti statistici
- **Innovazione** — gli sviluppatori possono creare applicazioni basate sui dati
- **Interoperabilità** — i dati possono essere integrati con altre banche dati (es. PSR)
- **Riuso** — i dati possono essere combinati con altre fonti per analisi avanzate

5. Gestione degli errori e robustezza

5.1 Normalizzazione URL

Stessa funzione `ensureHttps()` della banca dati PSR.

5.2 Gestione fetch fallita

Errori registrati nel README senza interrompere il processo.

5.3 Stati di caricamento

- "Caricamento confini geografici..." per la mappa
- "Download in corso..." per lo ZIP

5.4 Messaggi di errore

- "Errore: Leaflet non caricato..."
 - "Errore nel caricamento dei confini..."
 - "Nessun documento disponibile..."
 - "Nessun documento trovato..."
-

6. Comunicazione con la pagina madre

Sistema di resize dinamico via `postMessage` con tipo `'dbCSR:resize'`:

- Invio iniziale al load
 - Ritardi di 200ms e 800ms
 - `MutationObserver` e `ResizeObserver`
 - `setInterval` di sicurezza ogni 1.5 secondi
-

7. Complessità e valore dell'opera

7.1 Fasi di realizzazione

1. **Monitoraggio continuo**
 - Il periodo 2023-2027 è in piena evoluzione
 - Nuove versioni vengono pubblicate con frequenza variabile
 - Aggiornamenti costanti della banca dati
2. **Gestione della transizione normativa**
 - Passaggio da PSR a CSR
 - Numerazioni non uniformi (versioni 2.0, 2.1, 5, 7, 8)
 - Denominazioni regionali diverse
3. **Verifica documentale puntuale**
 - Ogni versione verificata incrociando date, numeri e URL
 - Controllo sui siti istituzionali
4. **Integrazione geografica**
 - Mappatura codici ISTAT
 - Gestione Province Autonome come entità separate
5. **Sviluppo front-end complesso**
 - Interazione sincrona tra mappa, menu e lista risultati
6. **Ottimizzazione per l'integrazione**
 - Comunicazione cross-frame per il dimensionamento nel portale

7.2 Valore per gli utenti

- **Ricercatori** — analisi dell'attuazione della nuova PAC
- **Amministratori pubblici** — monitoraggio dello stato di avanzamento
- **Tecnici del settore** — accesso alle versioni aggiornate
- **Cittadini** — trasparenza sull'uso dei fondi europei

8. Conclusioni

La banca dati complessa "CSR Regionali" rappresenta uno **strumento indispensabile** per seguire l'attuazione della nuova PAC 2023-2027 in Italia. La sua **natura dinamica** — con aggiornamenti frequenti man mano che le Regioni pubblicano nuove versioni — la rende un **osservatorio privilegiato** sullo stato di avanzamento della programmazione agricola europea nel nostro Paese.

La complessità di questa banca dati non risiede solo nella quantità di dati, ma nella **capacità di tenere il passo con un processo amministrativo in corso**, offrendo a ricercatori, amministratori e cittadini una **visione sempre aggiornata e storicamente tracciabile** dell'evoluzione delle politiche di sviluppo rurale. La natura **completamente open data** della banca dati ne amplifica il valore, rendendola una risorsa pubblica riutilizzabile da chiunque.

