



Ricerca di Sistema elettrico

Test di funzionalità e aggiornamento della piattaforma A.I.D.A.

Pasquale Regina e Nicola Colonna

TEST DI FUNZIONALITÀ E AGGIORNAMENTO PIATTAFORMA A.I.D.A.

Pasquale Regina, Nicola Colonna (ENEA)

Settembre 2013

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dello Sviluppo Economico - ENEA

Piano Annuale di Realizzazione 2012

Area: Produzione di energia elettrica e protezione dell'ambiente

Progetto: Sviluppo di sistemi per la produzione di energia elettrica da biomasse e l'upgrading dei biocombustibili

Obiettivo: Sviluppo dei sistemi di produzione di biocombustibili

Responsabile del Progetto: Vito Pignatelli, ENEA

Responsabile scientifico ENEA: Nicola Colonna

Si ringraziano per il contributo ed il supporto:

Maria Rita Mastrullo e Simona Scarfoglio dell'Università Federico II di Napoli.

Viviana Cigolotti e Vincenzo Gerardi dell'ENEA

Indice

SOMMARIO.....	4
1 INTRODUZIONE.....	5
2 DESCRIZIONE DELLE ATTIVITÀ SVOLTE E RISULTATI.....	6
2.1 TEST DI FUNZIONALITÀ E AGGIORNAMENTI DELLA PIATTAFORMA WEB AIDA	6
2.2 VERIFICA DI USABILITÀ.....	7
2.3 VERIFICA DELLA FUNZIONALITÀ DELL'APPLICAZIONE WEB	8
2.3.1 <i>Il ciclo For...Next</i>	9
2.3.2 <i>Sintassi per la creazione di un ciclo</i>	10
2.3.3 <i>Il contatore</i>	11
2.3.4 <i>L'istruzione Step</i>	11
2.3.5 <i>L'istruzione Next</i>	11
2.3.6 <i>Uscita dal ciclo</i>	12
2.3.7 <i>Le istruzioni da scrivere in un corpo For ... Next</i>	12
2.3.8 <i>Utilizzo e manipolazione del contatore</i>	12
2.3.9 <i>Che cosa è sconsigliato in un blocco For ... Next</i>	13
2.3.10 <i>Chiusura forzata di un ciclo</i>	14
2.3.11 <i>Il controllo del flusso di un ciclo</i>	15
2.3.12 <i>Controlli ASP.NET AJAX</i>	17
2.3.13 <i>Linguaggio PHP</i>	18
2.4 VERIFICA DEL CALCOLO E DEI RISULTATI A VIDEO.....	20
2.5 AGGIORNAMENTO E STIMA DATI BIOMASSE	21
2.5.1 <i>Le biomasse residuali agricole</i>	21
3 CONCLUSIONI.....	24
RIFERIMENTI BIBLIOGRAFICI.....	25

Sommario

La piattaforma web A.I.D.A. implementa uno strumento di calcolo per valutare la prefattibilità tecnico-economica di impianti di digestione anaerobica e di gassificazione. Offre un supporto nel valutare, per diverse tipologie di utenti, in funzione delle caratteristiche delle biomasse, il pre-dimensionamento di un impianto per la produzione di biogas o syngas e le sue prestazioni energetiche ed economiche in relazione agli incentivi attualmente in essere.

La piattaforma è principalmente composta dai seguenti elementi:

- Interfaccia utente, utilizzata per selezionare e/o inserire i dati di input utili alla simulazione (le biomasse, tipo e quantità, nonché le tipologie di processo e di motori per la generazione di energia).
- Database dei substrati e delle tecnologie.
- Funzioni di calcolo per il dimensionamento del processo e degli impianti.
- Principali risultati di prestazioni energetiche utili ad una valutazione di prefattibilità.

1 Introduzione

Nel corso delle tre precedenti annualità di lavoro, l'ENEA, in collaborazione con il DETEC Dipartimento di Energetica, Termofluidodinamica Applicata e Condizionamenti Ambientali dell'Università degli Studi di Napoli Federico II, ha sviluppato una piattaforma web denominata A.I.D.A. (Advanced and Innovative tool for Developing feasibility Analysis of biomass plants) che implementa uno strumento di calcolo in grado di analizzare alcune tra le filiere bio-energetiche più diffuse nel sistema produttivo nazionale, contestualizzandole dal punto di vista territoriale, allo scopo di poterne eseguire un'approfondita valutazione sotto i tre aspetti economico, energetico e ambientale che concorrono a determinarne la sostenibilità globale della valorizzazione energetica delle biomasse.

Le prime due annualità del progetto sono state rivolte alla definizione e implementazione della piattaforma web, mentre la terza annualità dell'Accordo di Programma è stata dedicata al completamento e aggiornamento di alcune funzioni, al trasferimento dell'applicativo presso i server del centro ENEA di Casaccia, alla definizione di un manuale per l'utente e alla verifica e collaudo funzionale della piattaforma al fine di rendere operativa la versione finale dell'applicativo per il pubblico. Nell'ultimo anno la piattaforma è stata oggetto di ulteriori analisi al fine di migliorare alcune funzioni e verificarne la reale operatività in condizioni d'uso. Il presente rapporto descrive brevemente le innovazioni, modifiche e integrazioni realizzate.

2 Descrizione delle attività svolte e risultati

2.1 Test di funzionalità e aggiornamenti della piattaforma web AIDA

L'attività relativa alla presente annualità ha preso corpo a seguito del trasferimento dell'applicativo web dall' ambiente di sviluppo verso un server di produzione appositamente allestito, configurato e integrato nell'infrastruttura di rete ENEA, ubicata presso il centro ENEA della Casaccia.

Attualmente l'applicativo web sviluppato è raggiungibile all'indirizzo:

<http://aida.casaccia.enea.it/aida/default.asp>.

Le attività condotte possono essere classificate sinteticamente in:

- Ulteriori verifiche e test di funzionalità della piattaforma;
- Verifica e controllo dei risultati di calcolo tramite simulazioni diversificate;
- Raccolta di indicazioni e suggerimenti da utenti selezionati per modifiche e/o integrazioni;
- Risoluzione di errori di visualizzazione e calcolo;
- Inserimento di nuove funzionalità ed informazioni per l'utenza.

A partire dal primo settembre 2012, data del trasferimento della piattaforma WEB A.I.D.A. presso il server ENEA della Casaccia, è iniziata una fase di test e di collaudo tecnico a cura dell'ENEA e del Dipartimento di Ingegneria Industriale dell'Università degli Studi di Napoli Federico II al fine di monitorare la funzionalità della piattaforma nel suo complesso.

In generale una applicazione web ha la necessità di essere sottoposta a continui e ripetuti test per verificarne la funzionalità, la sequenzialità delle operazioni, la corretta restituzione dei calcoli e delle elaborazioni svolte e implementate dal/nel codice.

Ne consegue la opportunità di sottoporre l'applicativo web ad una ampia utenza anche diversificata, non solo tecnici esperti, in grado di individuare a colpo d'occhio errori macroscopici di congruenza dei risultati, ma anche utenti che possano fornire indicazioni utili circa la fruibilità, semplicità d'uso e/o miglioramenti nell'uso quotidiano dello strumento stesso.

In sostanza una utenza in grado di fornire commenti e suggerimenti sia per correzioni e modifiche ma anche input per ulteriori e successive implementazioni.

La necessità di test su esposta è ancor più sentita per applicazioni simili a quella realizzata in quanto i tre percorsi, simili, di navigazione possono indurre l'utente ad errori grossolani oppure a perdere di vista il contesto/percorso di navigazione intrapreso.

Nonostante i test inizialmente condotti (prima sul server di sviluppo e successivamente sul server di produzione, prima della pubblicazione e messa online) abbiano accertato la funzionalità con i principali sistemi operativi e con i browser più diffusi, il processo di controllo e verifica delle funzionalità ed anche dei risultati di calcolo non si è mai arrestato. Gli autori concordano nel ritenere che solo un uso operativo continuo, quasi quotidiano, da parte di una utenza più ampia dia la possibilità di identificare eventuali problemi sfuggiti ai test formalizzati già effettuati.

Anche in questa ottica, nel dicembre 2012 è stato organizzato un seminario dal titolo "A.I.D.A. uno strumento web per l'analisi di fattibilità di impianti a biomasse" presso il centro ENEA della Casaccia per diffondere ai tecnici e ricercatori ENEA, esperti del settore, i risultati ottenuti e le prestazioni del sistema implementato.

Al termine sono stati invitati i partecipanti a far pervenire osservazioni e consigli per migliorare il servizio offerto.

Da riscontri pervenuti dai ricercatori coinvolti e da simulazioni numeriche per testare la piattaforma online è stato individuato un elenco di aggiornamenti da effettuare, distinti per 3 tipologie di categorie, cui è stato associato un diverso grado di priorità:

- Verifica del calcolo e dei risultati numerici a video;
- Verifica delle funzionalità del software;

- Verifica di usabilità.

Si intuisce facilmente il grado di priorità crescente negli aggiornamenti richiesti. Tuttavia si è ritenuto utile ed efficace seguire l'ordine inverso.

Tale scelta è stata dettata da opportunità legate alla tempistica di intervento, semplicità e rapidità di implementazione ed esecuzione ma anche dalla necessità di approfonditi controlli per verificare la ripetibilità degli inconvenienti riscontrati.

2.2 Verifica di usabilità

Per migliorare la usabilità della piattaforma sono stati realizzati aggiornamenti che hanno comportato:

- Inserimento di nuove voci di glossario;
- Inserimento sezione "Contatti" e sezione "Aggiornamenti" (Figura 1);
- Inserimento sezione "Accessibilità" con informazioni su compatibilità, ottimizzazione, software con il quale è stata realizzata la piattaforma (Figura 2);
- Aggiunta sezione "About AIDA" con breve descrizione in inglese di A.I.D.A..

Di particolare importanza la creazione di ulteriori sezioni quale "aggiornamenti" all'interno della quale è riportata, in formato tabellare, la sequenza cronologica delle modifiche, delle integrazioni e delle revisioni della piattaforma (Figura 3).



Figura 1. Nuovo header della pagina web con aggiunta nuove sezioni.

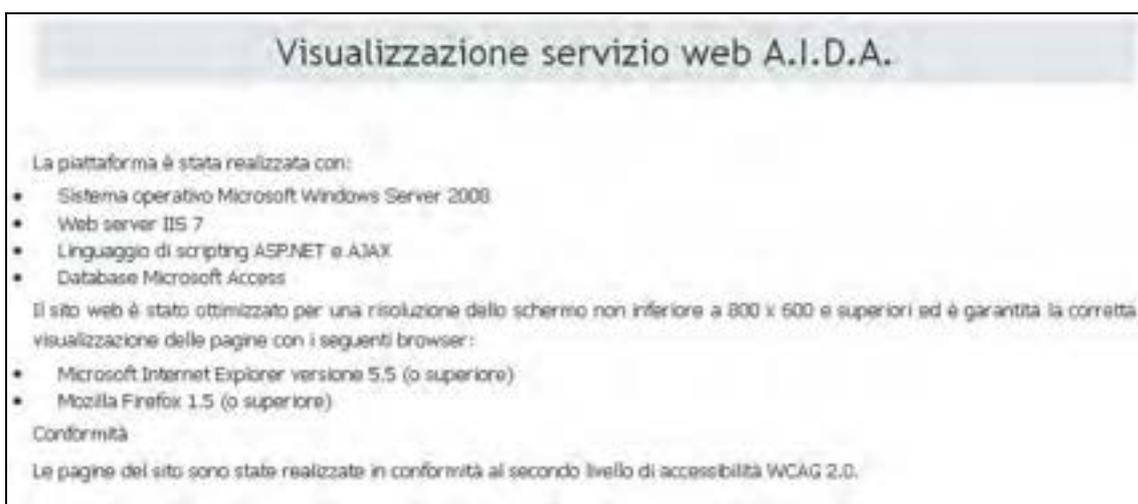


Figura 2. Descrizione della pagina "Accessibilità"

Altra sezione aggiunta è quella relativa ai “contatti”. Attualmente è presente un unico contatto denominato aidainfo@enea.it a cui l’utente può richiedere informazioni sul sistema e sulla piattaforma ma anche per segnalare malfunzionamenti o problemi. Nella fase iniziale si è pensato ad un unico contatto per evitare dispersione di informazioni e/o richieste da parte dell’utenza. Successivamente si provvederà ad una caratterizzazione delle competenze coinvolte e conseguente diversa lista di distribuzione delle segnalazioni (contatto per il webmaster, contatto per il web designer,...)

Aggiornamenti servizio web A.I.D.A.	
Sono riportate in sequenza cronologica le modifiche, le integrazioni e le revisioni della piattaforma.	
Cronologia modifiche ed aggiornamenti	
03 giugno 2013	Inserita la sezione About AIDA.
20 maggio 2013	Riviste alcune fasi di calcolo.
08 dicembre 2012	Inserita la sezione Aggiornamenti.
03 dicembre 2012	Disponibile online la sezione Info.
28 settembre 2012	Completamento test con utenti selezionati per verifica funzionalità e gradimento.
23 settembre 2012	Inizio fase di alfatesting.
11 settembre 2012	Completamento collaudo compatibilità browser. Verifica funzionalità server.
01 settembre 2012	Attivazione della piattaforma e del sito. Rilascio versione test di AIDA.

Figura 3. Schermata, aggiornata al 3 giugno 2013, della sezione “Aggiornamenti”

In virtù della presentazione della Piattaforma in ambito internazionale [1], effettuata nell’ambito della Conferenza internazionale sulle biomasse, tenutasi a Copenaghen ad inizio giugno 2013, si è ritenuto opportuno aggiungere una breve descrizione di A.I.D.A. in lingua inglese a beneficio di una utenza più ampia. E’ stata aggiunta una sezione “About A.I.D.A.” che introduce brevemente scopi e funzioni di A.I.D.A. e rimanda alla possibilità di chiedere maggiori informazioni ai curatori.

2.3 Verifica della funzionalità dell’applicazione web

Una delle peculiarità dell’applicazione web AIDA è la possibilità di eseguire uno studio di prefattibilità tecnico-economico-ambientale utilizzando matrici che possono essere facilmente “personalizzate”. È possibile infatti variare la tipologia delle biomasse considerate nonché la quantità per singola tipologia. Si ottiene una miscela o substrato le cui caratteristiche cambiano a seconda della composizione.

Tramite l’indirizzo e-mail presente nella sezione contatti sono giunte segnalazioni su un problema legato alla funzionalità nella composizione della matrice relativamente alla FORSU.

Per l’ambito territoriale di interesse, selezionando inizialmente la FORSU¹ come componente singolo della matrice² da inviare/utilizzare per il trattamento, la matrice stessa risultava correttamente caratterizzata. Nella fase successiva di stima della producibilità di biogas da FORSU erano invece visualizzati risultati non

¹ FORSU Frazione Organica dei Rifiuti Solidi Urbani

² Substrato selezionato dall’utente nella fase iniziale di analisi per determinare l’alimentazione dell’impianto

congruenti. Utilizzando una matrice differente in cui la FORSU risultava essere un componente della matrice stessa, la stima di producibilità di biogas è risultata corretta.

La rappresentazione, eventuale modifica (incremento/decremento dei componenti della miscela) e la successiva elaborazione dei dati costituenti la matrice avvengono per fasi successive di interscambio di informazioni tra l'applicazione web (pagina web) e il server. Questo dialogo avviene utilizzando sia codice Javascript che porzioni di codice AJAX. Si tratta infatti di una tabella/matrice dinamica con cicli annidati.

E' noto in ambito informatico che la creazione di pagine web dal contenuto dinamico realizzate con i linguaggi su menzionati, in particolari condizioni può generare errori simili dovuti proprio alla complessità del codice e al livello di "annidamento" delle istruzioni all'interno del codice stesso.

Nell'elaborazione della matrice sono stati utilizzati cicli ascendenti e discendenti, cicli di lettura e popolamento matriciale.

La trattazione di questo argomento meriterebbe un'analisi più accurata ma ciò esula dallo spirito e senso di questo rapporto. Di seguito alcuni dettagli utili per una migliore individuazione del problema riscontrato al fine di intuirne anche la soluzione adottata ed il lavoro di analisi effettuato.

2.3.1 Il ciclo For...Next

Se una azione o una qualsiasi elaborazione deve essere ripetuta più di una volta è cosa saggia affidarsi ai cicli che il linguaggio di sviluppo della pagina web mette a disposizione.

Il ciclo For ... Next serve per eseguire un blocco di istruzioni in modo ripetitivo.

La sintassi è:

```
For Contatore Inizio To Fine [Step incremento]
```

```
seguono le istruzioni da ripetere
```

```
Next [Contatore]
```

Dove "Inizio ", "Fine " e "incremento" possono essere rappresentati sia da valori numerici che da variabili numeriche.

Se incremento è 1 Step può essere omissso ed il Contatore viene incrementato di una unità ad ogni ciclo.

Ma incremento può essere anche un numero diverso da 1, ossia un qualsiasi numero, intero o decimale, positivo o negativo. In questo caso il Contatore viene incrementato o decrementato del valore indicato dallo Step ad ogni ciclo.

"[Contatore]" associato alla istruzione "Next" può essere tranquillamente omissso.

Tutte le istruzioni comprese in un blocco For ... Next vengono ripetute tante volte fino a quando il Contatore raggiunge il valore indicato dal limite Fine.

Quando viene incontrata l'istruzione Next al contatore viene sommato o sottratto l'incremento indicato dallo Step e viene controllato se il contatore ha superato il valore finale.

Se il valore finale non è stato superato l'esecuzione delle istruzioni riprende dalla prima istruzione che si trova subito dopo il For.

Se il valore finale è stato superato l'esecuzione delle istruzioni continua con la prima istruzione che si trova dopo l'istruzione Next.

Nel blocco del ciclo il contatore non deve essere modificato, pena un non corretto funzionamento del ciclo.

In questo ciclo:

```
For Contatore = 1 To 10
```

```
MsgBox "il contatore vale " & Contatore
```

```
Next
```

il messaggio viene mostrato 10 volte e ogni volta è mostrato il valore che la variabile Contatore ha assunto ad ogni ciclo. Assegnando nel ciclo alla variabile usata dal contatore un altro valore potremmo ottenere come risultato che il ciclo entri in un *loop* infinito come in questo caso:

```
For Contatore = 1 To 10
```

```
MsgBox "il contatore vale " & Contatore
```

```
Contatore = 3
Next
```

Il contatore rimarrà congelato a 4 per via della istruzione: Contatore = 3.

Diversamente se si usasse la variabile del contatore in modo oculato come nell'esempio seguente:

```
For Contatore = 1 To 10
  MsgBox "il contatore vale " & Contatore
  If Contatore = 3 Then Contatore = 6
Next
```

Questo modo fa sì che quando il contatore assume il valore di 3 passi subito a 6 saltando i valori 4 e 5.

All'uscita dal ciclo il contatore ha il valore superiore al valore finale dipendente dallo Step.

Più cicli possono essere nidificati ma non possono accavallarsi tra loro. L'ultimo ciclo aperto deve essere chiuso per primo.

```
For Contatore = 1 To 10
  MsgBox "il contatore esterno vale " & Contatore
  For Contatore2 = 1 To 5
    MsgBox "il contatore interno vale " & Contatore2
    If Contatore = 3 Then Contatore = 6
  Next Contatore2
Next Contatore
```

All'interno di un ciclo possono essere aperti e chiusi più cicli.

```
For Contatore = 1 To 10
  For Contatore2 = 1 To 5
    MsgBox "il contatore vale " & Contatore
  Next Contatore2
  MsgBox "qui si possono fare altre cose"
  For Contatore3 = 1 To 5
    MsgBox "il contatore vale " & Contatore
  Next Contatore3
Next Contatore
```

2.3.2 Sintassi per la creazione di un ciclo

Questa è la sintassi e la forma da usare:

```
For Contatore Inizio To Fine [Step incremento]
... blocco istruzioni da eseguirsi all'interno del ciclo
Next [Contatore]
```

primo caso: costruzione di un ciclo in senso ascendente

riempimento di una matrice con numeri casuali:

```
Sub riempimentoMatrice()
Dim MiaMatrice(1 To 10)
Dim I, A
Randomize (Timer)
For I = 1 To 10 Step 1
A = Int(Rnd() * 100) + 1
MiaMatrice(I) = A
Next I
End Sub
```

lettura di una matrice in senso ascendente:

```
Sub letturaMatrice()  
Dim MiaMatrice(1 To 10)  
Dim I, A  
Randomize Timer  
For I = 1 To 10 Step 1  
A = MiaMatrice(I)  
Next I  
End Sub  
lettura di un ciclo in senso discendente:  
Sub letturaMatrice2()  
Dim MiaMatrice(1 To 10)  
Dim I, A  
Randomize Timer  
For I = 10 To 1 Step -1  
A = MiaMatrice(I)  
Next I  
End Sub
```

Ovviamente nulla vieta di leggere le due matrici viste sopra nel senso inverso da come sono state create. Basta sapere che in questi casi la lettura avviene in senso inverso: ciò che è stato inserito prima verrà letto alla fine e ciò che è stato inserito alla fine verrà letto prima.

2.3.3 Il contatore

Il contatore del ciclo è una variabile di tipo numerico e varia il suo valore ogni volta si incontra l'istruzione "Next".

All'uscita dal ciclo il contatore avrà un valore superiore a quello indicato dall'indice maggiore dipendente dallo Step usato. Ma questo sarà analizzato nel paragrafo "Uscita dal ciclo".

E' necessario fare molta attenzione. Se nel corpo del ciclo dobbiamo usare delle variabili dobbiamo evitare assolutamente che tale variabile abbia lo stesso nome del contatore usato nel ciclo in quanto questo fatto cambierebbe l'andamento del ciclo. In certi casi porterebbe anche ad un loop infinito.

Tuttavia a volte possiamo avere la necessità di farlo. In ogni caso non è mai consentito assegnare al contatore, nel corpo di un ciclo, un valore stringa. Il contatore del ciclo deve sempre essere di tipo numerico anche se è lecito dichiararlo, nella zona delle dichiarazioni, come variabile.

2.3.4 L'istruzione Step

E' facoltativa nel caso di un ciclo che viaggia in senso ascendente e, se omissa, l'incremento è unitario; se l'incremento deve essere diverso da 1 sarà necessario indicare lo Step.

E' sempre obbligatoria nel caso di un ciclo che viaggia in senso discendente per indicare il decremento da effettuare durante il ciclo.

2.3.5 L'istruzione Next

In un normale ciclo l'istruzione Next indica la fine del blocco ed incrementa il contatore. Quando il contatore ha raggiunto il limite massimo indicato nella istruzione For consente l'uscita dal ciclo ed il Contatore ha un valore immediatamente superiore al limite massimo indicato in For

I cicli possono essere anche nidificati, ossia in un ciclo For ... Next è possibile nidificare tanti altri cicli.

In cicli nidificati le varie istruzioni Next vanno collocate in senso inverso di come sono state collocate le istruzioni For:

- la prima istruzione Next chiude l'ultimo For

- l'ultima istruzione Next chiude il primo For

2.3.6 Uscita dal ciclo

Da un ciclo si può uscire in due modi:

- In modo naturale: quando il contatore ha raggiunto il valore finale dichiarato all'inizio del ciclo viene terminato. Uscendo dal ciclo tuttavia il contatore non avrà il valore finale imposto nella dichiarazione del ciclo, ma il valore immediatamente successivo a quello e dipende dallo Step.
- In modo forzato: a volte può capitare che non è necessario completare il ciclo. Normalmente questo succede quando si verifica una determinata condizione. In questo caso per uscire si usa l'istruzione: Exit For

Dopo l'uscita forzata dal ciclo il valore che si trova nel contatore sarà quello che aveva al momento dell'uscita:

```
For A = 1 To 100
B = Int(Rnd() * 100) + 1
If B < A Then Exit For
Next
'all'uscita forzata dal ciclo il valore corrente del contatore
'sarà quello che aveva al momento della imposizione
'di interrompere il ciclo
```

2.3.7 Le istruzioni da scrivere in un corpo For ... Next

In un ciclo For ... Next è possibile scrivere qualsiasi istruzione che deve essere ripetuta per il numero di volte indicato nel ciclo. E' possibile leggere o scrivere i contenuti delle celle, i contenuti di matrici, elaborare dati e quant'altro.

Il contatore del ciclo può essere usato anche come puntatore agli indici di vari oggetti, agli indici delle matrici, alle celle per poter leggere o scrivere i relativi valori, agli indici dei controlli (ListBox, ComboBox, i nomi dei controlli di una UserForm).

2.3.8 Utilizzo e manipolazione del contatore

Occorre la massima prudenza nel voler manipolare il Contatore di un ciclo.

Il contatore può essere usato anche per compiere più operazioni contemporaneamente.

- Puntare a due oggetti diversi

Per esempio, posso volere che i contenuti delle varie locazioni di una matrice vengano scritti nelle celle di un foglio di lavoro nelle varie righe della prima colonna.

```
Sub doppioUso()
For I = 1 To 10
' si usa il contatore del ciclo per puntare alla locazione
' della matrice e lo stesso contatore aumentato di 5 per
' puntare contemporaneamente alle righe del foglio di
' lavoro a partire dalla 6^
Cells(I + 5, 1) = MiaMatrice(I)
Next I
End Sub
```

Con queste istruzioni il contatore del ciclo compie due operazioni:

- punta alla locazione della matrice da cui legge il valore

- punta alla riga del foglio di lavoro, aiutato dal calcolo "I + 5" dove deve scrivere il valore letto

Così:

- col contatore che ha il valore 1 all'inizio del ciclo viene puntata la prima locazione della matrice e la sesta riga del foglio di lavoro,
- col contatore che ha il valore 2 viene puntata la seconda locazione della matrice e la settima riga del foglio di lavoro,
- col contatore che ha il valore 3 viene puntata la terza locazione della matrice e la ottava riga del foglio di lavoro,
- ecc.
-

- Usare il contatore di un ciclo per eseguire dei calcoli

Col contatore possiamo anche eseguire calcoli complessi che ci possono servire per compiere determinate azioni.

Col seguente esempio vogliamo che il contenuto della matrice venga scritto ogni 5 colonne a partire dalla colonna 1.

```
Sub cicli9()
```

```
Dim MiaMatrice(1 To 10)
```

```
For I = 1 To 10
```

```
MiaMatrice(I) = Cells(I, 1)
```

```
Next
```

```
For A = 1 To 10
```

```
Cells(14, A * 5 - 5 + 1) = MiaMatrice(A)
```

```
Next
```

```
End Sub
```

Cosa è vietato in un blocco For ... Next

Abbiamo sopra descritto che un ciclo "For ... Next" ha un contatore.

Ebbene nel corpo di questo ciclo è assolutamente vietato usare variabili con lo stesso nome del contatore per assegnargli nuovi valori in quanto in questo caso il valore di tale variabile verrebbe ad influenzare in modo inatteso, se non rovinoso, l'andamento del ciclo, né tantomeno assegnare a tale variabile dei valori stringa.

Nella migliore delle ipotesi il ciclo verrà sconvolto perché quando il flusso di istruzioni arriverà all'istruzione "Next" si avrà un incremento del valore contenuto erroneamente in "A".

```
Sub cicli7()
```

```
For A = 1 To 100
```

```
A = MiaMatrice(A)
```

```
Cells(A, 1) = A
```

```
Next
```

```
End Sub
```

2.3.9 Che cosa è sconsigliato in un blocco For ... Next

Normalmente ci si affida ad un ciclo quando è necessario eseguire delle azioni ripetitive.

A volte i cicli vengono eseguiti per un numero elevato di volte. Ne consegue che tante volte occorrono secondi preziosi prima che un'elaborazione eseguita in un ciclo restituisca il controllo dell'applicativo.

Quindi occorre essere oculati nello scrivere certe istruzioni all'interno di un ciclo.

Vediamo questo banale esempio:

```
For A = 1 To 10000
```

```
B = InputBox("scrivi il nome da cercare")
```

```
If B = MiaMatrice(A) Then
```

```
Cells(A, 1) = "Ciao"
```

```
End If
Next
```

Scopo di questa routine è quello di cercare nella matrice un nome fornito dall'utente tramite una InputBox. Questa istruzione, messa in questa posizione, oltre ad infastidire l'utente che, in questo caso deve rispondere alla richiesta per ben 10.000 volte, rallenta notevolmente l'esecuzione della routine. Più corretto sarebbe essere accorti nel porre fuori del ciclo certe istruzioni che non dovrebbero essere eseguite nel ciclo, come in questo caso:

```
B = InputBox("scrivi un nome")
For A = 1 To 10000
If B = MiaMatrice(A) Then
Cells(A, 1) = "Ciao"
End If
Next
```

2.3.10 Chiusura forzata di un ciclo

Abbiamo visto che quando usiamo un ciclo tutte le istruzioni contenute tra For e Next vengono ripetute ciclicamente fino a quando il contatore non assume il valore massimo impostato nell'istruzione For.

Può succedere che in alcune circostanze, se si verifica una determina condizione, è inutile uscire anticipatamente dal ciclo. In questo caso è possibile usare l'istruzione Exit For la quale sposta il punto di esecuzione del programma alle istruzioni successive all'istruzione Next relativa al ciclo interrotto.

All'uscita forzata dal ciclo il contatore del ciclo conserva il valore che ha al momento dell'uscita. Solo se si esce dal ciclo il contatore assume il valore successivo al limite massimo imposto dallo Step.

Nell'esempio che segue si vuole cercare un nome memorizzato in una matrice:

```
Sub Trovalo()
B = InputBox("scrivi un nome")
For A = 1 To 10000
If B = MiaMatrice(A) Then
F = True
Exit For
End If
Next
If F Then
MsgBox "Il nome è stato trovato"
Else
MsgBox "Il nome non è stato trovato"
End If
End Sub
```

In questo esempio si vuole cercare nella matrice il nome inserito dall'utente.

Se il nome viene trovato si forza l'uscita anticipata dal ciclo e si passa ad eseguire le istruzioni successive all'istruzione Next, altrimenti si continua con l'esecuzione del ciclo fino alla fine.

Lavorando con più cicli nidificati e, verificatasi una determinata circostanza, si desidera uscire da tutti i cicli la situazione è lievemente più complessa, ma facilmente intuibile.

Un altro esempio esemplificativo.

```
For A = 1 To 50
Nome = Nomi(A)
For B = 1 To 10000
If Nome = Elenco(B) Then
FL = True
```

```
Exit For
End If
Next
If FL Then Exit For
Next
If FL Then
MsgBox "Il nome è stato trovato"
Else
MsgBox "Il nome non è stato trovato"
End If
```

In questo frammento di codice abbiamo due matrici:

Nomi() che contiene un elenco di nomi

Elenco() che ne contiene un altro

si eseguono dei confronti tra i nomi della prima matrice e quelli della seconda matrice.

Se il confronto risulta positivo si desidera uscire dai due cicli.

La cosa è fattibile con una variabile booleana che diventa True solo se il confronto è positivo: superato il confronto si pone a True la variabile booleana e si esce dal primo ciclo.

All'uscita dal primo ciclo, quello interno, ci si viene a trovare nel corpo del secondo ciclo, quello esterno.

Per uscire anche da questo secondo ciclo è necessario che la variabile booleana sia impostata a True. In questo caso, usando un secondo Exit For verremmo a trovarci al di fuori dei due cicli nidificati procedendo con l'esecuzione del resto delle istruzioni.

2.3.11 Il controllo del flusso di un ciclo

Finora abbiamo visto come realizzare delle strutture iterative che eseguano un certo numero di istruzioni fino a che la loro condizione di fine non viene soddisfatta. Tuttavia esistono altri due modi, in JavaScript, per uscire da un ciclo prima che questo giunga naturalmente a conclusione o per "saltare" momentaneamente, tutte o in parte, le istruzioni in esso contenute ricominciando poi un altro ciclo di istruzioni. Nel primo caso si fa ricorso all'istruzione *break*, nel secondo si utilizza *continue*. Queste due istruzioni espandono notevolmente le possibilità di utilizzare le strutture iterative, consentendo di modificare durante l'esecuzione il "comportamento" di un ciclo a seconda delle evenienze che si presentano. Andiamo ora a trattare nel dettaglio le due istruzioni.

L'istruzione break

Come è facile intuire l'istruzione *break* consente di interrompere l'esecuzione di un ciclo senza che la sua condizione di fine sia stata soddisfatta. L'istruzione *break* si rivela utile se si ha l'esigenza di uscire forzatamente da un ciclo quando una condizione "speciale" richiede la fine anticipata delle iterazioni. Il listato che segue è un esempio di un caso del genere:

```
<html>
<title>Utilizzo di break</title>
<head>
<script language="JavaScript">
<!--
    for (var i=1, somma=0, num=""; i<=10; i++) {
        num=prompt('Introdurre un valore: ');
        if (num.toLowerCase()=='fine') break;
        somma+=parseInt(num);
    }
    alert('La somma totale è: ' + somma);
-->
```

```
</script>
</head>
```

Nell'esempio un ciclo for è utilizzato per chiedere all'utente 10 valori di cui alla fine sarà visualizzata la somma. Normalmente il ciclo andrebbe avanti fino a che la variabile contatore *i* non fosse uguale a 10. Tuttavia in questo caso è stata prevista la possibilità che l'utente voglia terminare il ciclo anticipatamente, e per far questo non deve far altro che digitare la parola "fine" nella stessa casella di testo in cui digita i valori da sommare. Una volta che il ciclo viene interrotto dall'istruzione break il controllo passa alle istruzioni immediatamente successive al blocco di codice del ciclo; in questo caso all'istruzione alert. L'uso del metodo parseInt() è necessario per convertire una variabile stringa come num nel valore intero del numero digitato dall'utente.

L'utilizzo di label o "etichette" con break

Dalla versione 1.2 del linguaggio, in JavaScript l'istruzione break può essere utilizzata con maggiore precisione. È necessario specificare, infatti, che quando si utilizza una istruzione break in un ciclo che a sua volta è annidato in un altro ciclo - che sia dello stesso tipo o no, è ininfluente - tale istruzione consente di uscire solo dal ciclo in cui se ne fa uso, ossia da quello più annidato. Ne consegue che, se vi fosse bisogno di uscire da tutti i cicli, si dovrebbe ripetere la stessa condizione in ciascuno di essi. JavaScript però possiede una soluzione molto "elegante" per ovviare all'inconveniente; in esso, a differenza di molti altri linguaggi di programmazione, inclusi quelli più strettamente affini, è possibile utilizzare una label - che in inglese sta per "etichetta" - per uscire in un sol colpo da tutta la struttura nidificata, indipendentemente dal suo livello di annidamento e senza correre il rischio di destrutturare il programma come accade in altri linguaggi se si utilizza una label con un'istruzione goto. Ciò è possibile facendo precedere al codice del primo ciclo - ovvero di quello più esterno - un identificatore alfanumerico seguito dal segno dei due punti (:). Successivamente quando si utilizza l'istruzione break sarà sufficiente far seguire il nome dell'etichetta per ottenere di uscire dall'intera struttura nidificata. Anche in questo caso il controllo del flusso passa alla prima istruzione immediatamente successiva al blocco di codice contrassegnato dalla label. L'esempio qui sotto mostra l'utilizzo di un'istruzione break seguita dal nome di un'etichetta per uscire da tre cicli nidificati.

```
<html>
  <title>Utilizzo di break con una label</title>
  <head>
    <script language="JavaScript">
      <!--
        cicli_Annidati:
        for (var i=1; i<=10;) {
          for (var num=""; i<=10;) {
            num=prompt('Introdurre un valore: ', '');
            for (var somma=0; i<=10; i++) {
              if (num.toLowerCase()=='fine') break cicli_Annidati;
              somma+=parseInt(num);
            }
          }
        }
        alert('La somma totale è: ' + somma);
      -->
    </script>
  </head>
```

Nel listato appena mostrato l'esempio precedente è stato "spezzettato" in tre cicli For nidificati. Nel ciclo più interno è stata utilizzata un'istruzione break seguita dal nome della label "cicli_Annidati" che è posta prima del codice dei tre cicli. Provando l'esempio ci si renderà conto di come l'utilizzo di una sola istruzione

break con il nome di un'"etichetta" permetta di uscire istantaneamente da un numero qualsiasi di cicli nidificati.

-L'istruzione *continue*

L'istruzione continue invece consente di anticipare forzatamente l'iterazione di un ciclo saltando le istruzioni che si trovano fra essa e la condizione di fine. L'esempio seguente è una variante del primo; in esso è stata aggiunta un'istruzione continue che viene eseguita nel caso che l'utente prema "OK" senza aver digitato alcun numero valido.

```
<html>
<title>Utilizzo di continue</title>
<head>
<script language="JavaScript">
<!--
  for (var i=1, somma=0, num=""; i<=10; i++) {
    num=prompt("Introdurre un valore: ','");
    if (num.toLowerCase()=='fine') break;
    if (isNaN(parseInt(num))) continue;
    somma+=parseInt(num);
  }
  alert('La somma totale è: ' + somma);
-->
</script>
</head>
```

In questo caso, se l'utente digitasse caratteri alfanumerici o se premesse invio senza digitare niente, il ciclo continuerebbe le sue iterazioni ma senza eseguire alcuna somma. L'istruzione posta dopo continue, in questo caso il codice della somma aritmetica, viene saltata e il controllo ritorna alla "testa" del ciclo che provvede ad eseguire il controllo della condizione e, se ancora questa non dovesse essere stata soddisfatta, a incrementare nuovamente la variabile contatore i. Per capire la differenza tra questo e il codice dell'esempio precedente si provi a digitare caratteri non validi sia in questo che nell'altro. Si potrà verificare come nel primo, mancando questo tipo di controllo, le somme vengano ugualmente eseguite ma alla fine il risultato non sarà valido come numero ma sarà NaN, acronimo di Not a Number. Il metodo isNaN, utilizzato in questo secondo esempio, permette appunto di eseguire un controllo del genere.

Una volta riprodotto, esaminato e verificato l'inconveniente che si presentava all'interno della pagina web, si sono presentati tre scenari di soluzione del problema:

- Disassemblare le istruzioni a cicli nidificati con conseguente tentativo di semplificazione delle istruzioni e riassetto del codice con soluzioni più "elegant";
- Utilizzo di linguaggio e codice PHP;
- Utilizzo di specifici controlli ASP.NET AJAX.

2.3.12 Controlli ASP.NET AJAX

I controlli ASP.NET AJAX consentono di creare una vasta gamma di comportamenti del client con un numero limitato o addirittura senza script client, ad esempio l'aggiornamento parziale della pagina (aggiornamento di parti selezionate della pagina anziché aggiornamento dell'intera pagina con un postback) e la visualizzazione dello stato di avanzamento dell'aggiornamento durante postback asincroni. Gli aggiornamenti asincroni parziali della pagina evitano il sovraccarico dei postback a pagina completa. Nello specifico si sarebbe potuto utilizzare il controllo ScriptManagerProxy.

Tramite il controllo ScriptManagerProxy è possibile utilizzare componenti nidificati, come pagine di contenuto e controlli utente, per aggiungere riferimenti a script e servizi in pagine nelle quali è già definito un controllo ScriptManager in un elemento padre.

Una pagina Web può contenere un solo controllo ScriptManager, direttamente nella pagina stessa o indirettamente all'interno di un componente nidificato o padre. Il controllo ScriptManagerProxy consente di aggiungere script e servizi a pagine di contenuto e controlli utenti nei casi in cui la pagina master o la pagina host contiene già un controllo ScriptManager.

Quando si utilizza il controllo ScriptManagerProxy è possibile estendere gli insiemi di script e servizi definiti dal controllo ScriptManager. Se non si desidera includere script e servizi specifici in tutte le pagine che includono un particolare controllo ScriptManager, è possibile rimuoverli dal controllo ScriptManager e aggiungerli nelle singole pagine utilizzando il controllo ScriptManagerProxy in alternativa.

Per il corretto funzionamento del controllo ScriptManagerProxy sono necessarie impostazioni specifiche in un file web.config. Se si tenta di utilizzare questo controllo e il sito Web non contiene il file web.config necessario, verranno segnalati errori nella visualizzazione della pagina nella posizione in cui avrebbe dovuto essere visualizzato il controllo. Se si fa clic sul controllo in tale stato nella visualizzazione Progettazione, in Microsoft Expression Web verrà offerta la possibilità di creare un nuovo file web.config o di aggiornare il file web.config esistente.

2.3.13 Linguaggio PHP

La soluzione di impiego di codice PHP che avrebbe permesso rapidamente di risolvere il problema specifico con poche istruzioni, è in realtà stata rapidamente scartata perché avrebbe comportato il riscrivere parti di codice nell'interfaccia con altre pagine originariamente create in Javascript.

Di seguito, per completezza, si riportano le istruzioni PHP di gestione delle tabelle dinamiche con cicli annidati:

Code

source

```
<style type="text/css">
```

```
    table td{
```

```
        padding:4px;
```

```
        border:1px solid #333;
```

```
    }
```

```
</style>
```

```
<style type="text/css">
```

```
    table td{
```

```
        padding:4px;
```

```
        border:1px solid #333;
```

```
    }
```

```
</style>
```

I cicli in php

code

source

```
<table>
```

```
    <?php
```

```
    // 8 righe
```

```
    for ($tr = 0; $tr < 8; $tr++) {
```

```
        echo "<tr>";
```

```
        // 6 colonne per ogni riga
```

```
        for ($td = 0; $td < 6; $td++) {
```

```
            echo "<td>row" . $tr . " col" . $td . "</td>";
```

```
        }
```

```
    echo "</tr>";  
  }  
  ?>  
</table>
```

La soluzione adottata è stata quella di riesaminare in dettaglio il codice già implementato, disaggregare le istruzioni per determinare in quale punto esatto e grado di nidificazione l'errore si presentava. Successivamente il tutto è stato riassembleato introducendo però alcune peculiari semplificazioni. Ulteriori controlli sul codice ASP hanno permesso di risolvere il problema di funzionalità nella composizione del substrato e conseguente calcolo della producibilità di biogas da FORSU anche in associazione ad altre biomasse (aggiornamento del 20 maggio 2013).

2.4 Verifica del calcolo e dei risultati a video

Sempre tramite il contatto aidainfo@enea.it, è stato segnalato e successivamente risolto un errore di calcolo di uno dei parametri della miscela nel caso si fosse scelto come percorso la digestione anaerobica. In particolare è risultato non corretto il calcolo della %CH₄ miscela, percentuale di metano all'interno della miscela, del substrato composto in fase di scelta della simulazione.

Le verifiche avviate a seguito della segnalazione hanno permesso la risoluzione rapida dell'inconveniente e di conseguenza la scelta di inserire tale parametro nella restituzione a video dei primi risultati, seppur intermedi (aggiornamento del 20 maggio 2013).

Definizione Substrato

Spec. di proc. e di impianto

Analisi Energetica

Analisi Economica

Analisi Ambientale

Pubblica amministrazione 1 - Definizione substrato

Comune di ARCHI - CHIETI (CH) - Regione ABRUZZO

L'utente può aggiungere ulteriori biomasse a quelle scelte nel comune tramite il seguente menù a tendina:

Aggiungi biomassa: quantità:

CIVILE		
FORSU		51,87
TOTALI		61,87

Parametri Miscela	
% ST	15,00 %su tel quale
% SV	25,00 %ST
C/N	27,00
NTK	1,82 %ST
Densità	0,80 t/m ³
%CH ₄ biogas	51 %

Il substrato selezionato per la digestione ha un contenuto di sostanza secca pari al 15% per cui si consiglia la tecnica di digestione **SERISECCA**.

È opportuno nel caso si utilizzi il trattamento a umido avere un contenuto di solidi inferiore al 12%,aggiungendo un adeguato quantitativo di acqua.

- CH₄ - Metano:**
- C/N - Rapporto Carbonio:** Azoto presente nella biomassa in genere i valori ritenuti normali sono compresi tra 13 e 20.
- CO₂ - Anidride carbonica:** biostato di Carbonio.
- CO₂ CH₄ -** Unità di misura convenzionale che permette di paragonare insieme emissioni di gas serra diversi (CH₄, NO₂, CO₂) con differenti potenziali di riscaldamento globale (GWP global warming potential).
- COSTI FISSI -** sono costi indipendenti dal volume della vendite o dei servizi.
- COSTI VARIABILI -** si intendono tutti quei costi che dipendono direttamente dal livello produttivo.
- CSTR - Completely Stirred Reactor = reattore completamente miscelato** che riceve un flusso continuo di biomassa da trattare in entrata, emettendo un flusso pure continuo di materiale trattato in uscita strutturato come un zaffirato (Necessaria dotazione di agitazione meccanica).
- CV - Certificati Verdi:** costituiscono una forma di incentivazione della produzione di energia elettrica da fonti rinnovabili. Sono emessi dal Gestore del Servizio Energetico (GSE (Gestore Servizi Energetici) su richiesta del produttore di energia da fonti rinnovabili. Sono titoli.

Figura 4. Schermata, aggiornata al 20 maggio 2013, modifica i parametri della miscela

2.5 Aggiornamento e stima dati biomasse

Ultima attività svolta è stata la pianificazione e l'avvio del percorso di aggiornamento della base dati AIDA, sia relativamente alle matrici che al potenziale di biomasse nel nostro paese. In particolare è stato realizzato il calcolo del potenziale territoriale di alcune biomasse residuali lignocellulosiche al fine di fornire agli utenti della piattaforma informazioni circa l'evoluzione del potenziale lordo e disponibile di biomasse [2]. L'analisi è stata realizzata per le biomasse residuali agricole, erbacee ed arboree, ed i risultati sono stati prodotti e resi disponibili sul sito di A.I.D.A. come mappe e set di tabelle di dati regionali basati sui dati statistici ISTAT 2011. Il calcolo è stato realizzato tramite la medesima metodologia di calcolo esplicitata nel progetto "Atlante nazionale delle biomasse" i cui dati sono attualmente alla base del percorso di analisi per l'utente "Pubblica Amministrazione". Tale aggiornamento, in una ottica di servizio all'utenza del servizio, dovrebbe essere reso continuo e realizzato, per tutte le biomasse comprese originariamente in ATLANTE, con un maggior dettaglio a scala territoriale. Sarebbe inoltre auspicabile poter ampliare le biomasse di scarto, incluse nel database di Atlante, aggiungendo matrici quali i residui di colture orticole o di prodotti agroindustriali quali, ad esempio, il siero di latte o le buccette di pomodoro che sono suscettibili di impiego negli impianti di digestione anaerobica.

I dati Atlante necessitano sia di un aggiornamento temporale sia di poter essere fruiti in modo diverso, da soggetti istituzionali e non, a prescindere dallo specifico impiego per le valutazioni di fattibilità proposte da A.I.D.A.

I dati forniti sono comunque frutto di stime e pertanto costituiscono un riferimento utile ed una prima indicazione delle disponibilità presunte nell'area territoriale prescelta che al massimo si spinge al territorio provinciale. Per una pianificazione e valutazione di dettaglio sarebbe auspicabile disporre di set di dati alla scala territoriale comunale e poter meglio valutare la fattibilità di impianti basati sulle risorse di una area di comuni scelta a priori. Uno sforzo in tal senso potrà essere realizzato nel prosieguo del lavoro.

2.5.1 Le biomasse residuali agricole

Tutte le elaborazioni qui presentate sono state prodotte impiegando la base dati ISTAT relativa all'annualità 2011 e resa disponibile, su base provinciale, dall'Istituto.

I dati di superficie e produzione sulle coltivazioni provengono dalle indagini PSN (Programma Statistico Nazionale) relative all'anno 2011 e sono dati in prevalenza di natura estimativa, cioè sono generati da pool di esperti a disposizione delle Regioni che inviano dati all'Istituto Nazionale di Statistica. Per quanto riguarda le superfici delle colture arboree l'Istat fornisce sia il dato della superficie totale che della superficie in produzione. La prima include anche le superfici nei primi anni di impianto prima della vera e propria entrata in produzione. Per le nostre elaborazioni abbiamo sempre impiegato il dato relativo alle superfici in produzione.

Nel caso dei seminativi l'Istat fornisce sia il dato di produzione totale che di produzione effettivamente raccolta e per le nostre elaborazioni sui sottoprodotti abbiamo sempre utilizzato il dato di produzione totale.

Elemento fondamentale della stima della quantità teorica di sottoprodotti di una coltivazione è l'indice SP/P che esprime la quantità di residuo (SP) prodotta per ogni unità di prodotto primario (P).

Come già ricordato questo valore può variare in virtù di molteplici fattori ed è necessario assumere un valore medio attendibile. Nel nostro caso, data la scala di riferimento e come già realizzato in altre analisi regionali e nazionali si è scelto un valore univoco, valido a livello nazionale, eccetto che per alcune colture arboree (vite ed olivo) per le quali è stato possibile disporre di valori differenziati a livello regionale.

Per quanto la stima delle quantità di sottoprodotti generati dal sistema agricolo possa essere accurata e realizzata in modo trasparente e coerente i dati ottenuti rappresentano un potenziale o una disponibilità teorica di tipo fisico. Una frazione, talvolta molto rilevante delle biomasse ottenute non è disponibile per altri usi in quanto l'azienda stessa ne fa un uso diretto come intermedio di produzione o ne trae un beneficio, commerciandola.

Giungere a definire quanta biomassa sia effettivamente disponibile per impieghi di natura energetica necessita di una dettagliata conoscenza degli usi tradizionali e dei relativi benefici economici associati. Per una disamina dei problemi relativi alla stima ed una descrizione di dettaglio della metodologia impiegata si rimanda al lavoro citato in Bibliografia [2].

Nel seguito descriviamo per sommi capi i principali risultati ottenuti ed, allo stato attuale, validati e pubblicati solo per l'ambito nazionale e regionale. Altre informazioni e dati sono riportati nell'apposita sezione della piattaforma A.I.D.A.

Le colture erbacee di pieno campo occupano la quota più rilevante della S.A.U. nazionale seguite dai prati pascoli e dalle colture arboree. Tra di esse le colture più diffuse sono i cereali che occupano oltre il 50% delle superfici, seguite dalle foraggere, le piante industriali, le colture ortive e le leguminose da granella. Nella nostra elaborazione abbiamo preso in considerazione le colture più diffuse trascurando quelle coltivazioni che per le esigue superfici investite, la distribuzione geografica e i residui prodotti appaiono poco significative. In particolare abbiamo analizzato 17 colture suddivise in 4 categorie: i cereali, le oleaginose, le piante da industria e le leguminose da granella (Tabella 1).

Tabella 1. Potenziale lordo e disponibile da residui agricoli erbacei nel 2011 espresso in kt

Categoria residui	Superficie coltivata ha	Potenziale lordo kt (s.s.)	Potenziale disponibile kt (s.s.)
Cereali	3 412 865	11 489,5	5 018,3
Oleaginose	302 813	557,7	403,0
Piante da industria	121 632	231,2	181,8
Leguminose	71 203	176	52,8
Totale	3 908 513	12 454,4	5 655,9

E' evidente che la cerealicoltura ed in particolare le coltivazioni di frumento mais ed orzo producono le quantità più elevate di residui suscettibili di un impiego energetico che, in virtù del limitato contenuto di umidità è preferibilmente indirizzato verso la combustione o la gassificazione.

Le colture permanenti: vite, olivo e fruttiferi occupano nel nostro paese una superficie di oltre 2,4 milioni di ettari e costituiscono un elemento distintivo del paesaggio. Sia durante la fase di coltivazione che nelle fasi di trasformazione dei prodotti primari si ottengono quantità significative di sottoprodotti di scarto che sono suscettibili di molteplici impieghi. In questa prima fase di analisi abbiamo limitato la nostra attenzione agli scarti legnosi prodotti nella fase agricola, trascurando i residui delle lavorazioni (sanse, raspi, vinaccioli, gusci). La metodologia di calcolo considera separatamente due tipologie di residui le potature legnose ottenute annualmente ed il legno prodotto a fine ciclo di vita della coltura quando è necessario procedere all'espianto per il rinnovo.

La produzione complessiva di scarti dalle colture arboree ammonta ad oltre 6 milioni di tonnellate prodotti su oltre 2,4 milioni di ettari di superficie con un valore medio di 2,5 tonnellate di scarto secco ad ettaro (Tabella 2). L'olivo è la coltura che produce le maggiori quantità di scarti seguito dalla vite e dagli agrumi. Vite ed Olivo insieme producono quasi il 60% degli scarti di potatura nazionali.

Tabella 2. Potenziale lordo e disponibile da residui agricoli legnosi nel 2011 espresso in kt

Culture residui	Superficie ha	Potenziale lordo kt (s.s.)	Potenziale disponibile kt (s.s.)
<i>Olivo</i>	1 137 048	2 018,2	1288,9
<i>Vite</i>	700 098	1 436,8	1079,7
<i>Fruttiferi</i>	256 796	1 112,7	526,3
<i>Frutta in guscio</i>	141 933	443,6	277,1
<i>Agrumi</i>	161 317	1 033,7	902,0
Totale	2 397 192	6 044,9	4073,8

Per completezza e dimostrazione riportiamo una delle carte regionali ottenute che mostra come alcune tipologie di residui siano fortemente regionalizzati in quanto, in virtù delle differenze climatiche, le colture arboree dalle quali si ottengono sono distribuite in modo non omogeneo.

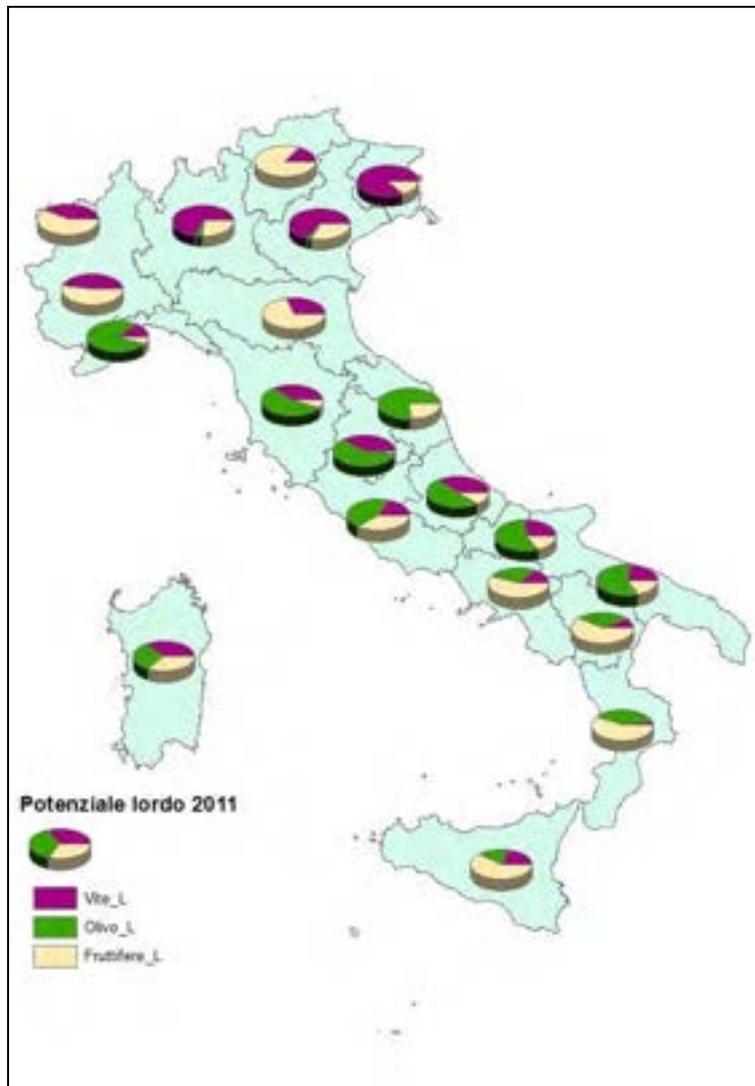


Figura 5. Distribuzione regionale delle diverse tipologie di residui legnosi

3 Conclusioni

La fase di test e successivi aggiornamenti hanno permesso di risolvere i problemi evidenziati e dato l'opportunità di fornire un servizio in continuo e progressivo miglioramento/aggiornamento.

Durante le attività di questa annualità è emerso un inconveniente/limitazione alle funzionalità attualmente implementate nell'applicazione web AIDA.

E' stata più volte ricordata la sequenza logica delle fasi della elaborazione di calcolo implementata nell'applicazione web AIDA tramite step successivi, all'interno di una singola pagina, per poi passare alla elaborazione di pagine seguenti. Conseguenza di ciò è l'impossibilità di utilizzare i pulsanti di spostamento tra le pagine web ma si devono necessariamente seguire le indicazioni presenti nel contenuto della pagina stessa.

In termini informatici si potrebbe riassumere questo concetto con un utilizzo di variabili di sessione ma non una memorizzazione/archiviazione delle stesse.

La mancata memorizzazione comporta sicuramente un'ulteriore elaborazione da parte del server (con conseguente riduzione dei volumi di *storage*) ma anche l'impossibilità di riprodurre la stessa elaborazione in modo immediato se non rieseguendo, punto per punto, gli *step* successivi introdotti e percorsi all'inizio della elaborazione.

Il gruppo di lavoro sta attualmente analizzando e lavorando su soluzioni parallele. Da una parte la necessità di fornire all'utente strumenti dedicati sia alla reportistica (es. file PDF) ma anche per permettere il salvataggio dei dati estrapolati, su file di tipo CSV, e consentire ulteriori elaborazioni o l'archiviazione dei risultati per confronti successivi.

Nello stesso tempo sia le tecnologie che gli elementi legislativi evolvono nel tempo e questo rende necessario intervenire in fasi successive per assicurare una corrispondenza con la realtà che si modifica. A titolo di esempio basta ricordare i cambiamenti introdotti dal legislatore italiano in termini di incentivi (conto termico) che si traducono in una esigenza immediata di modifica e aggiornamento non solo nel semplice contenuto esplicativo (realizzabile con pagine web di tipo statico, solo descrittive) ma anche della intera procedura di calcolo riguardante la valutazione economica della fattibilità di un intervento.

Per alcune delle attività sopracitate è stato valutato e pianificato il lavoro da fare in termini di entità di impegno in quanto alcune richiedono, se saranno ritenute necessarie e desiderate dall'utenza, una modifica profonda di alcune parti dell'architettura del software che richiedono una riscrittura di parti consistenti del codice software della piattaforma.

Completata l'analisi e la validazione del calcolo è adesso necessario dedicare una maggiore attenzione a quegli elementi che contribuiscono a fornire all'utenza un servizio qualitativamente valido attraverso elementi accessori quali set di dati più ampi sulle matrici e sulle biomasse ai fini di una valutazione più ampia e significativa della reale fattibilità di impianti legati al territorio come studi ed analisi di fattibilità realizzati con la piattaforma e resi disponibili agli utenti a titolo di esempio.

Riferimenti bibliografici

- [1] V. Cigolotti, N. Colonna, M.R. Mastrullo, A. Moreno, F. Reale, P. Regina e S. Scarfogliero, "A web tool for developing feasibility analysis of biomass plants" in proceedings of European Biomass Conference 3-7 June 2013, Copenhagen "21st European Biomass Conference and Exhibition". Eta Florence, Firenze. Pages 287-290
- [2] Colonna N., A. Macrì e P. Regina, "I sottoprodotti legnosi ed erbacei del settore agricolo italiano". Atti del Convegno: "I sottoprodotti Agroforestali e industriali a base rinnovabile". Progetto EXTRAVALORE, Ancona 26-27 settembre 2013. ISBN 978-88-906186 edito da CTI Milano