Agenzia nazionale per le nuove tecnologie,
l'energia e lo sviluppo economico sostenibile

MINISTERO DELLO SVILUPPO ECONOMICO

# Progettazione di strumento di calcolo per la redazione del bilancio energetico territoriale

A. Poggio, G. Cerino Abdin, A. Crocetta, L. Degiorgis, M. Noussan,
M. Fantino, G. Caragnano, M. Chiesa, F. Posca, P. Ruiu, R. Roberto

POLITECNICO DI TORINO
Dipartimento Energia

ISMB
Istituto Superiore Mario Boella

Report RdS/PAR2013/121

# PROGETTAZIONE DI STRUMENTO DI CALCOLO OPEN SOURCE PER LA REDAZIONE DEL BILANCIO ENERGETICO TERRITORIALE

A. Poggio, G. Cerino Abdin, A. Crocetta, L. Degiorgis, M. Noussan (Politecnico di Torino), M. Fantino, G. Caragnano, M. Chiesa, F. Posca, P. Ruiu (Istituto Superiore Mario Boella), R. Roberto (ENEA)

Settembre 2014

Report Ricerca di Sistema Elettrico
Accordo di Programma Ministero dello Sviluppo Economico - ENEA
Piano Annuale di Realizzazione 2013
Area :RAZIONALIZZAZIONE E RISPARMIO NELL'USO DELL'ENERGIA ELETTRICA
Progetto: Risparmio di energia elettrica nel settore civile, industria e servizi
Obiettivo: Studi e progettazione di interventi di efficienza energetica per le Pubbliche Amministrazioni
Responsabile del Progetto: Gaetano Fasano, ENEA

Il presente documento descrive le attività di ricerca svolte all'interno dell'Accordo di *collaborazione "Progettazione di strumento di calcolo open source per la redazione del bilancio energetico territoriale"*

Responsabile scientifico ENEA: Gaetano Fasano

Responsabile scientifico POLITO: Alberto Poggio.

# Indice


1 INTRODUZIONE ..... 5

Presupposti del lavoro ..... 5
Stato dell'arte della metodologia ..... 5
Finalità del lavoro ..... 6
Descrizione del gruppo di lavoro ..... 6

2 DESCRIZIONE DELLE ATTIVITÀ SVOLTE E RISULTATI ..... 8

2.1 ANALISI PRELIMINARE ..... 8
*2.1.1 Modello LEB (versione MS Excel)* ..... *8*
Motore di calcolo - LEBp ..... 8
Maschere di input - INPUTp ..... 9
*2.1.2 Formulazione della struttura* ..... *9*
*2.1.3 Formulazione del modello relazionale* ..... *9*
*2.1.4 Individuazione delle esigenze di rappresentazione grafica* ..... *10*
2.2 MODELLO DATI ..... 12
*2.2.1 Schema del database e relazioni* ..... *12*
*2.2.2 Tabelle* ..... *13*
Tabella vettore ..... 13
Tabella trasformazione ..... 14
Tabella famiglia ..... 15
Tabella quantità ..... 16
Tabella uso ..... 17
Tabella tipo_vettore ..... 18
Tabella consumo ..... 18
Tabella produzione ..... 19
*2.2.3 Interrogazioni del database (query)* ..... *20*
Query di popolamento ..... 20
Query di visualizzazione ..... 21
2.3 INTERFACCIA GRAFICA ..... 21
*2.3.1 Struttura generale del diagramma LEB* ..... *21*
Griglia di riferimento ..... 21
Codifica colore e dimensione degli elementi ..... 22
Posizionamento dinamico dei nodi ..... 22
*2.3.2 Costruzione del diagramma LEB* ..... *22*
Interrogazione del database ..... 22
Calcoli di posizionamento ..... 23
Generazione del file HTML ..... 24
*2.3.3 Interattività* ..... *24*
Navigazione (zoom, spostamenti, filtri) ..... 24
Mouseover su righe e colonne ..... 27
Mouseover su linee ..... 27
Mouseover su nodi ..... 27
Click su nodi ..... 27
2.4 STRUMENTI UTILIZZATI ..... 30
*2.4.1 Modello dati* ..... *30*
MySql ..... 30
Cloud e Virtualizzazione ..... 31
PhpMyAdmin ..... 32
Flyspeed SQL query ..... 33
*2.4.2 Interfaccia grafica* ..... *34*
HTML 5 ..... 34
SVG ..... 34
2.5 TEST DI APPLICAZIONE ..... 35

*2.5.1 Regione Piemonte* ..... *35*
Bilancio energetico regionale anno 2008 (storico) ..... 35
*2.5.2 Provincia di Torino* ..... *36*
Bilancio energetico provinciale anno 2011 (storico) ..... 36
Bilancio energetico provinciale anno 2020 (scenario previsionale) ..... 37

3 CONCLUSIONI ..... 40

Risultati del lavoro ..... 40
Sviluppi potenziali ..... 40

4 RIFERIMENTI BIBLIOGRAFICI ..... 42

APPENDICE – PIATTAFORMA SOFTWARE DI GESTIONE DEL DATABASE ..... 43

Queries utilizzate per il popolamento del database - MySQL (listato) ..... 43
Sorgente HTML della pagina web di restituzione grafica (listato) ..... 43

APPENDICE – CURRICULUM SCIENTIFICO DEL GRUPPO DI LAVORO ..... 84

# 1 Introduzione

### *Presupposti del lavoro*

Nell'ambito della Ricerca di Sistema Elettrico, una prima fase di attività svolta nel PAR 2012 ha riguardato la definizione di dettaglio di una metodologia innovativa per la costruzione e l'analisi di bilanci energetici territoriali denominata Local Energy Balance (LEB). Questo lavoro aveva preso spunto dai risultati del Progetto Strategico Interreg Alcotra RENERFOR [1].
Le conclusioni del lavoro hanno messo in evidenza le rilevanti potenzialità offerte dalla metodologia LEB rispetto alle statistiche energetiche attualmente disponibili [2].
L'articolata struttura del modello è stata sviluppata in modo da poter rispondere completamente alle crescenti esigenze di statistica e pianificazione energetica locale determinate dagli obiettivi delle politiche comunitarie [3]. Analogamente, il modello è risultato adeguato a supportare statistiche e analisi settoriali quali, in particolare, quelle relative ai consumi finali del settore terziario pubblico.
L'analisi svolta ha evidenziato come il grado di dettaglio richiesto sia significativamente più elevato rispetto a quello attualmente offerto dalla statistiche energetica convenzionale. Le verifiche condotte hanno indicato come una parte significativa dei dati necessari è già disponibile ed è attualmente utilizzata per la redazione dei Bilanci Energetici Regionali ENEA. Sono però necessarie integrazioni con ulteriori informazioni, al fine di raggiungere il grado di dettaglio richiesto per la quantificazione delle grandezze necessarie a definire gli obiettivi europei 2020 del Pacchetto Clima ed Energia.

### *Stato dell'arte della metodologia*

LEB rappresenta il sistema energetico del territorio in esame attraverso uno schema fisico dei flussi energetici e delle fasi di produzione, trasformazione, reti di trasporto e distribuzione, consumi dell'energia. Tutte le fonti e i vettori energetici sono singolarmente individuati da specifici flussi e nodi. Ciascun flusso è tracciato in termini di contenuto di energia rinnovabile e di emissioni di $CO_2$ originata dal suo processo di generazione. I consumi relativi ai principali aggregati sono evidenziati da specifici contatori.
Rispetto alle metodologie convenzionali attualmente in uso, LEB offre una serie di elementi innovativi, quali:

- una rappresentazione dettagliata ed esaustiva degli elementi maggiormente impattati dalle esigenze di analisi e pianificazione, quali quelli inerenti le energie rinnovabili, la cogenerazione e la trigenerazione, la generazione distribuita presso gli utenti con i relativi scambi bidirezionali con le reti, il teleriscaldamento e il teleraffrescamento
- un'impostazione di calcolo che consente analisi pluriennali combinando dati storici e scenari previsionali, con relative verifiche del posizionamento rispetto agli obiettivi 2020 (riduzione dei consumi complessivi di energia, quota di energia da fonti rinnovabili per usi finali, emissioni di gas serra)
- la garanzia di coerenza intrinseca al modello di calcolo tra tutti i dati e risultati, anche in presenza di molteplici variazioni associate alle diverse ipotesi di scenario previsionale

Al livello di sviluppo raggiunto con le attività PAR 2012, LEB è implementato attraverso un sistema di fogli di calcolo MS Excel. In questa versione preliminare, LEB ha già trovato significative applicazioni nella pianificazione energetica locale.
La Provincia di Torino (ora Città Metropolitana) ha adottato stabilmente il LEB per la redazione e l'analisi dei propri bilanci energetici provinciali [4][5][6] che, grazie allo strumento, sono disponibili nella serie storica completa 2000 - 2011. Analogamente, il LEB è stato applicato alla predisposizione degli scenari previsionali 2020 contenuti nel nuovo Piano di Azione per l'Energia Sostenibile recentemente adottato dall'ente [7].

La Regione Valle d'Aosta ha utilizzato il LEB in occasione dell'ultimo aggiornamento del proprio bilancio energetico regionale [8][9].

### *Finalità del lavoro*

Il lavoro svolto e la relativa esperienza conseguita hanno messo in evidenza, da parte degli operatori della pianificazione energetica locale, alcuni elementi:

- una richiesta crescente di utilizzo del LEB per riuscire a tracciare il dettaglio richiesto dall'applicazione degli obiettivi energetici comunitari
- una difficoltà a gestire la conseguente complessità intrinseca al modello di bilancio energetico territoriale adottato
- l'esigenza di poter utilizzare la metodologia LEB attraverso uno strumento di gestione dei dati con maggiori caratteristiche di interazione con l'utente

Queste sollecitazioni hanno orientato gli obiettivi delle attività PAR 2013 sulla progettazione e la realizzazione prototipale di uno strumento software open source per l'implementazione della metodologia LEB. Tale strumento è costituito da due componenti:

- un database di riferimento, strutturato per rappresentare in modo completo i dati di un generico sistema energetico territoriale
- un'interfaccia grafica per la gestione del database e la rappresentazione delle relative informazioni

Gli elementi emersi nei test condotti nella precedente di fase di lavoro hanno permesso di individuare i principali criteri per la definizione dell'architettura del database e della sua interfaccia, come di seguito elencato:

- una struttura di calcolo costruita in funzione della disponibilità attuale di informazioni energetiche dirette da parte dei soggetti preposti alla redazione di bilanci energetici
- la predisposizione per l'effettuazione di integrazioni con ulteriori informazioni energetiche dirette potenzialmente disponibili alla pubblica amministrazione o stime indirette definibili da altre informazioni territoriali
- l'applicabilità dello strumento a differenti livelli di scala territoriale (nazionale, regionale, comunale, fino al bilancio energetico di un singolo ente pubblico), con un'organizzazione dei dati che ne consente l'integrazione in un unico sistema informativo

Pertanto le attività sono state finalizzate allo sviluppo dello strumento software, secondo le fasi di lavoro di seguito riepilogate:

- la reingegnerizzazione del precedente modello LEB (versione MS Excel) e la conseguente formulazione dell'organizzazione del database
- la progettazione del modello dati e dell'interfaccia grafica, attraverso la realizzazione (a livello di prototipo) dei relativi codici di calcolo e gestione
- la conduzione di alcuni test applicativi per la verifica della corretta funzionalità del prototipo

### *Descrizione del gruppo di lavoro*

La metodologia LEB è stata sviluppata, nel progetto RENERFOR e nel PAR 2012 della Ricerca di Sistema Elettrico, dal Gruppo di Ricerca in Sistemi per l'Energia e l'Ambiente del Dipartimento Energia del Politecnico di Torino in collaborazione con il Laboratorio Biomasse e Sistemi Energetici dell'Unità Tecnica Tecnologie Saluggia di Enea.
Le specificità delle attività svolte nel PAR 2013 hanno richiesto ulteriori apporti di competenze rispetto a quelle competenze già presenti in materia di statistica e modellistica energetica. Il gruppo di lavoro iniziale è stato integrato da alcuni colleghi dell'Istituto Superiore Mario Boella, l'unità di ricerca congiunta con il Politecnico di Torino dedicata alle Tecnologie dell'Informazione e della Comunicazione (ICT). Questo

supporto ha reso disponibili capacità operative relative alla progettazione di infrastrutture di calcolo per grandi quantità di dati (tra cui le tecnologie di Cloud Computing) e di sistemi di interazione con l'utente.

# 2 Descrizione delle attività svolte e risultati

L'attività si è concentrata sullo sviluppo dello strumento software LEB, nelle sue due componenti: il database e l'interfaccia grafica.
In primo luogo è stata svolta un'ampia analisi preliminare per fissare i principali criteri progettuali e definire la struttura di massima dello strumento, a partire dallo studio di dettaglio del modello LEB nella sua precedente versione (MS Excel).
La seconda parte di attività è stata dedicata all'elaborazione del modello che ospita la base dati (modello dati), attraverso la progettazione dei suoi elementi costitutivi, delle reciproche relazioni funzionali e delle funzioni di interrogazione.
In sinergia con la definizione del modello dati, la terza parte di attività ha riguardato lo studio delle dell'interfaccia grafica, prima attraverso una disamina tra differenti alternative di rappresentazione e successivamente con la costruzione del diagramma di flusso energetico.
Entrambe le attività di sviluppo hanno prodotto una versione prototipale dei codici di calcolo e gestione dello strumento, sia per quanto concerne il database che l'interfaccia grafica. Nell'ultima parte di attività sono stati svolti una serie di prove di applicazione del prototipo sui bilanci energetici di due casi-studio: la Regione Piemonte e la Provincia di Torino.
Nel seguito è riportata una descrizione di dettaglio delle attività sopracitate.

## *2.1 Analisi preliminare*

### 2.1.1 Modello LEB (versione MS Excel)

Per poter effettuare una progettazione adeguata sia per quanto riguarda la visualizzazione grafica, sia per l'implementazione del modello e della base dati è stato condotto uno studio sul funzionamento del LEB. Nella versione attuale MS Excel(R) [1], il software risulta implementato attraverso l'ausilio di diverse cartelle di lavoro, che sono di seguito elencate:

1. LEBp.xlsx, che costituisce il motore di calcolo
2. INPUTp.xlsx, nel quale sono riportate le maschere per il caricamento dei dati

Nel corso dell'analisi iniziale, lo studio delle due cartelle è risultato determinante per la successiva progettazione del modello dati.

#### *Motore di calcolo - LEBp*

La cartella di lavoro LEBp include il motore di calcolo del LEB e la sua rappresentazione grafica, nonché alcuni fogli di supporto, come le definizioni dei flussi, i poteri calorifici, ecc. In questa cartella di lavoro sono stati sviluppati tutti i calcoli dei flussi energetici necessari nella versione a consuntivo per la redazione di un bilancio energetico locale.
La cartella di lavoro è collegata ad altre cartelle, che hanno il compito di gestire i dati di input e di output richiesti e forniti agli utenti.

[1] Durante la fase di analisi e le successive è stata utilizzata la versione di MS Excel(R) 2010.

*Maschere di input - INPUTp*

La cartella di lavoro INPUTp raccoglie tutte le maschere di input da fornire all'utente, sviluppato a supporto del motore di calcolo. Questo foglio raccoglie i dati di input necessari ai vari livelli per il calcolo delle informazioni mancanti. Tipici dati di input sono i consumi energetici delle trasformazioni e le relative energie prodotte, mentre il sistema calcola efficienze medie di esercizio. In alcuni casi risulta invece necessario stimare delle efficienze medie del parco energetico (ad esempio nel caso di trasformazioni presso gli usi finali) al fine di calcolare le energie prodotte, che non sono disponibili come dato misurato.
In questa prima fase di sviluppo del database, sono state sviluppate delle routine in VBA per l'estrazione automatica dei dati dalla cartella di input per il popolamento di alcuni record del database. Questa fase di migrazione si è resa necessaria per una verifica dei calcoli ed un confronto con la struttura attuale del LEB. In una fase successiva verranno sviluppate apposite maschere di input al fine di svincolare lo strumento dalla struttura sviluppata in Excel(R) attualmente in essere.

### 2.1.2 Formulazione della struttura

Studiando come sono stati organizzati i dati all'interno del Foglio di Lavoro LEB contenuto in LEBp, sono state individuate le entità e le relazioni di partenza che consentono la modellizzazione della base dati [10]. Da un'analisi preliminare eseguita su questo foglio risulta chiaro che esso rappresenta un insieme di tre livelli logici:

- Livello I: Dati energetici; in questo livello ci sono i valori energetici rappresentati staticamente, in modo aggregato o come calcoli all'interno di tabelle;
- Livello II: rappresentazione grafica del flusso lungo un percorso in cui i vettori energetici attraversano i nodi di trasformazione, di bilancio o di meter;
- Livello III: Suddivisione del LEB in due macro segmenti:
    - Supply (sezione a sinistra del meter centrale)
    - Usi Finali (sezione a destra del meter centrale)

La classificazione delle informazioni in diversi livelli rende possibile separare logicamente il contenuto del foglio LEB in modo che:

- il livello I sia utilizzato per definire il modello dati del database, le query e le relazioni;
- il livello II aiuti il lavoro di definizione del layout grafico da esporre attraverso una interfaccia web;
- il livello III sia utile per la progettazione del database e per lo sviluppo della grafica.

### 2.1.3 Formulazione del modello relazionale

A partire dall'analisi effettuata sull'attuale Modello LEB (versione MS Excel) e sui dati di input è stato possibile individuare due entità fondamentali da cui costruire il nuovo modello relazionale:

- flussi di vettori energetici
- nodi, che indicano trasformazioni, conversioni, bilanci o contatori

L'entità principale del modello sviluppato è il flusso, il quale rappresenta una quantità riferita ad una determinata area geografica (regione, provincia, comune). Tale entità può assumere significati differenti in relazione ad un determinato vettore energetico: quantità di energia, quota di energia rinnovabile oppure quantità di emissioni di CO2.
Una seconda entità fondamentale è costituita dai nodi che possono assumere il significato di trasformazioni oppure di bilanci. Essi sono rappresentati da blocchi grafici attraversati, sia in ingresso che in uscita, da flussi di vettori (Figura 1). Il significato dei flussi in input e output al nodo dipende dalla tipologia dello stesso.

Nei nodi di tipo trasformazione (trasformazioni meccaniche, trasformazioni chimiche, conversioni energetiche), i flussi di vettori in input rappresentano i consumi della trasformazione mentre quelli in output rappresentano le relative produzioni, ciascuna caratterizzata da una specifica efficienza di trasformazione.
Nel caso di nodi di tipo bilancio riferiti ad un singolo vettore energetico (bilanci di singoli vettori), i flussi di vettori in input (produzioni interne, importazioni, esportazioni, variazioni stoccaggi, perdite di trasporto e distribuzione) si sommano algebricamente per determinare la quantità utilizzata nell'area geografica in esame (consumo interno).
Infine i nodi di tipo bilancio in cui confluiscono differenti vettori energetici, sono riferiti ad una specifica tipologia di uso finale (contatori di consumi) della quale conteggiano la quantità consumata.
La caratterizzazione svolta del modello relazionale costituisce il riferimento per l'articolazione della struttura database in tabelle che rappresenti le entità fondamentali sopradescritte.

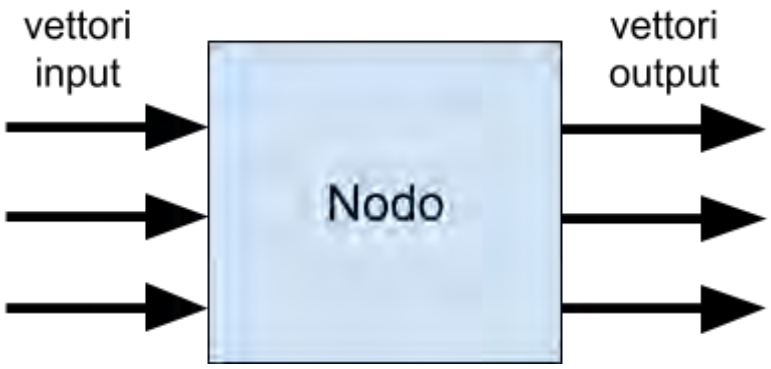



**Figura 1. Rappresentazione concettuale di un nodo di tipo trasformazione nel modello LEB**

### 2.1.4 Individuazione delle esigenze di rappresentazione grafica

I diagrammi di flusso (o grafi) costituiscono una delle principali modalità di visualizzazione applicate alla rappresentazione grafica di informazioni contenute in tabelle di dati.
Il modello LEB nella sua precedente versione (MS Excel) presenta una rappresentazione grafica basata su un diagramma di flusso statico che non presenta alcuna forma di interazione, sia verso l'utente che in relazione alla natura e all'entità dei dati inseriti. In diretta analogia con quanto già evidenziato nell'analisi del modello relazionale, in tale visualizzazione si distinguono due elementi principali: i nodi e le connessioni tra i nodi stessi. Con altra terminologia, le connessioni e i nodi possono essere indicati rispettivamente come flussi (a quali corrispondono i vettori energetici) e blocchi (che rappresentano le trasformazioni, le conversioni, i bilanci e i contatori).
Rispetto a questa modalità preesistente, l'analisi preliminare ha permesso di individuare:

- l'esigenza prioritaria di una nuova interfaccia grafica dinamica e interattiva
- i requisiti del nuovo strumento di rappresentazione, da finalizzare ad una più intuitiva visualizzazione dei dati LEB, mantenendo però contestualmente la disponibilità di una rilevante quantità di informazioni e di dettagli

Analizzando lo stato dell'arte nel campo della visualizzazione di dati energetici, emerge come le rappresentazioni più comunemente utilizzate siano il diagramma di flusso e il diagramma di Sankey (talvolta denominato anche come diagramma fiume), spesso impiegate per indicare i flussi di energia o di materiali scambiati tra gli elementi che costituiscono un sistema più o meno complesso.

Il diagramma di Sankey è costituito da un insieme di linee la cui larghezza è proporzionale alla quantità di energia che passa da una parte all'altra del sistema. Sono presenti diverse visualizzazioni Sankey, molte delle quali sono statiche, o create per la stampa. Altre invece presentano diversi livelli di interazione. In Figura 2 sono riportati due esempi recenti, tratti da diagrammi proposti dall'International Energy Agency (IEA) [11] e dal Canada Energy Systems Analysis Research (CESAR) dell'Università di Calgary [12].

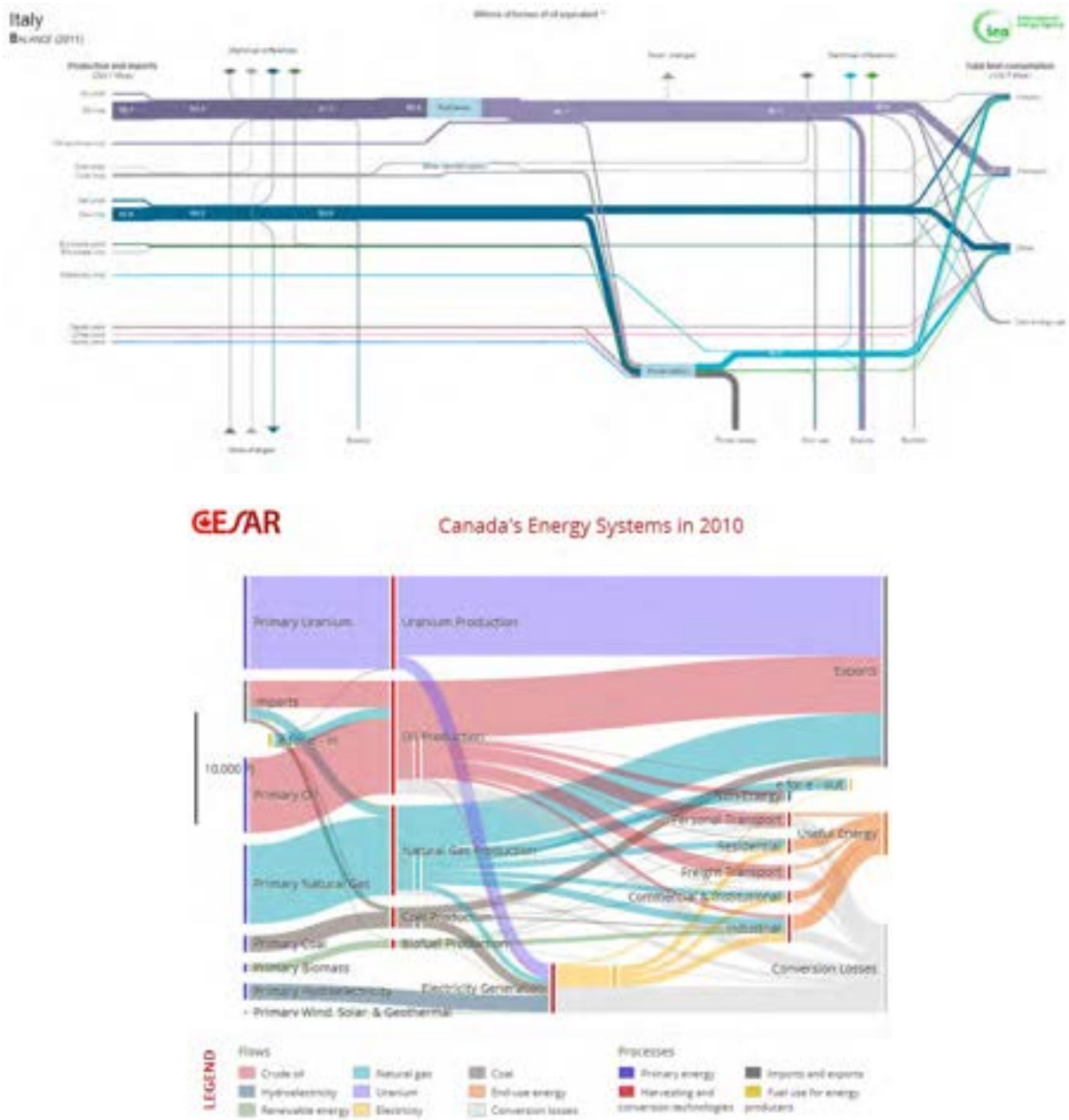



**Figura 2. Esempi di diagramma di Sankey applicato alla rappresentazione di dati energetici**

Malgrado l'ampia varietà di tipologie presenti, sia in letteratura che tra i software disponibili, nessuna delle rappresentazioni esaminate è però in grado di soddisfare completamente le esigenze esplicitate per la visualizzazione dei dati LEB. In particolare (come si nota negli esempi proposti) vi sono difformità nella struttura dello schema di flusso; inoltre le rappresentazioni sono spesso sintetiche, ignorando così il livello di dettaglio informativo che il LEB è in grado di offrire.

L'indagine svolta ha indicato pertanto che, in termini generali, la tipologia del diagramma di Sankey presenta le caratteristiche richieste per la rappresentazione dei dati contenuti nel database LEB. Risulta però necessario uno sviluppo di uno specifico diagramma di flusso con struttura idonea a visualizzare integralmente il modello LEB di bilancio energetico.
La soluzione scelta è stata pertanto quella di mantenere la struttura generale e la topologia dello schema di visualizzazione precedentemente implementato nella versione preesistente (MS Excel). Su questa base si è quindi effettuato lo studio degli elementi di interattività da introdurre. Contestualmente si è proceduto all'individuazione dell'ambiente di sviluppo in un'applicazione web-based, al fine di semplificare il più possibile l'accesso allo strumento e il suo impiego tramite un normale browser internet, senza la necessità di software aggiuntivi.

## 2.2 Modello dati

### 2.2.1 Schema del database e relazioni

Gli esiti dello studio sull'articolazione della modello LEB, svolto nell'analisi preliminare, hanno fornito i presupposti per la costruzione dell'architettura generale da seguire come riferimento nella costruzione del database relazionale. In Figura 3 è riportato lo schema, con l'indicazione dei suoi elementi costitutivi e delle reciproche relazioni.

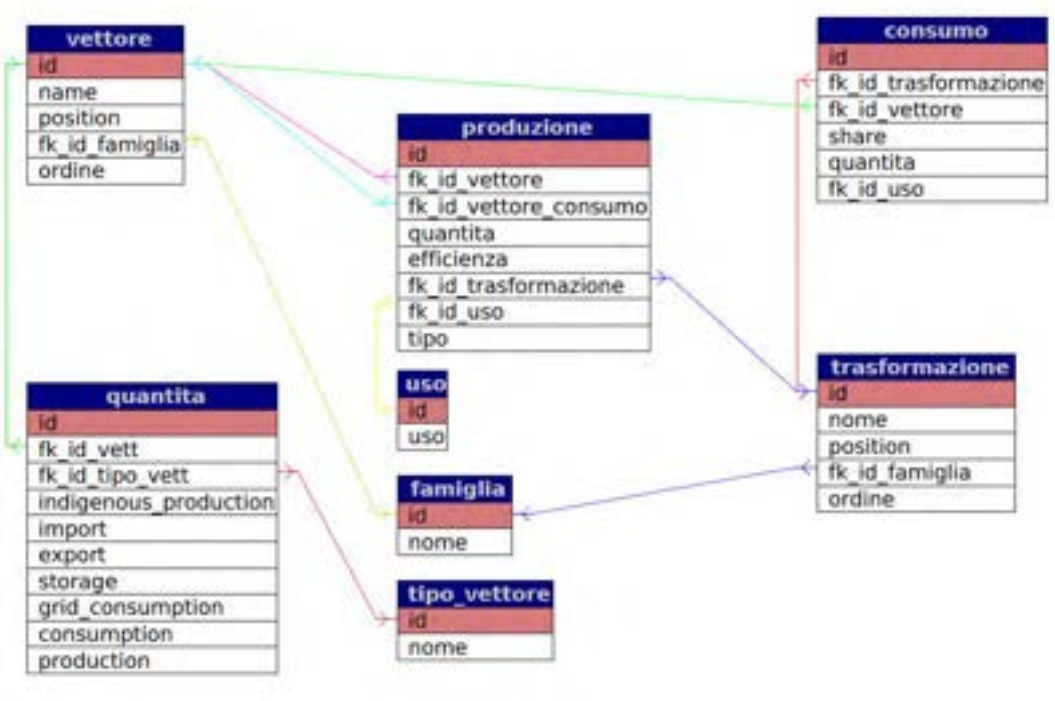



**Figura 3. Schema del database e relazioni**

### 2.2.2 Tabelle

La struttura del database è composta da una serie di tabelle che permettono di modellizzare le entità fondamentali del sistema LEB. Queste tabelle ospitano i dati provenienti dagli input e permettono di calcolare altri valori attraverso specifiche query. Le tabelle possono essere di tipo statico o dinamico:

- tabelle statiche: contengono valori fissi nel tempo che vengono inseriti all'inizio della vita del database e vengono modificati raramente: tipicamente le modifiche su questi record vengono effettuate da utenti con privilegi di amministratore; le dimensioni di tali tabelle sono perciò fisse nel tempo
- tabelle dinamiche: tipicamente vengono popolate importando i dati da file di input o attraverso l'elaborazione o la combinazione di dati già presenti nel database; le tabelle sono popolate dagli utenti che possiedono gli input; in questo caso le dimensioni sono variabili nel tempo

Di seguito vengono descritte le tabelle attualmente presenti nel database e i campi che esse contengono. Per ciascun campo vengono riportate le caratteristiche quali, il nome, il tipo di dato e altre caratteristiche utili per la memorizzazione del dato.

#### *Tabella vettore*

La tabella vettore (Tabella 1) raccoglie le informazioni sui vettori energetici rappresentati nel LEB. La tabella contiene una riga per ciascun vettore energetico di cui si vuole tenere traccia.

**Tabella 1. Struttura della tabella vettore del database**

| # | Campo | Tipo | Collation | Attributi | Null | Predefinito | Extra |
|---|---|---|---|---|---|---|---|
| 1 | id | int(10) | | UNSIGNED | No | Nessuno | AUTO_INCREMENT |
| 2 | name | text | utf8_general_ci | | No | Nessuno | |
| 3 | position | int(11) | | | No | Nessuno | |
| 4 | fk_id_famiglia | int(11) | | | No | Nessuno | |
| 5 | ordine | int(11) | | | No | Nessuno | |

I vettori energetici inseriti nel database sono i seguenti:

| | |
|---|---|
| crude oil | fuel oil |
| natural gas | liquefied petroleum gas |
| natural gas liquids | patent fuel |
| liquefied natural gas | coke oven coke |
| coal | coke oven gas |
| nuclear | syngas |
| energy crops & forestry | nuclear fuel |
| waste | log wood |

| | |
|---|---|
| hydro | compressed wood |
| sea | chipped wood |
| wind | cold extracted biooil |
| sun | biogasoline |
| geo/aero/hydro-thermal | biodiesel |
| electricity | sewage sludge & other biogas |
| heat | landfill biogas |
| cool | biogas |
| motor gasoline | biomethane |
| aviation fuels | refuse derived fuel |
| kerosene | energy |
| gas/diesel oil | |

La tabella è di tipo statico poiché i valori contenuti saranno modificati molto raramente. I campi della tabella sono i seguenti:

- id - identificativo univoco del vettore, è generato in modo automatico
- name - nome del vettore inserito da input
- position - colonna in cui posizionare il nodo di bilancio del vettore (seguendo la notazione precedentemente adottata nel LEB, assume valori quali: 10, 21, 22, 23, ecc.)
- fk_id_famiglia - identificativo della famiglia di appartenenza, costituisce il collegamento con la tabella famiglia
- ordine - ordine del nodo di bilancio del vettore all'interno della famiglia

### *Tabella trasformazione*

La tabella trasformazione (Tabella 2) descrive i nodi trasformazione.

**Tabella 2. Struttura della tabella trasformazione del database**

| # | Campo | Tipo | Collation | Attributi | Null | Predefinito | Extra |
|---|---|---|---|---|---|---|---|
| 1 | id | int(10) | | UNSIGNED | No | Nessuno | AUTO_INCREMENT |
| 2 | nome | text | utf8_general_ci | | No | Nessuno | |
| 3 | position | int(11) | | | No | Nessuno | |
| 4 | fk_id_famiglia | int(11) | | | No | Nessuno | |
| 5 | ordine | int(11) | | | No | Nessuno | |

I valori contenuti sono di tipo statico e sono i seguenti:

| | |
|---|---|
| REGASIFICATION | SOLAR THERMODYNAMIC |
| PATENT FUEL PRODUCTION | SOLAR |
| WOOD TRANSFORMATION | WIND TURBINES |
| BIOOIL COLD EXTRACTON | SEA TURBINES |
| RDF PRODUCTION | HYDROELECTRIC |
| CRUDE OIL REFINERY | NUCLEOTHERMOELECTRIC |
| COKE OVENS | THERMOELECTRIC |
| GASIFICATION | COGENERATION |
| NUCLEAR FUEL PRODUCTION | THERMAL |
| BIOREFINERY | DISTRICT HEATING |
| ANAEROBIC DIGESTION | COMPRESSION REFRIGERATION |
| GEOTHERMOELECTRIC | ABSORPTION REFRIGERATION |
| GEOTHERMAL | BALANCE |
| COMPRESSION HEAT PUMPS | METER |
| ABSORPTION HEAT PUMPS | METER FINAL GRID SUPPLIES |
| PHOTOVOLTAIC | METER ULTIMATE USES |

I campi contenuti sono i seguenti:

- id - identificativo univoco del vettore, è generato in modo automatico
- nome - nome della trasformazione
- position - colonna in cui posizionare il nodo di bilancio del vettore (seguendo la notazione precedentemente adottata nel LEB, assume valori quali: 10, 21, 22, 23, ecc.)
- fk_id_famiglia - identificativo della famiglia di appartenenza, costituisce il collegamento con la tabella famiglia
- ordine - ordine del nodo di bilancio del vettore all'interno della famiglia

### *Tabella famiglia*

La tabella famiglia (Tabella 3) è una tabella statica che riporta le famiglie dei vettori rappresentati.

**Tabella 3. Struttura della tabella famiglia del database**

| # | Campo | Tipo | Collation | Attributi | Null | Predefinito | Extra |
|---|---|---|---|---|---|---|---|
| 1 | id | int(10) | | | No | Nessuno | AUTO_INCREMENT |
| 2 | nome | varchar(60) | utf8_general_ci | | No | Nessuno | |

Le famiglie possono contenere diversi vettori energetici e identificano la posizione verticale (rappresentata come una riga orizzontale nel diagramma LEB) in cui posizionare graficamente i nodi. I valori contenuti sono i seguenti:

| | |
|---|---|
| electricity | waste |
| heat | hydro |
| cool | sea |
| oil | wind |
| natural gas | solar |
| coal | geo/aero/hydro-thermal |
| nuclear | various |
| biomass | |

La tabella contiene i seguenti campi:

- id - identificativo univoco del vettore, è generato in modo automatico
- nome - nome della famiglia

### *Tabella quantità*

La tabella quantità (Tabella 4) raccoglie i valori di quantità dei vettori energetici di tipo energia, rinnovabile e CO2. Si tratta di una tabella dinamica, in quanto una parte dei valori contenuti è raccolta come dato di input, mentre altri valori sono il risultato di calcoli svolti a partire da ulteriori dati del database.
I valori contenuti sono i seguenti:

- indigenous production (input)
- import (input)
- export (input)
- storage (input)
- grid_consumption (input)
- consumption (calcolato)
- production (calcolato)

I campi della tabella sono i seguenti:

- id - identificativo univoco del vettore, è generato in modo automatico
- fk_id_vett - identificativo del vettore, è il collegamento con la tabella vettore
- fk_id_tipo_vett - tipo di quantità, è il collegamento con la tabella tipo_vettore
- indigenous_production - quantità di energia prodotta localmente per un dato vettore
- import - quantità di energia importata per un dato vettore
- export - quantità di energia esportata per un dato vettore
- storage – variazione della quantità di energia negli stoccaggi per un dato vettore
- grid_consumption - perdite di energia per il transito su reti di trasporto e distribuzione per un dato vettore
- consumption - quantità di energia totale consumata per un dato vettore
- production - quantità di energia totale prodotta per un dato vettore

**Tabella 4. Struttura della tabella quantità del database**

| # | Campo | Tipo | Collation | Attributi | Null | Predefinito | Extra |
|---|---|---|---|---|---|---|---|
| 1 | id | int(10) | | UNSIGNED | No | Nessuno | AUTO_INCREMENT |
| 2 | fk_id_vett | int(10) | | | No | Nessuno | |
| 3 | fk_id_tipo_vett | int(10) | | | No | Nessuno | |
| 4 | indigenous_production | float | | | No | Nessuno | |
| 5 | import | float | | | No | Nessuno | |
| 6 | export | float | | | No | Nessuno | |
| 7 | storage | float | | | No | Nessuno | |
| 8 | grid_consumption | float | | | No | Nessuno | |
| 9 | consumption | float | | | No | Nessuno | |
| 10 | production | float | | | No | Nessuno | |

### *Tabella uso*

La tabella uso (Tabella 5) è di tipo statico e contiene i valori relativi all'uso di un determinato nodo.
I valori possono essere:

| | |
|---|---|
| SUPPLY - MECHANICAL | COMMERCIAL AND PUBLIC SERVICES / OTHER |
| SUPPLY - CHIMICAL | AGRICULTURE&FORESTRY / FISHING |
| SUPPLY - ENERGY CONVERSION | INDUSTRY |
| RESIDENTIAL | TRANSPORT |

**Tabella 5. Struttura della tabella uso del database**

| # | Campo | Tipo | Collation | Attributi | Null | Predefinito | Extra |
|---|---|---|---|---|---|---|---|
| 1 | id | int(11) | | | No | Nessuno | AUTO_INCREMENT |
| 2 | uso | varchar(100) | utf8_general_ci | | No | Nessuno | |

I primi tre valori indicano la fornitura del vettore energetico, suddiviso in funzione della sua provenienza (trasformazioni meccaniche, trasformazioni chimiche, conversioni energetiche). Gli ultimi cinque valori indicano l'uso finale dei vettori, suddiviso per settore di utilizzo (residenziale, terziario, agricoltura, industria, trasporti).
I campi contenuti nella tabella sono i seguenti:

- id - identificativo univoco dell'uso, è generato in modo automatico
- uso - nome dell'uso

### *Tabella tipo_vettore*

La tabella tipo_vettore (Tabella 6) è una tabella statica che contiene l'informazione utile per l'identificazione del tipo di quantità inserita nella tabella quantità.

**Tabella 6. Struttura della tabella tipo_vettore del database**

| # | Campo | Tipo | Collation | Attributi | Null | Predefinito | Extra |
|---|---|---|---|---|---|---|---|
| 1 | id | int(11) | | | No | Nessuno | AUTO_INCREMENT |
| 2 | nome | varchar(100) | utf8_general_ci | | No | Nessuno | |

I valori possono essere:

energia

energia rinnovabile

emissioni di CO2

I campi contenuti nella tabella sono i seguenti:

- id - identificativo univoco del tipo di vettore, è generato in modo automatico
- uso - nome del tipo di vettore

### *Tabella consumo*

La tabella consumo (Tabella 7) contiene i valori di consumo di un dato vettore in un determinato nodo di trasformazione. La tabella è di tipo dinamico ed è in stretta relazione con la tabella trasformazione e con la tabella produzione.

**Tabella 7. Struttura della tabella consumo del database**

| # | Campo | Tipo | Collation | Attributi | Null | Predefinito | Extra |
|---|---|---|---|---|---|---|---|
| 1 | id | int(10) | | UNSIGNED | No | Nessuno | AUTO_INCREMENT |
| 2 | fk_id_trasformazione | int(10) | | UNSIGNED | No | Nessuno | |
| 3 | fk_id_vettore | int(10) | | UNSIGNED | No | Nessuno | |
| 4 | share | double | | UNSIGNED | No | Nessuno | |
| 5 | quantita | float | | | No | Nessuno | |
| 6 | fk_id_uso | int(11) | | | No | Nessuno | |
| 7 | ren | float | | | No | Nessuno | |

I campi contenuti nella tabella sono i seguenti:

- id - identificativo univoco del consumo, è generato in modo automatico
- fk_id_trasformazione - identificativo della trasformazione cui è riferito il consumo
- fk_id_vettore - identificativo del vettore consumato
- share - quota del consumo in questione rispetto al consumo complessivo della trasformazione
- quantita - quantità di energia consumata
- fk_id_uso - identificativo del tipo di uso del vettore
- ren - quantità di energia rinnovabile associata al consumo

### *Tabella produzione*

**La tabella produzione (**

Tabella 8) contiene i valori di produzione di un dato vettore in un determinato nodo di trasformazione. La tabella è di tipo dinamico ed è in stretta relazione con la tabella trasformazione e con la tabella consumo.
I campi contenuti nella tabella sono i seguenti:

- id - identificativo univoco della produzione, è generato in modo automatico
- fk_id_trasformazione - identificativo della trasformazione cui è riferita la produzione
- fk_id_vettore - identificativo del vettore prodotto
- fk_id_vettore_consumo - identificativo del vettore consumato per produrre la quantità di energia in esame
- quantita - quantità di energia prodotta
- efficienza – valore dell'efficienza di trasformazione o conversione, espresso come rapporto tra quantità di energia prodotta e quantità di energia consumata
- fk_id_uso - identificativo del tipo di uso del vettore
- tipo - descrizione del record che indica il tipo di produzione (esempio: indigenous production, final production, gross electricity production); è utilizzato per discriminare le quantità da utilizzare in alcune query
- ren - quantità di energia rinnovabile

**Tabella 8. Struttura della tabella produzione del database**

| # | Campo | Tipo | Collation | Attributi | Null | Predefinito | Extra |
|---|---|---|---|---|---|---|---|
| 1 | id | int(10) | | UNSIGNED | No | Nessuno | AUTO_INCREMENT |
| 2 | fk_id_vettore | int(10) | | UNSIGNED | No | Nessuno | |
| 3 | fk_id_vettore_consumo | int(11) | | | No | Nessuno | |
| 4 | quantita | float | | | No | Nessuno | |
| 5 | efficienza | double | | | No | Nessuno | |
| 6 | fk_id_trasformazione | int(11) | | | No | Nessuno | |
| 7 | fk_id_uso | int(11) | | | No | Nessuno | |
| 8 | tipo | varchar(150) | utf8_general_ci | | No | Nessuno | |
| 7 | ren | float | | | No | Nessuno | |

### 2.2.3 Interrogazioni del database (query)

Una volta caricati tutti i dati di input (provenienti dalle apposite tabelle LEB di inserimento) è possibile effettuare una serie di interrogazioni al database (query) con lo scopo di combinare le informazioni presenti al fine di generare nuovi dati. Le query sono state scritte utilizzando il linguaggio SQL e sono di due tipi:

- query di popolamento - sono interrogazioni che permettono di generare nuovi dati che saranno inseriti in campi di tabelle già esistenti o in nuove tabelle
- query di visualizzazione - estraggono o generano dati utilizzati per la visualizzazione grafica

Di seguito un elenco delle principali query realizzate per il progetto e collezionate in file con estensione *.sql.

*Query di popolamento*

- **ins_ren_quant.sql:** inserisce nella tabella consumo le quantità di energia rinnovabile a partire dalla quantità di energia. Il valore può assumere valori 1 (vettore interamente rinnovabile), 0 (vettore interamente non rinnovabile) oppure il risultato di un calcolo effettuato sulla base dei contributi di energia rinnovabile presenti negli altri vettori che concorrono alla produzione del vettore in questione
- **upd_eff_prod.sql:** calcola e aggiorna il valore del campo efficienza della tabella produzione
- **upd_ren_cons.sql:** calcola e aggiorna il valore del campo rinnovabili della tabella consumo
- **upd_ren_prod.sql:** calcola e aggiorna il valore del campo rinnovabili della tabella produzione
- **upd_tot_cons_quant_energ.sql:** calcola e aggiorna il valore del campo consumo della tabella quantità sommando tutti i consumi per un dato vettore in ingresso alle trasformazioni
- **upd_tot_prod_quant_energ.sql:** calcola e aggiorna il valore del campo produzione della tabella quantità sommando tutte le produzioni per un dato vettore in uscita dalle trasformazioni
- **upd_tot_cons_quant_ren.sql:** calcola e aggiorna il valore del campo consumo della tabella quantità sommando tutti i consumi rinnovabili per un dato vettore in ingresso alle trasformazioni

- **upd_tot_prod_quant_ren.sql:** calcola e aggiorna il valore del campo produzione della tabella quantità sommando tutte le produzioni rinnovabili per un dato vettore in uscita dalle trasformazioni

*Query di visualizzazione*

- **Bilancio.sql:** estrae le posizioni di tutti i nodi di bilancio
- **final_grid_supply_meter.sql:** estrae i collegamenti dei contatori degli usi finali
- **link_final_production.sql:** estrae le uscite delle trasformazioni degli usi finali
- **link_final_trasformation.sql:** estrae i collegamenti tra le trasformazioni degli usi finali e il contatore centrale
- **link_meter.sql:** estrae i collegamenti tra il contatore centrale e i bilanci dei singoli vettori
- **link_meter_final.sql:** estrae i collegamenti tra il contatore centrale e i contatori degli usi finali
- **link_primacolonna.sql:** estrae i collegamenti tra i dati di input e i bilanci dei vettori primari
- **nodi_final_transformation.sql:** estrae i nodi di trasformazione da visualizzare negli usi finali
- **posizioni.sql:** estrae le posizione di tutti i nodi da visualizzare
- **trasformazioni.sql:** estrae i nodi di trasformazione da inserire nel diagramma LEB e le relative posizioni
- **trasformazioni_input.sql:** estrae le quantità consumate e prodotte da tutti i nodi di trasformazione del diagramma LEB
- **uses.sql:** estrae i nomi degli usi
- **vettori.sql:** estrae i nomi di tutti i vettori

## *2.3 Interfaccia grafica*

### 2.3.1 Struttura generale del diagramma LEB

Come definito a seguito dell'analisi preliminare, la rappresentazione grafica dei dati LEB è stata sviluppata secondo la tipologia del diagramma di Sankey, a partire dalla modalità di visualizzazione della versione preesistente (MS Excel).
Il diagramma LEB (Figura 4) è stato costruito sulla base di quattro elementi costitutivi, ai quali corrispondono precisi significati descrittivi e di quantificazione numerica:

- righe, che definiscono la famiglia di vettori
- colonne, che definiscono la tipologia dei nodi
- nodi, rappresentati da rettangoli del colore della famiglia alla quale appartengono
- linee, di spessore pari alla quantità del flusso e del colore del vettore che rappresentano

*Griglia di riferimento*

La topologia di base del diagramma segue quella della visualizzazione preesistente (MS Excel).
Una volta identificati i differenti vettori energetici, questi sono aggregati nelle rispettive famiglie di appartenenza (esempio: prodotti petroliferi, biomasse, energia elettrica). A ciascuna famiglia di vettori è associata una riga della matrice nella quale sono rappresentate le linee (flussi di vettori) e i nodi (trasformazioni, conversioni, bilanci, contatori) ad essa riferiti.

Analogamente i nodi sono disposti in colonne differenti, secondo la tipologia della trasformazione o conversione da essi rappresentata.
In questo modo è stato possibile definire un sistema di riferimento secondo una griglia o matrice utile a definire le posizioni specifiche (in termini di riga e colonna) dei diversi elementi del diagramma.
All'interno della griglia sono stati predisposti una serie di spazi orizzontali e verticali da mantenere liberi da nodi (trasformazioni, conversioni, bilanci, contatori). Tali spazi costituiscono i canali nei quali far transitare le linee (flussi di vettori) in modo da evitare sovrapposizioni tra elementi del diagramma.

#### *Codifica colore e dimensione degli elementi*

L'aspetto del diagramma LEB è stato curato in modo che sia possibile identificare a primo impatto i diversi elementi.
In particolare le famiglie di vettori (linee) e le relative trasformazioni e bilanci (nodi) sono contraddistinti da colori identificativi, scelti secondo una codifica unica per tutti i diagrammi LEB.
Inoltre ciascuna linea (flusso di vettore) è caratterizzata da uno spessore variabile, dimensionato in proporzione alla quantità di energia rappresentata. Analogamente variano anche le altezze dei blocchi che indicano i nodi (trasformazioni, conversioni, bilanci, contatori), dimensionate in proporzione alla quantità complessiva di energia che attraversa il nodo (ovvero la somma dei flussi in ingresso).
Questa modalità di visualizzazione, peculiare dei diagrammi di Sankey, permette di apprezzare andamento e variazioni delle quantità di energia lungo tutto il diagramma LEB.

#### *Posizionamento dinamico dei nodi*

Ciascuna linea è definita da un nodo di partenza, un nodo di arrivo e uno spessore variabile. in un singolo nodo possono confluire più linee, ciascuna delle quali deve poter sviluppare un proprio percorso visibile nel diagramma LEB e senza ingenerare sovrapposizioni o accavallamenti dei percorsi.
La scelta di utilizzare elementi di dimensioni variabili ha pertanto comportato l'implementazione di un posizionamento dinamico dei nodi. Per ogni sessione di rappresentazione grafica, la posizione finale di ogni singolo nodo è il risultato di un calcolo in funzione della dimensione delle linee in ingresso e uscita ai nodi presenti nelle colonne che precedono quella in cui è collocato il nodo corrente.

### 2.3.2 Costruzione del diagramma LEB

La costruzione del diagramma LEB e la sua successiva visualizzazione sono state implementate realizzando una procedura articolata nelle fasi di seguito illustrate.

#### *Interrogazione del database*

In primo luogo si procede all'effettuazione delle interrogazioni al database, per ottenere le informazioni inerenti i seguenti elementi:

- le righe (famiglie di vettori) e le colonne (tipologie di trasformazioni) che definiscono la griglia di riferimento del diagramma
- l'elenco dei nodi (trasformazioni, conversioni, bilanci, contatori) e delle linee (flussi di vettori) da rappresentare nel diagramma

**Figura 4. Visualizzazione globale del diagramma LEB**

### *Calcoli di posizionamento*

Sulla base delle informazioni dal database si effettuano i calcoli per il posizionamento dei nodi. Per ciascuno dei nodi sono determinati i seguenti elementi:

- la collocazione del nodo all'interno della griglia di riferimento del diagramma, in termini di riga e colonna
- l'altezza del blocco che rappresenta il nodo, come somma delle dimensioni delle linee che confluiscono al nodo

Quindi si calcolano gli spazi da assegnare ai canali di transito predisposti nella griglia, individuando le linee con percorsi di attraversamento tra colonne o righe e sommandone le relative dimensioni. I risultati di

questi calcoli consentono il distanziamento reciproco dei vari nodi, sia in orizzontale che in verticale, al fine di evitare sovrapposizioni delle linee.

#### *Generazione del file HTML*

Definiti i calcoli relativi alle posizioni dei vari elementi del diagramma, si passa alla generazione del relativo file HTML.
In primo luogo si procede all'inserimento dei nodi nel diagramma, attraverso l'aggiunta di un elemento <rect> per ciascuno di essi secondo il posizionamento calcolato.
Successivamente è possibile generare le linee che devono connettere i nodi. Per ciascuna di esse si procede a definire il percorso, determinando tutti i singoli punti di passaggio che lo costituiscono. Ogni linea è inserita nel diagramma attraverso l'aggiunta di un elemento <path> che ha come attributo "d" il percorso definito.
Contestualmente all'inserimento di nodi e linee, a ciascuno di questi sono assegnati i relativi eventi interattivi (meglio illustrati nel seguito della presente trattazione).

### 2.3.3 Interattività

#### *Navigazione (zoom, spostamenti, filtri)*

Nella parte superiore della pagina di visualizzazione è stata inserita una sezione dedicata alla navigazione del diagramma LEB (Figura 5). Da qui è possibile ridimensionare il diagramma effettuando variazioni di scala (pulsanti "+" e "-"); inoltre è possibile effettuare spostamenti all'interno del diagramma mediante traslazioni (frecce direzionali).

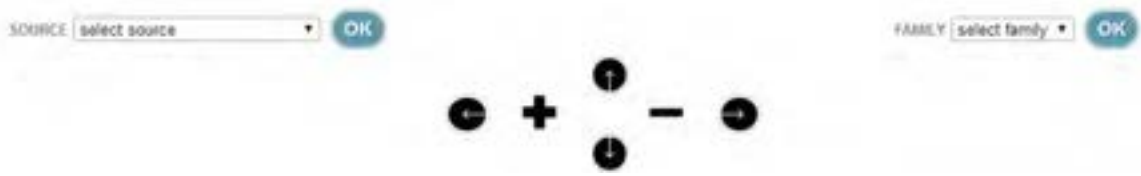
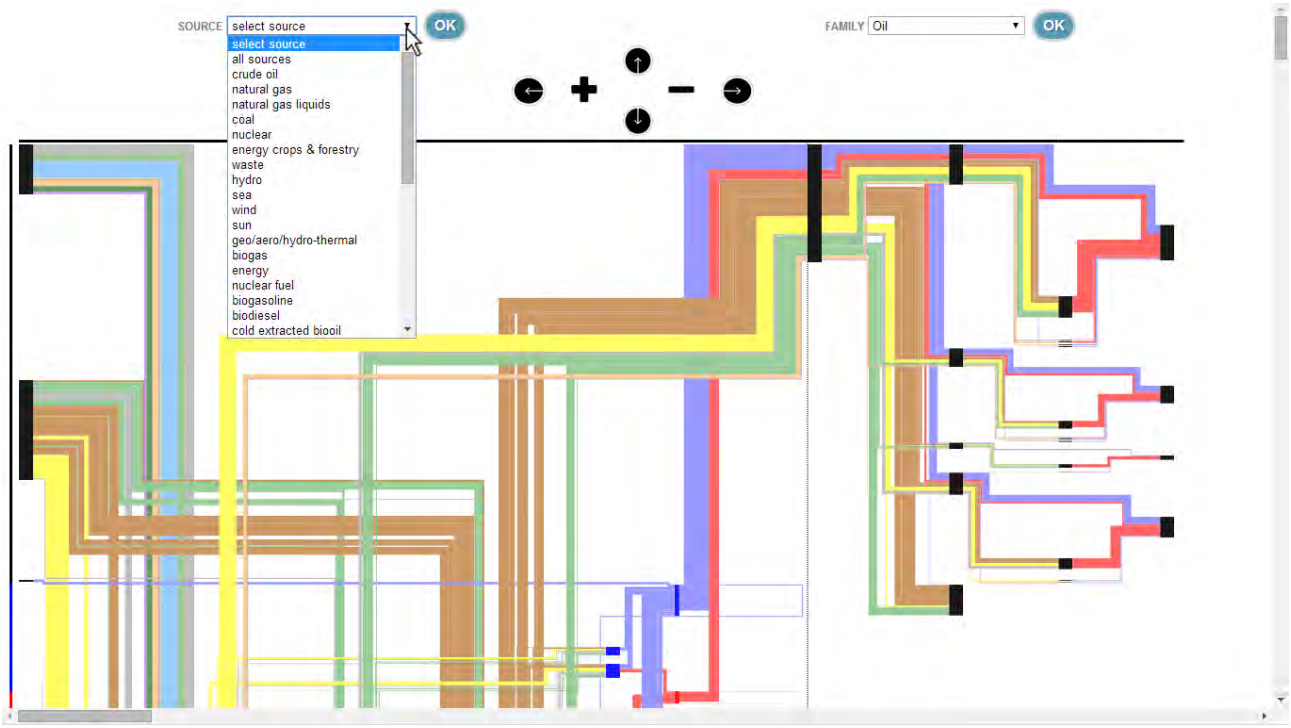



**Figura 5. Sezione navigazione del diagramma LEB**

In prossimità della sezione di navigazione sono inoltre presenti due filtri di visualizzazione per la selezione di specifiche porzioni o componenti del diagramma LEB:

- filtro vettore (in alto a sinistra): permette di selezionare un determinato vettore e mettere in evidenza i flussi ad esso relativi (Figura 6)
- filtro famiglia (in alto a destra): permette di selezionare una determinata famiglia di vettori e mettere in evidenza i flussi che vi appartengono (Figura 7)



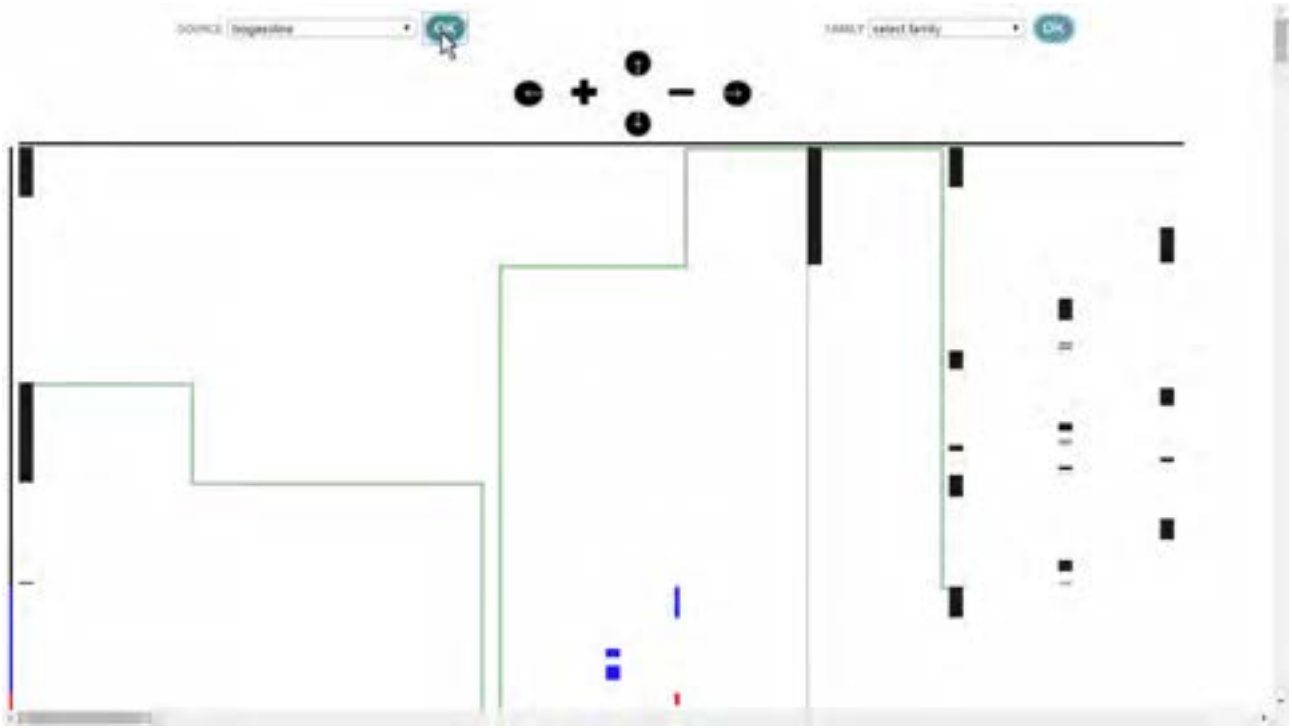



Figura 6. Filtro selezione vettore del diagramma LEB; esempio selezione vettore "biogasoline"

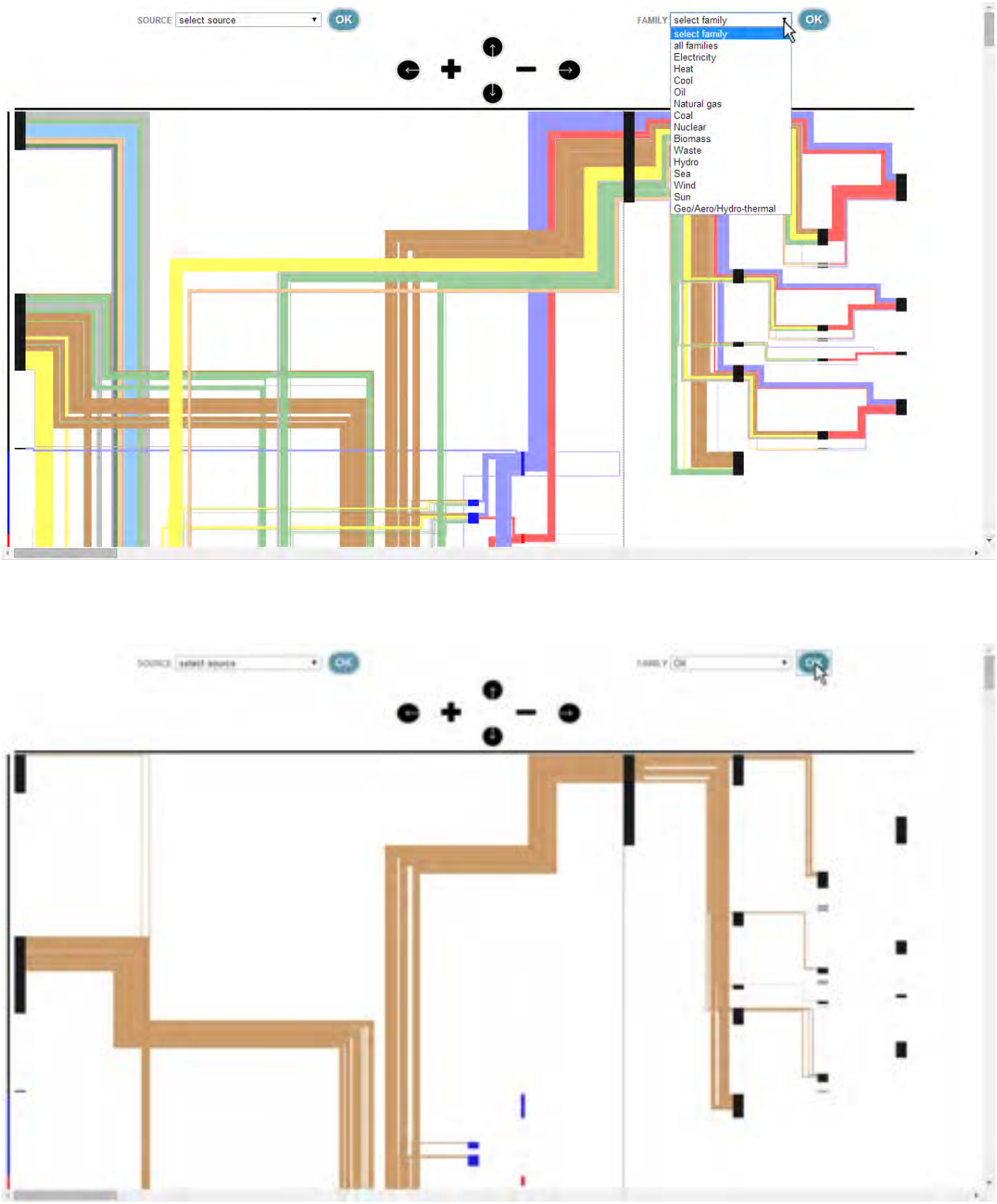



**Figura 7. Filtro selezione famiglia del diagramma LEB; esempio selezione famiglia "Oil"**

### *Mouseover su righe e colonne*

Le righe (famiglie di vettori) e le colonne (tipologie di trasformazioni) sono identificate da blocchi di intestazione, ai quali è stato assegnato un evento di mouseover. Al passaggio del puntatore del mouse su una colonna, è visualizzato in trasparenza un rettangolo grigio di selezione dell'intera colonna e il nome della stessa (Figura 8). Allo stesso modo, per una riga è visualizzato un rettangolo di selezione di colore corrispondente a quello della famiglia in questione e il suo nome (Figura 9).

### *Mouseover su linee*

Ad ogni linea (flusso di vettore) è stato assegnato un evento di mouseover. Al passaggio del puntatore del mouse su una linea, tutte le altre linee sono poste in semitrasparenza evidenziando così solo il flusso interessato (Figura 10). Nel medesimo evento è inoltre inclusa la visualizzazione di un tooltip contenente le informazioni inerenti il flusso in questione:

- il tipo di vettore
- la natura, in termini di produzione o consumo, con la relativa provenienza o destinazione
- la quantità rappresentata

### *Mouseover su nodi*

Analogamente alle linee, è stato assegnato un evento di mouseover anche ad ogni nodo. Al passaggio del puntatore del mouse su un nodo (Figura 11) si visualizza un tooltip contenente alcune informazioni sul nodo in questione:

- il nome del nodo
- un'icona identificativa della trasformazione o conversione rappresentata dal nodo

### *Click su nodi*

Oltre all'evento di mouseover, ai nodi è stato inoltre assegnato un evento di click. Cliccando con il puntatore del mouse su un nodo (Figura 12) si visualizza un pop-up contenete informazioni inerenti il nodo, quali:

- il nome del nodo
- la quantità totale di energia elaborata nella trasformazione o conversione rappresentata dal nodo
- una tabella di consumo, contenente tutte le informazioni (vettore, quantità, nodo di provenienza) riguardanti i flussi utilizzati nel nodo in questione
- una tabella di produzione, contenente tutte le informazioni (vettore, quantità, nodo di destinazione) riguardanti i flussi prodotti dal nodo in questione
- un diagramma a torta che indica l'efficienza e la perdita della trasformazione o conversione rappresentata dal nodo

Nel medesimo evento di click su nodo sono evidenziati i flussi connessi al nodo in questione (in ingresso o uscita), ponendo in semitrasparenza tutti gli altri flussi e nodi.

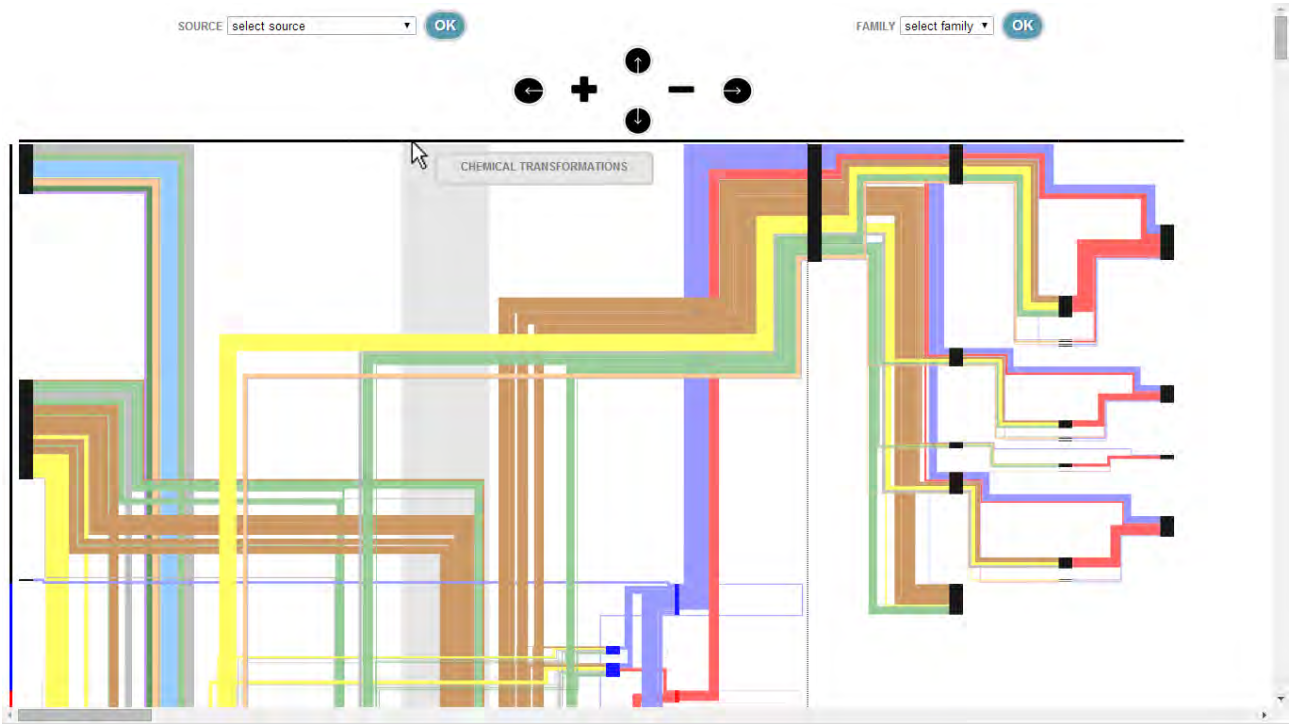



Figura 8. Esempio di mouseover su colonna nel diagramma LEB

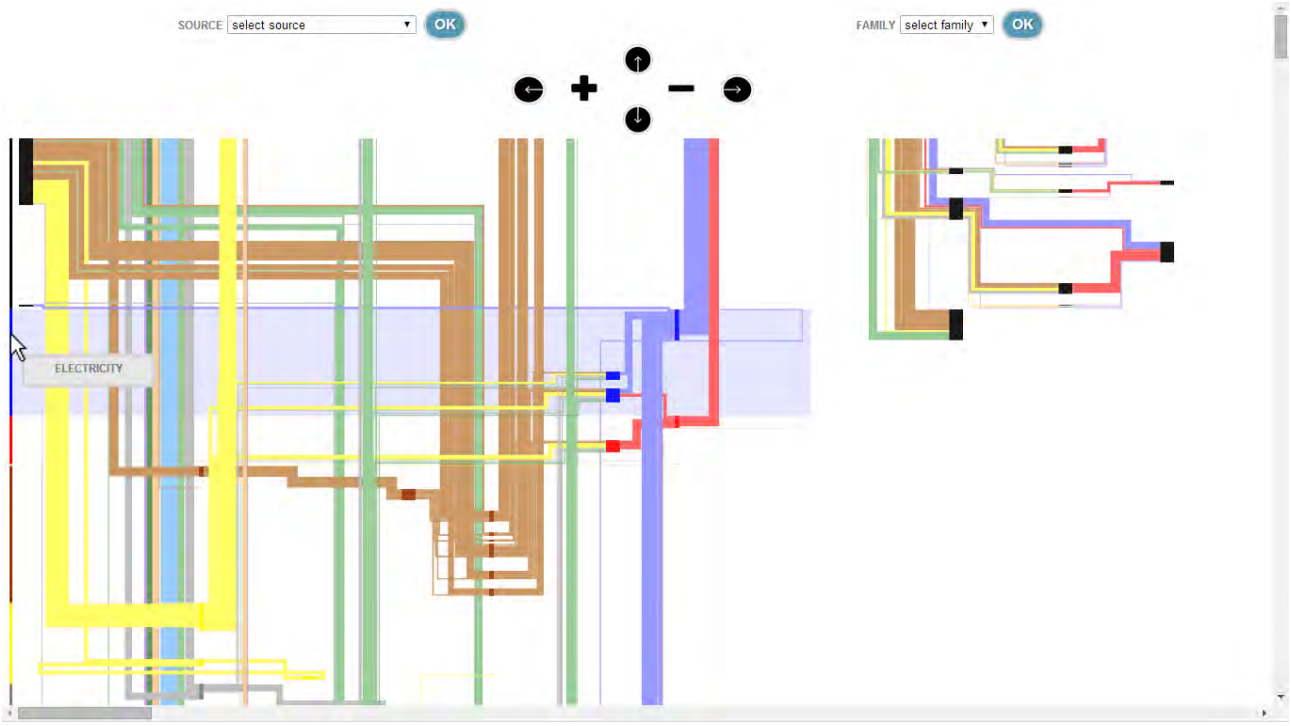



Figura 9. Esempio di mouseover su riga nel diagramma LEB

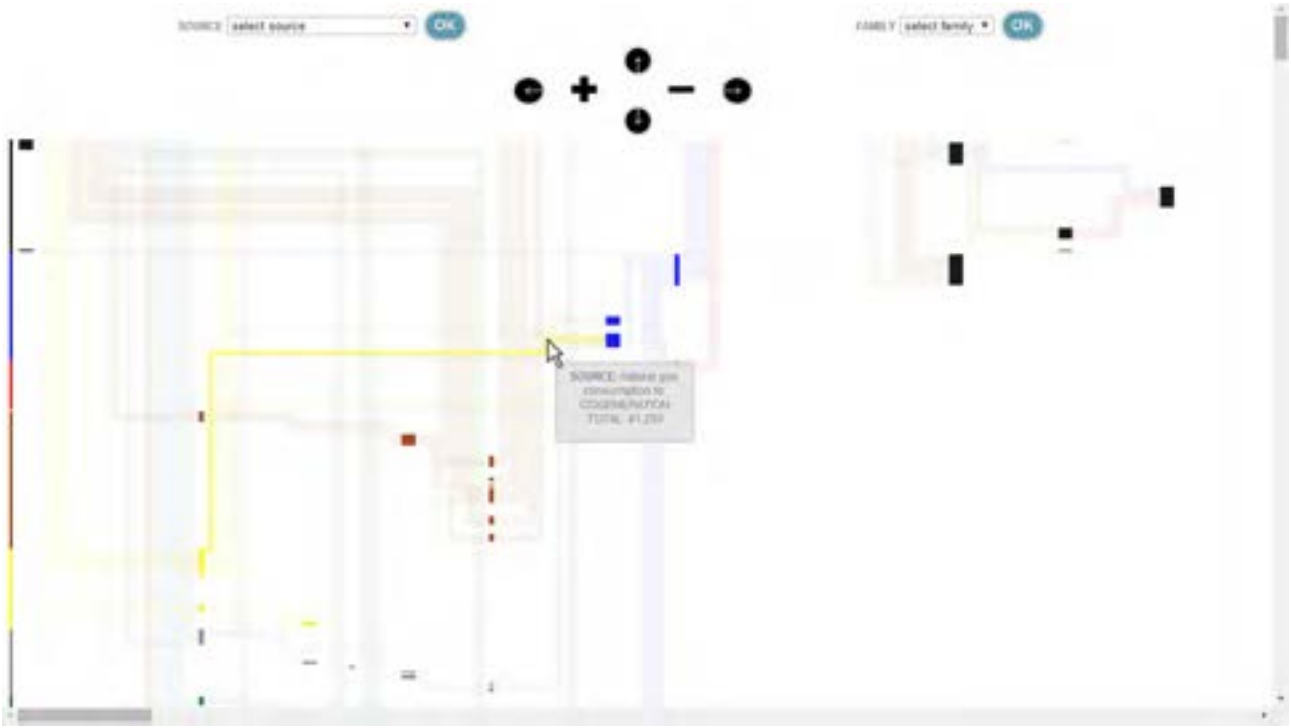


Figura 10. Esempio di mouseover su linea nel diagramma LEB

Figura 11. Esempio di mouseover su nodo nel diagramma LEB

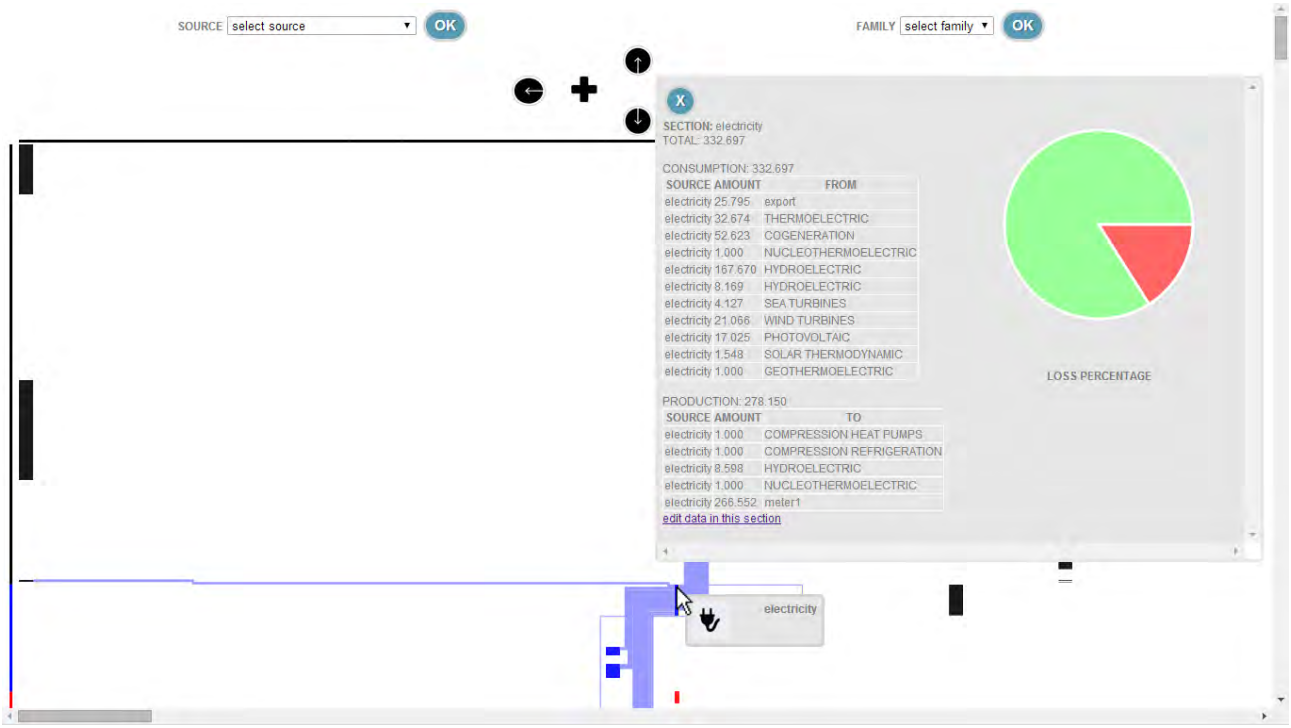



**Figura 12. Esempio di click su nodo nel diagramma LEB**

## 2.4 *Strumenti utilizzati*

Nelle presente sezione del rapporto si riportano alcune indicazioni di sintesi in merito agli ambienti e ai software utilizzati per lo sviluppo del database relazionale e dell'interfaccia grafica del modello LEB.

### 2.4.1 Modello dati

#### *MySql*

Per implementare la base dati in un sistema informativo si è scelto di utilizzare un software molto diffuso e ampiamente supportato che una delle più importanti aziende di software per database.
MySQL o Oracle MySQL è un Relational database management system (RDBMS[2]) composto da un client a riga di comando e un server. Entrambi i software sono disponibili sia per sistemi Unix e Unix-like che per Windows; le piattaforme principali di riferimento sono Linux e Oracle Solaris. MySQL è un software libero rilasciato a doppia licenza, compresa la GNU General Public License ed è sviluppato per essere il più possibile conforme agli standard ANSI SQL e ODBC SQL.
I sistemi ed i linguaggi di programmazione che supportano MySQL sono molto numerosi: ODBC, Java, Mono, .NET, PHP, Python e molti altri.

[2] Il termine Relational database management system (RDBMS), si indica un sistema di gestione dati basato sul modello relazionale.

Le piattaforme LAMP e WAMP incorporano MySQL per l'implementazione di server per gestire siti web dinamici, inoltre molti dei Content Management System come WordPress, Joomla e Drupal nascono proprio con il supporto predefinito a MySQL.
MySQL Standard è una versione del software disponibile con licenza GPL ed supportata da una vasta e attiva comunità di sviluppatori open source. Con questa versione è possibile fruire delle seguenti funzionalità:

- Architettura Pluggable Storage Engine
- Diversi motori di storage:
  - InnoDB
  - MyISAM
  - NDB (MySQL Cluster)
  - Memory
  - Merge
  - Archive
  - CSV
  - e molti altri ancora
- MySQL Replication per migliorare le prestazioni e la scalabilità delle applicazioni
- MySQL Partitioning per migliorare le prestazioni e la gestione di applicazioni di database di grandi dimensioni
- Stored procedure per migliorare la produttività degli sviluppatori
- Trigger per l'applicazione di regole di business complesse a livello di database
- Viste per garantire la massima protezione delle informazioni sensibili
- Schemi di prestazioni per il monitoraggio del consumo di risorse a livello di utente/applicazione
- Schemi di informazioni per un accesso semplice ai metadati
- MySQL Connectors (ODBC, JDBC, .NET, ecc.) per la creazione di applicazioni in più lingue
- MySQL Workbench per la modellazione visiva dei dati e lo sviluppo/amministrazione di istruzioni SQL

### *Cloud e Virtualizzazione*

Esistono diverse modalità di configurazione di un database su di un sistema. MySQL è diventato un potente e diffuso database relazionale Open Source soprattutto per La semplicità, la velocità e la flessibilità con cui realizza l'accesso e la gestione del dato, in particolare in ambienti web. Ma le sue funzionalità ne consentono un uso anche in modo massivamente parallelo e in ambiente Cloud Computing.
MySQL è facilmente utilizzabile in ambienti virtualizzati. Generalmente non è necessaria alcuna variazione nella configurazione per adattare il server ad un ambiente virtualizzato ma, nel caso si verifichi l'insorgere di limiti prestazionali legati all'accesso verso lo storage, è possibile introdurre un set di regole.
MySQL fornisce direttamente il supporto per alcune soluzioni di Cloud pubblico, in particolare su Amazon EC2. Una prima opzione sarebbe utilizzare direttamente le AMIs (Amazon Machine Images [http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html]) e predisporre l'ambiente in piena autonomia, in alternativa è possibile utilizzare anche il servizio Amazon RDS (Relational Database Service [http://aws.amazon.com/rds/details/]).
Oltre alla configurazione di un server MySQL su servizio di cloud pubblico è possibile predisporre una ambiente di cloud privato. In questo modo si ha un accesso più diretto alla base dati ed è inoltre possibile garantire un livello di privacy adeguato alle proprie esigenze. Ci sono diversi vantaggi derivanti dalla scelta di una implementazione di MySQL in cloud, tra cui le seguenti:

- consistenza dei dati - per una soluzione database la consistenza dei dati è fondamentale; le basi dati devono essere disegnate in modo adeguato, in particolare se utilizzate in Cloud; se si adotta la replicazione è opportuno disabilitare gli accessi in scrittura sugli Slave; la tecnica dello sharding deve essere utilizzata in modo significativo nelle applicazioni

- disponibilità dei dati - la disponibilità dei dati è un elemento critico di un'applicazione; se l'istanza che gestisce la base dati Master va in fault c'è un forte rischio di blocco dell'applicazione; quindi deve essere posta molta cura alle procedure di promote
- fine tuning - l'accesso allo storage presenta tipicamente tempi di latenza molto maggiori nelle architetture in cloud; risulta quindi molto importante il tuning della base dati per evitare accessi che possono essere risolti in cache
- backup - gli ambienti di virtualizzazione offrono strumenti specifici per la creazione di immagini (snapshot) dei sistemi; si tratta di tool molto efficienti ed è opportuno sfruttarli appieno anche per le attività amministrative

### *PhpMyAdmin*

Per poter amministrare un database MySQL tramite un qualsiasi browser, abbiamo utilizzato lo strumento phpMyAdmin (Figura 13). Si tratta di un'applicazione web scritta in PHP e rilasciata con licenza GPL [https://www.gnu.org/licenses/licenses.it.html], ed è indirizzata sia agli amministratori del database.
phpMyAdmin permette di creare un database da zero, creare le tabelle ed eseguire operazioni di ottimizzazione sulle stesse. Presenta un feedback sulla creazione delle tabelle per evitare eventuali errori. Sono previste delle funzionalità per l'inserimento dei dati (popolazione del database), per le query e per il backup dei dati.
L'amministratore ha anche a disposizione un'interfaccia grafica per la gestione degli utenti: l'interfaccia permette l'inserimento di un nuovo utente, la modifica della relativa password e la gestione dei permessi che l'utente ha sul database.

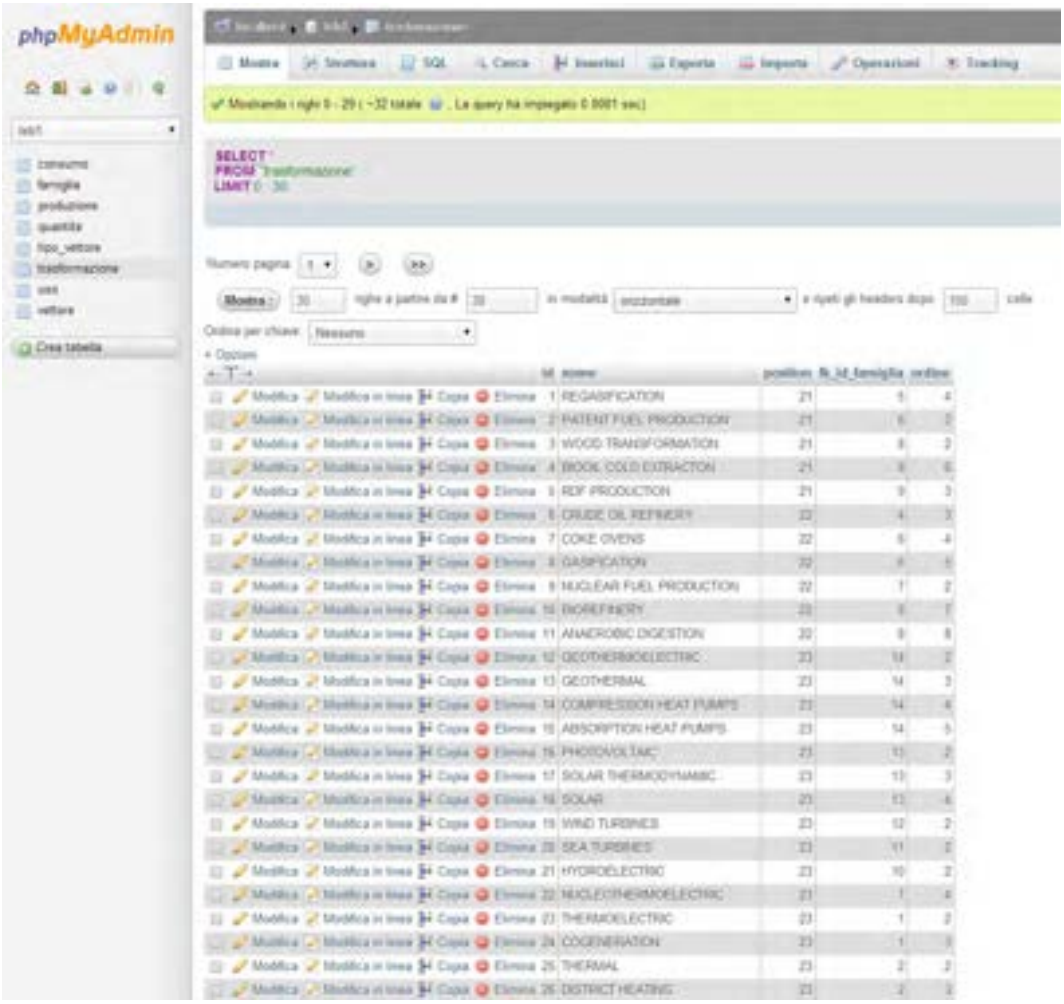



**Figura 13. Esempio di schermata software phpMyAdmin**

### *Flyspeed SQL query*

FlySpeed Query SQL (Figura 14) è uno strumento di SQL query, che consente la gestione delle interrogazioni al database in maniera visuale. E' stato principalmente progettato per soddisfare le esigenze degli utenti di database che hanno bisogno di lavorare con i dati attraverso query complesse, ed è meno indicato per la progettazione. Vi permette di creare query su molti server di database differenti, lavorando sia attraverso la sintassi SQL sia attraverso dei tool grafici.
Attraverso lo strumento FlySpeed Query SQL sono state generate la maggior parte delle query utilizzate dall'interfaccia grafica per esporre i dati ed i flussi.

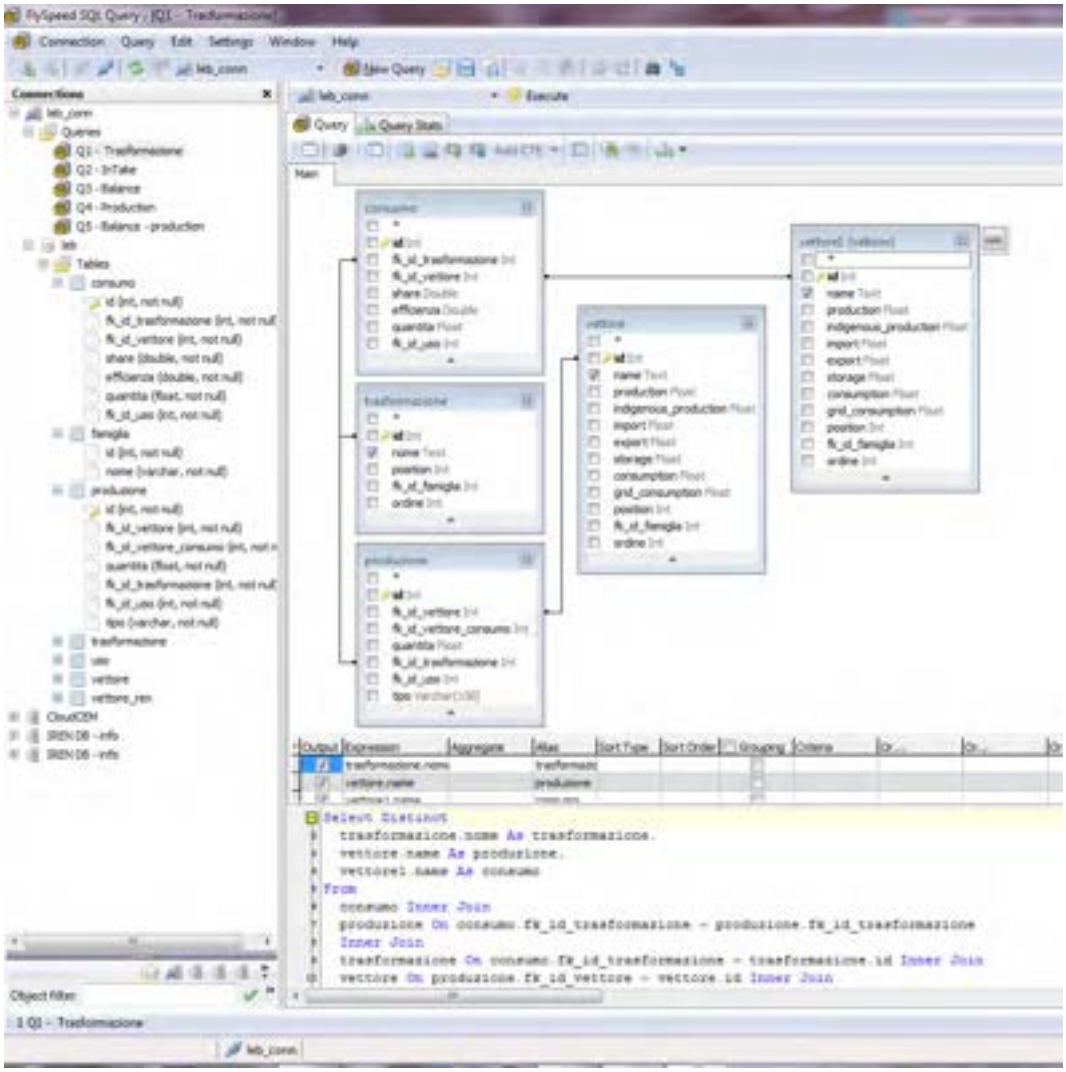


**Figura 14. Esempio di schermata software Flyspeed SQL query**

### 2.4.2 Interfaccia grafica

#### *HTML 5*

Per lo sviluppo dell'interfaccia è stato utilizzato il linguaggio di markup HTML 5 per realizzare la struttura della pagina web e CSS3 per definire lo stile grafico degli elementi presenti nella stessa.

#### *SVG*

Per realizzare la grafica dello schema è stato utilizzato il linguaggio SVG, utile per la definizione di grafica vettoriale sul web, permettendo scalamento e ridimensionamento senza alcuna perdita di qualità.
È stato utilizzato il tag <svg> all'interno del quale è contenuto tutto lo schema con i nodi e i flussi energetici. Il tag <svg> permette di definire al suo interno grafica vettoriale grazie ad una serie di tag specifici per disegnare forme geometriche o intere immagini scalabili. In particolare sono stati utilizzati tag <rect> per i nodi e tag <path> per i flussi, questi elementi una volta inseriti nella pagina sono presenti come oggetti nel DOM e possono essere modificati tramite codice JavaScript

## 2.5 Test di applicazione

Le attività di sviluppo sopradescritte hanno permesso di realizzare una prima versione dello strumento software LEB. Tale versione prototipale è stata sottoposto a test di applicazione su casi-studio costituiti da bilanci energetici di territori reali.
Alcuni test erano già stati condotti nel corso delle attività del PAR 2012, per verificare la disponibilità di dati preesistente e individuare le esigenze di approfondimento e integrazione, in relazione al maggior dettaglio della basi dati richiesta per l'applicazione della metodologia LEB. Questa seconda serie di test si è invece concentrata sulla verifica della corretta funzionalità del prototipo, in particolare per quanto concerne l'operatività dei differenti codici di calcolo, gestione e visualizzazione che costituiscono il motore del database e dell'interfaccia grafica.
In base alle indicazioni operative ricevute da ENEA, i casi –studio sono stati individuati nei territori della Regione Piemonte e della Provincia di Torino. Nel seguito è riportata una descrizione di dettaglio degli esiti dei test svolti.

### 2.5.1 Regione Piemonte

*Bilancio energetico regionale anno 2008 (storico)*

Il primo caso studio scelto è la Regione Piemonte. Si è fatto riferimento ai dati presenti nel Bilancio Energetico Regionale aggiornato al 2008, sviluppato da ENEA [13]. Alcuni dati sono stati integrati con stime effettuate durante la definizione della "Relazione programmatica sull'Energia" [14].
Attualmente la redazione di bilanci energetici regionali è effettuata prevalentemente in forma tabellare, seguendo la metodologia utilizzata per i bilanci nazionali, sia per quanto riguarda i bilanci di sintesi sia per i bilanci di dettaglio. All'interno del progetto Alcotra Renerfor è stata applicata la prima versione del modello LEB alle regioni che hanno partecipato al progetto. In questa fase il test di applicazione è stato esteso allo strumento sviluppato nella presente attività.
Il bilancio energetico della Regione Piemonte al 2008 evidenzia una sostanziale prevalenza di consumi di energia fossile, con una forte importazione di gas naturale per la produzione di energia elettrica e per i consumi degli usi finali, che si affianca ad un'importazione di petrolio greggio che viene poi raffinato nella regione per produrre vettori secondari destinati in parte al consumo interno e in parte all'esportazione verso regioni vicine. Si evidenzia inoltre un saldo negativo di energia elettrica verso l'esterno, con importazioni maggiori delle esportazioni. Riguardo alle fonti rinnovabili si osserva l'utilizzo della fonte idrica per la produzione di energia elettrica.
Le principali criticità riscontrate sono state causate dalla mancanza di alcuni dati necessari per la compilazione completa del LEB. In particolare, alcuni dati necessari sono dati di dettaglio non sempre disponibili dalle statistiche su scala regionale. I dati del bilancio energetico regionale hanno quindi dovuto essere integrati con stime, in particolare riguardo ai seguenti aspetti:

- produzione di energia in sistemi di teleriscaldamento e conseguenti consumi presso gli utenti finali;
- consumi di biomassa legnosa per riscaldamento presso gli utenti finali;
- conversioni energetiche presso gli utenti finali (principalmente riscaldamento), definizione di rendimenti medi stagionali del parco impianti e studio della penetrazione di alcune tecnologie (es. solare termico, pompe di calore, etc.).

In Figura 15 è riportata una schermata di esempio del test di applicazione, che evidenzia la prevalenza di consumi di energia fossile nel sistema energetico regionale. Si osserva inoltre come gli usi finali siano caratterizzati da una prevalenza di energia termica rispetto all'energia elettrica, a causa delle caratteristiche climatiche del territorio e delle tecnologie utilizzate per il riscaldamento degli edifici.

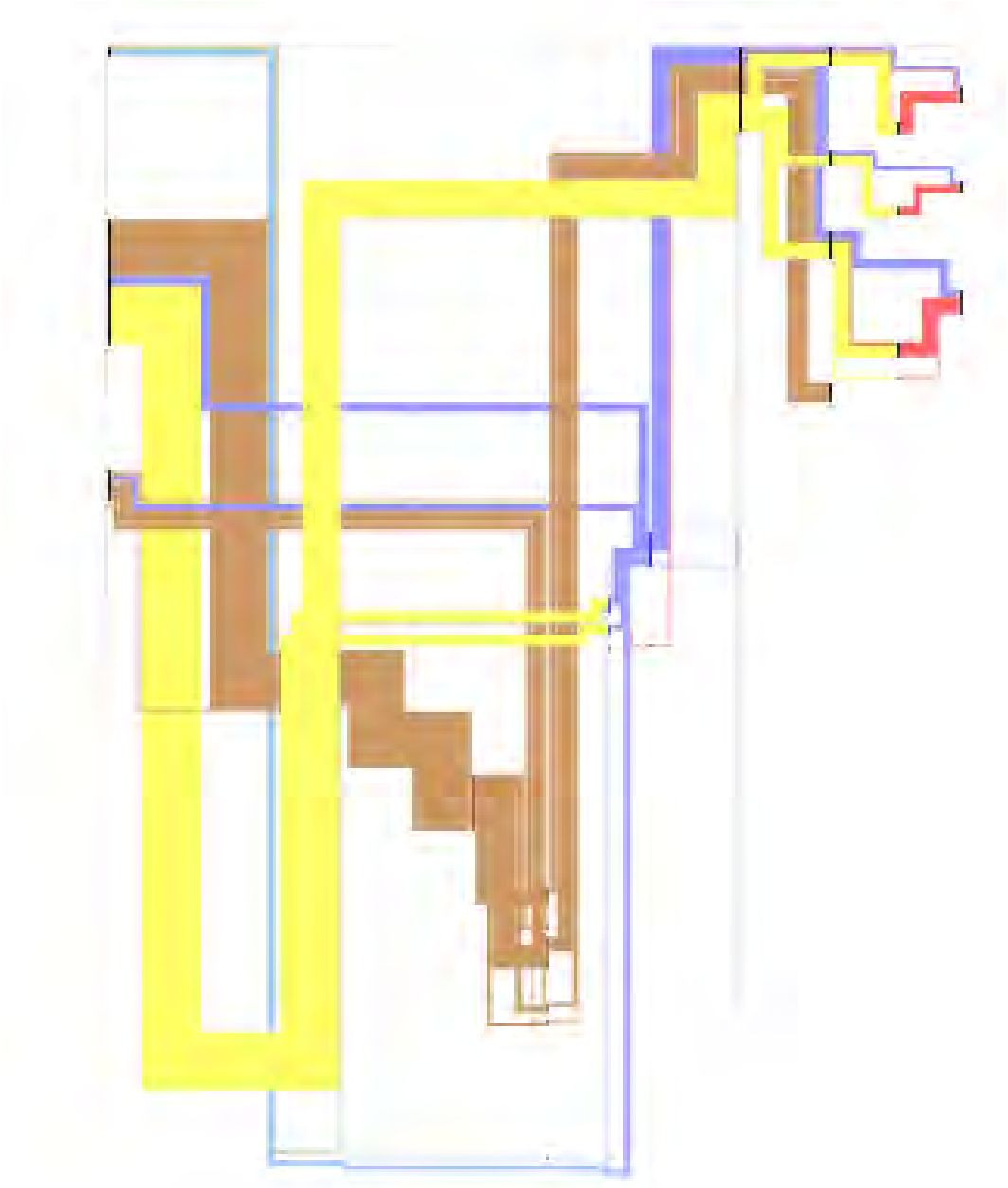


**Figura 15. Schermata di esempio – Regione Piemonte, storico**

### 2.5.2 Provincia di Torino

Il secondo caso studio considerato per l'applicazione del prototipo è stata la Provincia di Torino. Questo caso può beneficiare di una buona disponibilità di dati di dettaglio su un arco temporale considerevole. La Provincia di Torino ha una tradizione decennale di Bilanci Energetici Provinciali, con la pubblicazione biennale di un Rapporto sull'Energia in cui viene descritto in dettaglio il sistema energetico territoriale.

#### *Bilancio energetico provinciale anno 2011 (storico)*

I dati utilizzati in questo test di applicazione sono i più recenti disponibili, e sono stati pubblicati nell'Ottavo Rapporto sull'Energia, redatto nel 2012 [4]. Questa analisi è stata sviluppata dalla Provincia di Torino

avvalendosi della metodologia LEB, in collaborazione con il gruppo di ricerca di Sistemi per l'Energia e l'Ambiente del Politecnico di Torino.
Nel Bilancio Energetico Provinciale si osserva la presenza predominante del gas naturale, sia per la produzione di energia elettrica e calore in impianti di teleriscaldamento, sia per le conversioni energetiche presso gli utenti finali. A questo si affianca un'importazione di prodotti petroliferi già raffinati per il consumo negli usi finali, prevalentemente nel settore dei trasporti. Non sono presenti raffinerie sul territorio.
Si osserva come la maggior parte dei consumi finali sia concentrata nel settore residenziale, con consumo di combustibili per il riscaldamento degli edifici. Il declino della produzione industriale negli ultimi anni ha contribuito significativamente, unitamente alle caratteristiche del territorio.
Riguardo alle fonti rinnovabili si osserva la presenza significativa di impianti idroelettrici, affiancata ad un consumo apprezzabile di biomassa legnosa, principalmente per riscaldamento del settore residenziale. Altre fonti rinnovabili presenti non emergono dalla rappresentazione grafica in quanto il loro contributo rimane ancora marginale rispetto alle altre fonti.
In questo caso le criticità legate alla carenza di dati sono state già affrontate e risolte in una fase precedente. Pertanto è stato possibile applicare al prototipo un set di dati di input congruente con quelli richiesti. Questo ha permesso una rappresentazione abbastanza fedele dei flussi energetici costituenti il sistema energetico provinciale.
In Figura 16 è riportata una schermata di esempio del test di applicazione, che evidenzia i contributi forniti dai diversi vettori energetici nel mix di consumo provinciale. La Figura 17 fornisce invece un dettaglio in merito ai consumi di energia negli usi finali.
Per motivi grafici le figure non sono in scala con quella riportata precedentemente per la Regione Piemonte. Una scala uniforme per diverse realtà territoriali infatti permetterebbe da un lato di confrontarle tra di loro, ma si perderebbe l'informazione visiva relativa ai diversi flussi energetici presenti sul territorio, a causa delle loro ridotte dimensioni.

### *Bilancio energetico provinciale anno 2020 (scenario previsionale)*

Un ulteriore test di applicazione è stato svolto sui dati di scenario al 2020 sviluppati per la Provincia di Torino all'interno dell'Ottavo Rapporto sull'Energia [4].
Le principali differenze che emergono rispetto al bilancio a consuntivo del 2011 sono una diminuzione dei fabbisogni per gli usi finali, causati da interventi di efficienza energetica promossi nei diversi settori degli usi finali. Questo fenomeno racchiude diverse tendenze: da un lato si osserva una diminuzione dei fabbisogni di calore ed energia elettrica, dall'altro un aumento dell'efficienza media di conversione presso gli utenti finali e la penetrazione maggiore di tecnologie come le pompe di calore e i pannelli solari termici, che contribuiscono ad aumentare la quota di energia rinnovabile negli usi finali.

**Figura 16. Schermata di esempio – Provincia di Torino, storico**

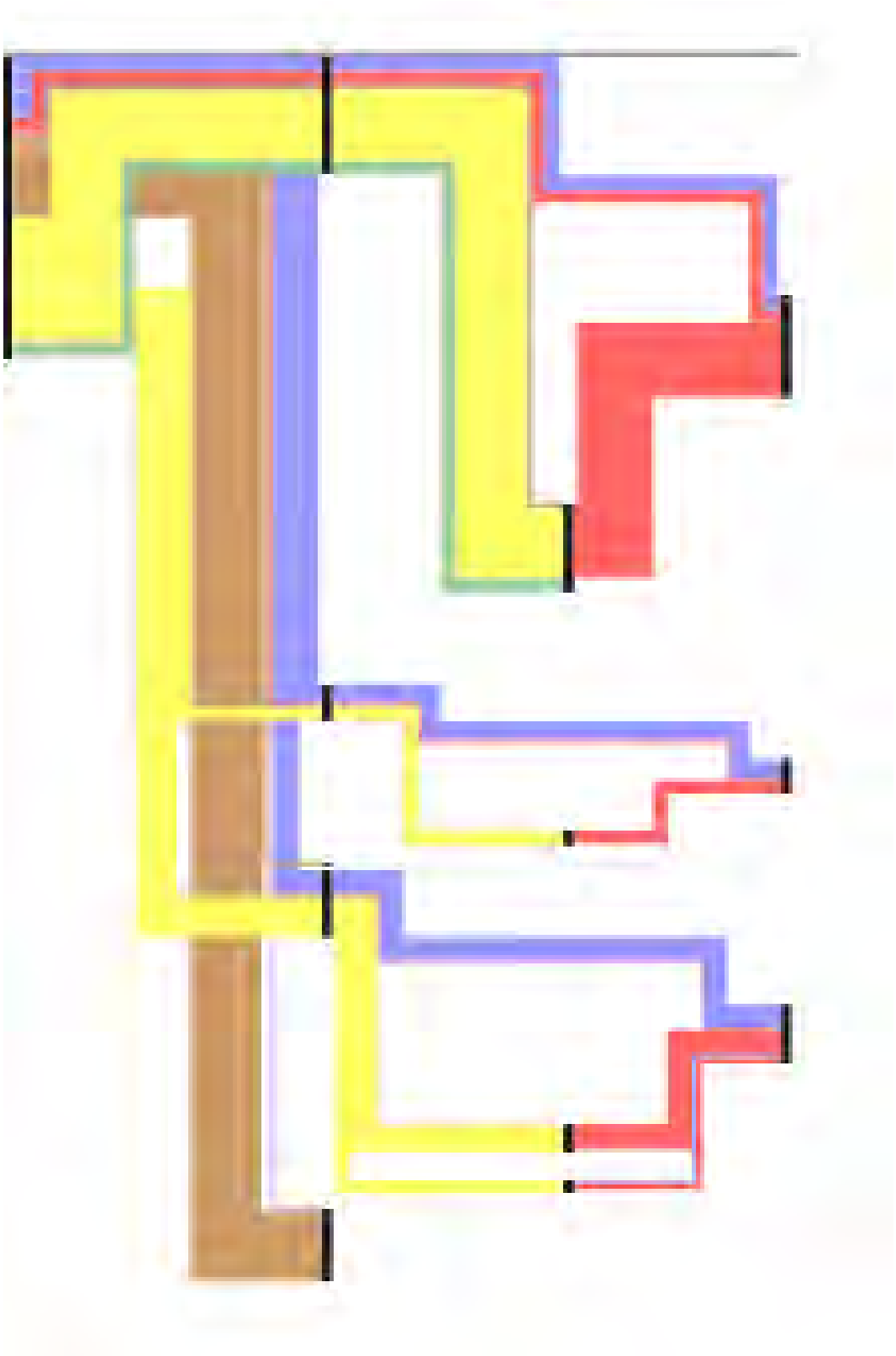


**Figura 17. Schermata di esempio – Provincia di Torino, storico, usi finali**

# 3 Conclusioni

### *Risultati del lavoro*

L'attività del PAR 2013 si colloca in continuità con le precedenti fasi di lavoro svolte sul modello di bilancio energetico territoriale LEB, nel progetto RENERFOR e nella Ricerca di Sistema Elettrico. Il risultato dell'attività è costituito dallo sviluppo di uno strumento software open source per l'implementazione della metodologia. Nel percorso di sviluppo della metodologia LEB, il prodotto realizzato (ancorché ancora a livello prototipale) rappresenta un significativo progresso rispetto al livello conseguito con le attività PAR 2012.
Il nuovo database consente una notevole estensione della capacità operativa dello strumento, grazie al superamento dei limiti di elaborazione dell'ambiente di sviluppo utilizzato per la versione preesistente (MS Excel). La scelta di un modello dati di tipo relazione consente inoltre una massimizzazione della flessibilità della sua struttura e una contestuale minimizzazione della presenza di dati ridondati. Questi benefici si traducono in un incremento della velocità di accesso al dato, sia in lettura che in scrittura.
La nuova interfaccia di visualizzazione utilizza una grafica vettoriale scalabile, sia in relazione alle dimensioni effettive del diagramma da rappresentare, sia in funzione della quantità di dati presenti nel database. Pur mantenendo integralmente il dettaglio dei dati di origine, il risultato informativo offerto dalla navigazione nel diagramma LEB consente una lettura con differenti livelli di approfondimento e selezione.
I principali punti di forza dello strumento sviluppato possono essere riepilogati come segue:

- l'architettura predisposta per l'elaborazione di grandi moli di dati, per rendere possibili molteplici analisi pluriennali di set di dati storici e previsionali, senza vincoli di numerosità e dettaglio
- la virtualizzazione del sistema con la gestione centralizzata delle basi informative, che consente l'accesso multiplo e condiviso ai dati garantendone contestualmente la loro consistenza e la tracciabilità dei loro aggiornamenti
- il disegno modulare aperto ad ulteriori adattamenti metodologici ed estensioni degli elementi rappresentanti nel modello di bilancio energetico, ad esempio per quanto riguarda la tipologia delle trasformazioni e il dettaglio negli usi finali
- la semplicità e libertà di impiego del sistema, offerta dallo sviluppo interamente open source dei codici di calcolo, gestione e visualizzazione

L'operatività dello strumento è stata confermata dagli esiti positivi dei test condotti su casi-studio reali. Il modello dati e la rappresentazione grafica si sono dimostrati adeguati all'analisi dei bilanci energetici dei territori studiati.

### *Sviluppi potenziali*

Grazie allo stadio di avanzamento conseguito nel suo strumento software di implementazione, la metodologia LEB rende ora possibili una serie di nuove funzionalità relative alla gestione dei dati energetici:

- il confronto diretto e la combinazione di bilanci energetici relativi a unità territoriali a scale differenti (stato, regioni province, comuni)
- la codifica, il confronto e la condivisone di dati energetici disponibili presso soggetti produttori (impianti, utilities) e utilizzatori (edifici, enti pubblici, imprese)
- l'integrazione multilivello tra i due elementi sopracitati e il trasferimento di informazioni tra l'ambito della pianificazione energetica e quello della gestione (energy manager)

La metodologia e lo strumento software LEB costituiscono un protocollo per la codifica, la conservazione, la stima, l'analisi e la condivisione di informazioni quantitative relative al settore energetico. La strutturazione di un unico sistema informativo di riferimento che consegua un'effettiva interoperabilità multilivello dei

dati energetici, dal singolo utente alle differenti unità territoriali di applicazione, rappresenta quindi una delle principali direttrici di sviluppo dell'attività di ricerca sin qui svolta.

# 4 Riferimenti bibliografici

[1] Regione Piemonte, “RENERFOR, Iniziative di cooperazione per lo sviluppo delle fonti di energia rinnovabili (bosco ed acqua) nelle Alpi Occidentali, il risparmio energetico e la riduzione delle emissioni di gas ad effetto serra”, Programma di cooperazione transfrontaliera tra Italia e Francia Alcotra 2007-2013, www.renerfor.eu

[2] A. Poggio, G. Cerino Abdin, A. Crocetta, L. Degiorgis, M. Noussan, R. Roberto, M. Caldera, M. Gualtieri, F. Hugony, “Metodologia per i bilanci energetici territoriali e la pianificazione delle politiche energetiche: uno strumento per la verifica del raggiungimento degli obiettivi strategici nazionali e comunitari”, Report Ricerca di Sistema Elettrico, Agosto 2013

[3] J. M. Barroso, “Climate and energy priorities for Europe: the way forward”, European Council, 20-21 marzo 2014, Brussels, Commissione Europea, http://goo.gl/rZ6C2A

[4] Provincia di Torino, “Ottavo rapporto sull’energia”, Novembre 2012, http://goo.gl/jCCcK3

[5] S. De Nigris, “L'ottavo rapporto sull'energia della Provincia di Torino”, Verso un nuovo sistema energetico: dal Piano alle azioni locali, 7 novembre 2012, Torino, Provincia di Torino, http://goo.gl/jCCcK3

[6] A. Poggio, “Scenari energetici al 2020 tra burden sharing system e trend in atto”, Verso un nuovo sistema energetico: dal Piano alle azioni locali, 7 novembre 2012, Torino, Provincia di Torino, http://goo.gl/jCCcK3

[7] Provincia di Torino, “Piano d’azione per l’energia sostenibile della Provincia di Torino”, Gennaio 2014, http://goo.gl/McDXGr

[8] R. Guglielminotti, “Consumi e fonti energetiche rinnovabili sul territorio regionale”, Energia, legno, acqua: pianificazione e gestione energetica sostenibile delle biomasse forestali e dell’idroelettrico nell’ambito del progetto strategico Renerfor, 9 maggio 2013, Aosta, Regione Valle d’Aosta, http://goo.gl/w5N4sq

[9] M. Noussan, A. Poggio, “Local Energy Balance: Da Renerfor un nuovo strumento per la pianificazione territoriale”, Energia, legno, acqua: pianificazione e gestione energetica sostenibile delle biomasse forestali e dell’idroelettrico nell’ambito del progetto strategico Renerfor, 9 maggio 2013, Aosta, Regione Valle d’Aosta, http://goo.gl/w5N4sq

[10] J.R. Carter, “Database Design and Programming with Access, SQL and Visual Basic”, 2002, McGraw-Hill

[11] International Energy Agency, “Statistics. Energy balance flows”, 2014, http://goo.gl/hEOJ88

[12] Canada Energy Systems Analysis Research, “Sankey diagrams of Canada's energy systems”, University of Calgary, 2014, http://goo.gl/XK8XoU

[13] Unità Tecnica Efficienza Energetica, “I B.E.R. 2005-2008 e le statistiche energetiche 1988-2008”, ENEA, 2013, http://goo.gl/M58BkJ

[14] Regione Piemonte, “Relazione Programmatica sull'Energia”, D.G.R. n. 30-12221, 28 settembre 2009, http://goo.gl/eKUuJq

# Appendice – Piattaforma software di gestione del database

## *Queries utilizzate per il popolamento del database - MySQL (listato)*

```
' inserisci valori di efficienza nella tabella produzione (query upd_eff_prod)
“Update produzione, (Select produzione.quantita / consumo.quantita As efficiency, produzione.id as id From consumo Inner Join produzione On produzione.fk_id_trasformazione = consumo.fk_id_trasformazione And produzione.fk_id_uso = consumo.fk_id_uso And produzione.fk_id_vettore_consumo = consumo.fk_id_vettore) as t “ & _
“set produzione.efficienza = t.efficiency where produzione.id = t.id”

' aggiorna valori di rinnovabili nella tabella consumo (query upd_ren_cons)
“Update consumo, (Select consumo1.fk_id_vettore As vett, 0 As quantita From consumo consumo1 Where (consumo1.fk_id_vettore < 7) Or (consumo1.fk_id_vettore > 13 And consumo1.fk_id_vettore < 28) Union Select consumo2.fk_id_vettore, consumo2.quantita From consumo consumo2 Where (consumo2.fk_id_vettore > 27) Or (consumo2.fk_id_vettore > 7 And consumo2.fk_id_vettore < 14)) as t " & _
“set consumo.ren = t.quantita where t.vett = consumo.fk_id_vettore”

' inserisci valori (query upd_tot_quant_energ)
Update quantita, (Select Sum(consumo.quantita) As q, consumo.fk_id_vettore as vett " & _
    "From consumo where consumo.fk_id_trasformazione = 30 Group By consumo.fk_id_vettore) as prod, " & _
    "(Select Sum(consumo.quantita) as q, consumo.fk_id_vettore as vett From produzione Inner Join consumo On consumo.fk_id_vettore = produzione.fk_id_vettore_consumo " & _
    "Group By consumo.fk_id_vettore) as cons " & _
    "set quantita.production = prod.q, quantita.consumption = cons.q " & _
    "where quantita.fk_id_vett = prod.vett and quantita.fk_id_vett = cons.vett and quantita.fk_id_tipo_vett = '1' “

' inserisci valori delle rinnovabili (query upd_tot_quant_ren)
“Insert into quantita(fk_id_vett, fk_id_tipo_vett, indigenous_production, import, export, storage, grid_consumption, consumption, production) “ & _
“Select quantita.fk_id_vett, 2 As fk_id_tipo_vett, 0 As indigenous_production, 0 As import, 0 As export, 0 As storage, 0 As grid_consumption, 0 As consumption, 0 As production From quantita Where (quantita.fk_id_vett < 7 Or (quantita.fk_id_vett > 13 And quantita.fk_id_vett < 28)) And quantita.fk_id_tipo_vett = '1' “ & _
“Union “ & _
“Select quantita.fk_id_vett, 2 As fk_id_tipo_vett, quantita.indigenous_production, quantita.import, quantita.export, quantita.storage, quantita.grid_consumption, quantita.consumption, quantita.production From quantita Where (quantita.fk_id_vett > 27 Or (quantita.fk_id_vett > 7 And quantita.fk_id_vett < 14)) And quantita.fk_id_tipo_vett = '1' “

' inserisci valori delle rinnovabili (query upd_ren_prod)
“Update produzione, (Select consumo.ren * produzione.efficienza As ren, produzione.id as id From produzione Inner Join consumo On produzione.fk_id_vettore_consumo = consumo.fk_id_vettore And consumo.fk_id_trasformazione = produzione.fk_id_trasformazione And consumo.fk_id_uso = produzione.fk_id_uso) as t “ & _
“set produzione.ren = t.ren where produzione.id = t.id “
```

## *Sorgente HTML della pagina web di restituzione grafica (listato)*

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta charset="utf-8" />
<title></title>
<link type="text/css" href="jquery-ui.css" rel="stylesheet" />
<style>
body {
color:#808080;
font-family: Arial;
font-size: 12px;
}
#bottoni {
text-align:center;
position:relative;
top:0;
left:0;
```

```
height:40px;
width:50%;
background-color:white;
float:left;
}
#bottoniF {
text-align:center;
position:relative;
top:0;
left:0;
height:40px;
width:50%;
background-color:white;
float:right;
}
rect {
z-index:1;
}
.vettore {
z-index:-1;
stroke-opacity:0;
pointer-events:all;
fill:none;
pointer-events:visibleStroke;
}
.vettore:hover {
stroke-opacity:1;
}
#info {
position:fixed;
right:0;
top:70px;
width:505px;
height:500px;
background-color:#e6e6e6;
float:right;
margin:10px;
padding:5px;
border: 3px #e6e6e6 solid;
border-radius: 5px;
overflow:scroll;
}
#infoLink {
position:fixed;
right:0;
top:0;
width:200px;
height:70px;
background-color:#e6e6e6;
float:right;
margin:10px;
padding:5px;
border: 3px #d3d3d3 solid;
border-radius: 5px;
text-align: center;
}
#infoRect {
position:fixed;
right:0;
top:0;
width:100px;
```

```
height:40px;
background-color:#e6e6e6;
float:right;
margin:10px;
padding:5px;
border: 3px #d3d3d3 solid;
border-radius: 5px;
}
#infoFC {
position:fixed;
right:0;
top:0;
height:20px;
background-color:#e6e6e6;
float:right;
margin:10px;
padding:5px;
border: 3px #d3d3d3 solid;
border-radius: 5px;
text-align:center;
}
button {
color: #dddddd;
background-color: #808080;
}
.myButton {
background-color: #599bb3;
-moz-border-radius: 42px;
-webkit-border-radius: 42px;
border-radius: 42px;
border: 3px solid #d3d3d3;
display: inline-block;
cursor: pointer;
color: #ffffff;
font-family: Arial;
font-size: 15px;
font-weight: bold;
padding: 5px 9px;
text-decoration: none;
text-shadow: 2px 1px 0px #3d768a;
}
.myButton:hover {
background-color: #408c99;
}
.myButton:active {
position: relative;
top: 1px;
}
.b_vet, .b_fam {
cursor: pointer;
height: 20px;
-moz-border-radius: 42px;
-webkit-border-radius: 42px;
border-radius: 42px;
border: 1px solid #d3d3d3;
margin-top:2px;
}
.control {
-moz-border-radius: 42px;
-webkit-border-radius: 42px;
border-radius: 40px;
```

```
border: 1px solid #d3d3d3;
}
#controller {
width:100%;
height:100px;
text-align:center;
position:relative;
top:0px;
background-color:white;
clear:both;
}
#up {
width: 30px;
height: 30px;
background-color: #808080;
margin:0 auto;
background-image: url("icone/frecciaSU.jpg");
background-size: 100% 100%;
}
#left {
width: 2.5%;
height: 30px;
background-color: #808080;
float:left;
margin-left:40%;
margin-right: 2%;
background-image: url("icone/frecciaSX.jpg");
background-size: 100% 100%;
}
#sliderCont {
height:30px;
width:10%;
float:left;
}
#right {
width: 2.5%;
height: 30px;
float:left;
margin-left:2%;
margin-right:40%;
background-image: url("icone/frecciaDX.jpg");
background-size: 100% 100%;
}
#down {
width: 30px;
height: 30px;
background-color: #808080;
margin:0 auto;
clear:both;
background-image: url("icone/frecciaGIU.jpg");
background-size: 100% 100%;
}
#in {
width: 30px;
height: 30px;
background-color: #808080;
margin:0 auto;
float:left;
background-image: url("icone/piu.jpg");
background-size: 100% 100%;
}
```

```
#out {
width: 30px;
height: 30px;
background-color: #808080;
margin:0 auto;
float:right;
background-image: url("icone/meno.jpg");
background-size: 100% 100%;
}
p {
margin: 0 0 0 0;
}
#holder {
margin: 0 auto;
width: 200px;
height: 300px;
float:right;
text-align:center;
position:relative;
top:20px;
}
.tabella {
border-collapse: collapse;
}
.tabella, th, td {
border:1px solid white;
}
</style>
<script src="jquery.js" type="text/javascript"></script>
<script src="jquery-ui.js" type="text/javascript"></script>
</head>
<script>
$(document).ready(function () {
$("#info").hide();
$("#infoLink").hide();
$("#infoRect").hide();
$("#infoFC").hide();
var fattore = 1; //fattore divisione spessore link (maggiore il valore minore lo spessore)
var vShift = 0; //fattore si scostamento verticale tra una classe/subclasse e la successiva
var screenW = window.innerWidth * 1 ;
var colori = ["#1f77b4", "#ff7f0e", "#2ca02c", "#d62728", "#9467bd", "#8c564b", "#e377c2", "#7f7f7f", "#bcbd22", "#17becf", "#aec7e8", "#ffbb78", "#98df8a", "#3333CC", "#ff9896", "#FF3300", "#c5b0d5", "#FF6666", "#996600", "#FF9900", "#CCCC33", "#CCCCFF", "#CC0033", "#FF99FF", "#33FF00", "#FFCC99", "#669966", "#996699", "#993333", "#FFCC33", "#990099", "#000033", "#66FF99", "#336633", "#FFCCCC", "#CC99FF", "#FF9999", "#3366CC", "#CC6633", "#660033", "#99CCFF", "#FFFFCC", "#CC9999", "#CCFF99", "#CC6666", "#66CC33", "#FF6633", "#CC3333", "#6600CC", "#00CCFF", "#CCFFCC"];
var colFam = ["#000000", "#0000FF", "#FF0000", "#B7DEE8", "#993300", "#FFFF00", "#808080", "#800080", "#008000", "#003300", "#00CCFF", "#000080", "#CC99FF", "#FFCC00", "#666699", "#000000"];
var coloriLink = ["#000000", "#9999FF", "#FF6666", "#CCCCFF", "#CC9966", "#FFFF66", "#C0C0C0", "##CC99FF", "#99CC99", "#669966", "#99CCFF", "#6666CC", "#CC99FF", "#FFCC99", "#9999CC", "#000000"];
var colonne = ["gross consumption", "primary sources", "mechanical transformations", "secondary sources", "chemical transformations", "secondary_sources", "energy conversions", "tertiary sources", "final uses meter", "meter by uses", "final uses", "ultimate uses"]; //, "final grid supplies meter"];
var nomif = ["Local balance", "Electricity", "Heat", "Cool", "Oil", "Natural gas", "Coal", "Nuclear", "Biomass", "Waste", "Hydro", "Sea", "Wind", "Sun", "Geo/Aero/Hydro-thermal","Energy"];
var system = [["indigenous production", "gross consumption"], ["import", "gross consumption"], ["export", "gross consumption"], ["storage", "gross consumption"]];
var meter = [["final user", "final uses meter"]];
var xcol;
var nodi = [];
var nodiFinal = [];
nodi = nodi.concat(system);
var bilancio = [];
```

```
var bilancio2 = [];
var bilancio3 = [];
var bilancio4 = [];
var uses = [];
var nodiFinal = [];
var lastMeter = [];
var links = [];
var tempLinks = [];
var linksBKup = [];
var selezionato=0;
//var links = [["natural gas", "gasification", "15", "natural gas"], ["gasification", "gasoline", "8", "natural gas"], ["solar", "Hydropower", "6", "solar"], ["natural gas", "Solar Thermal", "13", "natural gas"], ["natural gas", "Gas/diesel oil", "8", "natural gas"], ["Thermal", "patent fuel", "6", "natural gas"], ["Hydropower", "consumption", "5", "solar"], ["compression refrigeration", "cipped wood", "7", "electricity"], ["heat", "Solar Thermal", "9", "heat"], ["cool", "biogas", "15", "coal"], ["cool", "Hydropower", "5", "coal"]]; //source, target, value, vettore
//var links = [["motor gasoline", "Oil Refinery", "15", "motor gasoline"], ["Oil Refinery", "gas/diesel oil", "7", "motor gasoline"], ["Oil Refinery", "motor gasoline_", "8", "motor gasoline"], ["motor gasoline_", "biodiesel", "8", "motor gasoline"], ["biodiesel", "thermoelectric", "8", "motor gasoline"], ["thermoelectric", "heat_", "8", "motor gasoline"], ["heat_", "meter1", "8", "motor gasoline"], ["heat_", "meter1", "5", "electricity"], ["heat_", "meter1", "5", "heat"], ["meter1", "residential", "18", "motor gasoline"], ["electricity", "Power Plants", "10", "electricity"], ["Power Plants", "aviation fuel", "10", "electricity"], ["aviation fuel", "thermal", "10", "electricity"], ["thermal", "cool_", "5", "electricity"], ["thermal", "heat_", "5", "electricity"], ["crude oil", "Oil Refinery", "25", "crude oil"], ["Oil Refinery", "kerosene", "5", "crude oil"], ["Oil Refinery", "Fuel oil", "10", "crude oil"], ["Oil Refinery", "liquified petroleum gas", "15", "crude oil"], ["liquified petroleum gas", "meter2", "15", "crude oil"], ["Fuel oil", "meter2", "10", "crude oil"], ["kerosene", "meter2", "5", "crude oil"], ["meter2", "industry", "25", "crude oil"], ["industry", "consumption", "15", "crude oil"], ["residential", "consumption", "18", "motor gasoline"], ["commercial", "consumption", "10", "cool"], ["industry", "final energy production", "10", "crude oil"], ["heat", "Thermal", "5", "heat"], ["cool", "Thermal", "5", "cool"], ["Thermal", "thermal", "5", "heat"], ["Thermal", "thermal", "5", "cool"], ["thermal", "heat_", "5", "heat"], ["thermal", "cool_", "5", "cool"], ["cool_", "commercial", "10", "cool"], ["hydro", "Hydropower", "15", "hydro"]]; //source, target, value, vettore
var lineeIn = [];
var lineeOut = [];
var vettori = [];
var trasformazioni = [];
var trasformazioni2 = [];
var trasformazioni3 = [];
var svg = document.getElementsByTagName("g")[0];
/*svg.setAttribute("width", screenW);
svg.setAttribute("height", window.innerHeight);
var svg1 = document.getElementsByTagName("svg")[0];
svg1.setAttribute("width", window.innerWidth);
svg1.setAttribute("height", window.innerHeight-190);*/
svg.addEventListener("mousewheel", MouseScroll, false);
svg.setAttribute("width", screenW);
svg.setAttribute("height", window.innerHeight);
svg.addEventListener("click", dovesono);
//svg.setAttribute("width", 12500);
//svg.setAttribute("height", 11600);
var svg1 = document.getElementsByTagName("svg")[0];
svg1.setAttribute("width", window.innerWidth);
svg1.setAttribute("height", window.innerHeight - 190);
//svg1.setAttribute("width", 12500);
//svg1.setAttribute("height", 11600);
var colonna; //nome colonna cliccata
var elenco; //array temporaneo corrispondente alla colonna selezionata
var width = 150 / fattore; //larghezza nodo
var widthB = 50 / fattore; //larghezza nodo bilancio
var sel_vect = "";
var vetto = "";
var altezze = [];
var posizioni = [];
var posizioniFinal = [];
var famiglie = [];
var nodiOrdinati = [];
var linkOrdinati = [];
var altezze2 = []; //altezze per famiglia
var altezzeSub = []; //altezze x subfamiglia
```

```
var spessori = []; //spessori tra colonne
var tempspesOut = [];
var tempspesIn = [];
var lastF; //ultima famiglia
var ics;
var matrix = [1, 0, 0, 1, 0, 0];
svg.setAttribute("transform", "matrix(" + matrix.join(' ') + ")");
xmlhttp = new XMLHttpRequest();
getVettori();
//getBilancio();
function MouseScroll(e) {
console.log("scrollo", e.wheelDelta);
if (e.wheelDelta == 120) { var scale = 1.2; }
else { var scale = 0.9; }
var svg = document.getElementsByTagName("g")[0];
for (var e = 0; e < matrix.length; e++) {
matrix[e] *= scale;
}
matrix[4] += (1 - scale) * window.innerWidth / 2;
matrix[5] += (1 - scale) * (window.innerHeight - 190) / 2;
console.log(matrix);
var matrix2 = "matrix(" + matrix.join(' ') + ")";
svg.setAttributeNS(null, "transform", matrix2);
}
function dovesono() {
var largo = svg.getAttribute("width") / 2;
var alto = svg.getAttribute("height") / 2;
matrix[4] += largo - event.clientX ;
matrix[5] += alto - event.clientY;
var matrix2 = "matrix(" + matrix.join(' ') + ")";
svg.setAttributeNS(null, "transform", matrix2);
console.log(event.clientX, event.clientY, largo, alto);
}
function init() {
removeZero();
console.log("LINKS", linkOrdinati);
blocchi();
$("#close").click(close);
ordina();
riempiSpessori();
getAltezze();
creaNodi();
//creaNodiFinal();
//stampa();
//console.log("ALTEZZE ",altezze);
linksBKup = linkOrdinati;
creaGriglia();
ics = 100;
for (var v = 0; v < 8; v++) {
ics += (spessori[v][1]);
}
rigaSeparazione();
creaLinks();
creabottoni();
var svg = document.getElementsByTagName("g")[0];
var zoomFact = 1366/0.1;
for (var e = 0; e < matrix.length; e++) {
matrix[e] *= window.innerWidth/zoomFact;
```

```
}
matrix[4] += 0;
matrix[5] += 0;
var matrix2 = "matrix(" + matrix.join(' ') + ")";
svg.setAttributeNS(null, "transform", matrix2);
}
function removeZero() {
for (var i = links.length - 1; i > -1; i--) {
if (links[i][2] < 1) {
links.splice(i, 1);
}
}
}
function stampa() {
for (var i = 0; i < linkOrdinati.length; i++) {
console.log('{"source":"' + linkOrdinati[i][0] + '","target":"' + linkOrdinati[i][1] + '","value":"' + linkOrdinati[i][2] + '","vettore":"' + linkOrdinati[i][3] +
'"},');
}
}
function creabottoni() {
$("#numV").append('<option value="" >select source</option>');
$("#numV").append('<option value="" >all sources</option>');
$("#numF").append('<option class="b_fam" value="" >select family</option>');
$("#numF").append('<option class="b_fam" value="" >all families</option>');
for (var i = 0; i < vettori.length; i++) {
// var b = document.createElement("input").button();
//$("#bottoni").append('<input class="b_vet" type="button" value="' + vettori[i][0] + '" style="background-color:' + colori[i] + '; color:white">');
$("#numV").append('<option class="b_vet" value="' + vettori[i][0] + '" >' + vettori[i][0] + '</option>');
//var b = document.getElementsByClassName("b_vet")[i];
//b.addEventListener("click", seleziona, false);
}
for (var j = 1; j < lastF; j++) {
// var b = document.createElement("input").button();
//$("#bottoniF").append('<input class="b_fam" type="button" value="' + j + '" style="background-color:#c0c0c0; color:white">');
$("#numF").append('<option class="b_fam" value="' + j + '" >' + nomif[j] + '</option>');
//var bf = document.getElementsByClassName("b_fam")[j];
//bf.addEventListener("click", selFamiglia, false);
}
var go = document.getElementById("vai");
go.addEventListener("click", seleziona, false);
var go2 = document.getElementById("invio");
go2.addEventListener("click", selFamiglia, false);
}
function rigaSeparazione() {
var path = "M" + ics + "," + 0 + " L" + ics + "," + 10550;
var link = document.createElementNS('http://www.w3.org/2000/svg', 'path');
//link.setAttribute("id", "griglia" + (i * j));
link.setAttribute("d", path);
link.setAttribute("class", "griglia");
link.setAttribute("stroke-linecap", "butt");
link.setAttribute("style", "stroke:black; fill:none; stroke-opacity:1; stroke-width:10; stroke-dasharray:10,10;");
$("g").prepend(link);
}
function creaGriglia() {
var y = 50;
var yPos = [];
var xPos = 100;
var lunghezza = 100;
var larghezza = 0;
for (var w = 0; w < 8; w++) {
```

```
if (w == 1 || w == 3 || w == 5 || w == 7) {
lunghezza += spessori[w][1];
} else {
lunghezza += spessori[w][1];
}
}
for (var i = 0; i < lastF; i++) {
for (var j = 1; j < altezzeSub[i].length; j++) {
y += (altezzeSub[i][j]*2);
yPos[i] = y;
}
//SELEZIONE FAMIGLIE
var t = document.createElementNS('http://www.w3.org/2000/svg', 'text');
t.setAttribute("class", "txtF");
t.setAttribute("text-anchor", "start");
t.setAttribute("style", "font-family: Arial; font-size: 100px;");
t.setAttribute("x", 20);
t.setAttribute("baseline-shift", "sub");
var r = document.createElementNS('http://www.w3.org/2000/svg', 'rect');
r.setAttribute("class", "rectF");
r.setAttribute("title", nomif[i]);
r.setAttribute("x", 0);
r.setAttribute("width", 30);
var rf = document.createElementNS('http://www.w3.org/2000/svg', 'rect');
rf.setAttribute("class", "rectFtot");
rf.setAttribute("id", "rectFtot_"+i);
rf.setAttribute("x", 30);
rf.setAttribute("width", lunghezza);
if (i == 0) {
r.setAttribute("id", "f0");
t.setAttribute("y", 50);
r.setAttribute("height", yPos[i] - 50);
r.setAttribute("y", 50);
rf.setAttribute("height", yPos[i] - 50);
rf.setAttribute("y", 50);
}
else {
r.setAttribute("id", "f" + i);
t.setAttribute("y", yPos[i - 1]);
r.setAttribute("height", yPos[i] - yPos[i - 1]);
r.setAttribute("y", yPos[i - 1]);
rf.setAttribute("height", yPos[i] - yPos[i - 1]);
rf.setAttribute("y", yPos[i - 1]);
}
r.setAttribute("fill", colFam[i]);
rf.setAttribute("fill", colFam[i]);
rf.setAttribute("style", "opacity:0");
var txtNode = document.createTextNode(nomif[i]);
t.appendChild(txtNode);
//$("g").prepend(t);
$("g").prepend(r);
$("g").prepend(rf);
//RIGHE ORIZZONTALI
/*var path = "M" + 0 + "," + y + " L" + (screenW) + "," + y;
var link = document.createElementNS('http://www.w3.org/2000/svg', 'path');
link.setAttribute("id", "griglia" + (i * j));
link.setAttribute("d", path);
link.setAttribute("class", "griglia");
link.setAttribute("stroke-linecap", "butt");
link.setAttribute("style", "stroke:black; fill:none; stroke-opacity:1; stroke-width:5;");
```

```
$("g").prepend(link);
*/
}
//RETTANGOLI SELEZIONE COLONNE
for (var v = 0; v < spessori.length; v++) {
larghezza =spessori[v][1];
var rc = document.createElementNS('http://www.w3.org/2000/svg', 'rect');
rc.setAttribute("class", "rectC");
rc.setAttribute("title", colonne[v]);
rc.setAttribute("id", "c" + v);
rc.setAttribute("x", xPos);
rc.setAttribute("y", 0);
rc.setAttribute("width", larghezza);
rc.setAttribute("height", 30);
//rc.setAttribute("fill", colori[v + 1]);
rc.setAttribute("fill", "#000000");
$("g").prepend(rc);
var rct = document.createElementNS('http://www.w3.org/2000/svg', 'rect');
rct.setAttribute("class", "rectCtot");
rct.setAttribute("id", "rectCtot_" + v);
rct.setAttribute("x", xPos);
rct.setAttribute("y", 50);
rct.setAttribute("width", larghezza);
rct.setAttribute("height", yPos[spessori.length - 1] + yPos[(spessori.length - 1)] - yPos[(spessori.length - 2)]);
//rct.setAttribute("fill", colori[v + 1]);
rct.setAttribute("fill", "#000000");
rct.setAttribute("style", "opacity:0");
$("g").prepend(rct);
xPos += spessori[v][1];
}
$(".rectF")
.mouseenter(function () {
var n = this.getAttribute("id").split("f")[1];
$("#rectFtot_" + n).css('opacity', 0.1);
var testo4 = "<b>" + this.getAttribute("title") + "</b>";
console.log(testo4);
$("#infoFC").css({ "width": "" + (testo4.length * 7) + "px" });
//$("#info").animate({ top: "0" }, 1000, "linear");
$("#infoFC").css("left", event.clientX);
$("#infoFC").css("top", event.clientY);
$("#infoFC").show();
$("#infoFC p").html(testo4.toUpperCase());
//console.log("icone/" + nome.replace(" ", "_") + ".png");
})
.mouseleave(function () {
$(".rectFtot").css('opacity', 0);
$("#infoFC").hide();
$("#infoFC p").html("");
});
$(".rectC")
.mouseenter(function () {
var n2 = this.getAttribute("id").split("c")[1];
$("#rectCtot_" + n2).css('opacity', 0.1);
var testo5 = "<b>" + this.getAttribute("title") + "</b>";
console.log(testo5);
$("#infoFC").css({ "width": "" + (testo5.length * 7) + "px" });
//$("#info").animate({ top: "0" }, 1000, "linear");
$("#infoFC").css("left", event.clientX);
```

```
$("#infoFC").css("top", event.clientY);
$("#infoFC").show();
$("#infoFC p").html(testo5.toUpperCase());
//console.log("icone/" + nome.replace(" ", "_") + ".png");
})
.mouseleave(function () {
$(".rectCtot").css('opacity', 0);
$("#infoFC").hide();
$("#infoFC p").html("");
});

//RIGHE VERTICALI
/* var ics = 100;
for (var v = 0; v < 8; v++) {
ics += (spessori[v][1]);
}
var path = "M" + ics + "," + 0 + " L" + ics + "," + 10550;
var link = document.createElementNS('http://www.w3.org/2000/svg', 'path');
//link.setAttribute("id", "griglia" + (i * j));
link.setAttribute("d", path);
link.setAttribute("class", "griglia");
link.setAttribute("stroke-linecap", "butt");
link.setAttribute("style", "stroke:black; fill:none; stroke-opacity:1; stroke-width:10; stroke-dasharray:10,10;");
$("g").prepend(link);*/
/*for (var d = 0; d < spessori.length; d++) {
if (d == 0) {
cerchi[d].setAttribute("cx", 20);
cerchi[d + 1].setAttribute("cx", 20);
d++;
}
else {
cerchi[d].setAttribute("cx", spessori[d - 1][1] + 20);
cerchi[d + 1].setAttribute("cx", spessori[d - 1][1] + 20);
d++;
}
}
}*/
}
function selpath(e) {
var vetto = e.target.getAttribute("value");
selezionato = 1;
var linksCopy = [];
var timer = 0;
for (var i = 0; i < linkOrdinati.length; i++) {
timer = i;
console.log("LINK " + linkOrdinati[i] + " VETTO " + vetto);
if (linkOrdinati[i][0] == vetto) {
tempLinks.push(linkOrdinati[i]);
if (linksCopy.indexOf(linkOrdinati[i][1]) == -1) { console.log("AGGIUNGO " + linkOrdinati[i][1]); linksCopy.push(linkOrdinati[i][1]); }
}
if (i == linkOrdinati.length - 1 && linksCopy.length == 0) { console.log("ESCI"); break; }
if (i == linkOrdinati.length - 1) {
vetto = linksCopy[0];
linksCopy.splice(0, 1);
console.log("VETTO " + vetto);
i = 0;
}
}
console.log(tempLinks);
eliminaLink();
```

```
lineeIn = [];
lineeOut = [];
trovato = 0;
shiftOut = 0;
shiftIn = 0;
//creaLinks();
}
function seleziona(e) {
for (var j = 0; j < linkOrdinati.length; j++) {
$("#p" + j).mouseenter(pMe);
$("#p" + j).mouseleave(pMl);
}
//console.log("vett selezioanto ", sel_vect, e.target.getAttribute("value"), "vetto ",vetto);
var val = document.getElementById("numV");
var indici = [];
var tLinks = linkOrdinati.slice(0);
if (vetto != val.options[val.selectedIndex].value && val.options[val.selectedIndex].value!="") {
//$("input").css("border", "1px solid rgb(211, 211, 211)");
//tempLinks = [];
console.log("DESELEZIONO IL PRIMO");
vetto = val.options[val.selectedIndex].value;
sel_vect = val.options[val.selectedIndex].value;
//var v = e.target.getAttribute("value");
console.log("Cambio ", sel_vect, this.value, "link totali", tLinks.length, linkOrdinati.length);
if (selezionato == 0 || selezionato == 1) {
// $(this).css("border", "3px solid yellow");
//console.log(" qqq selezionato");
//var vetto = e.target.getAttribute("value");
selezionato = 1;
var linksCopy = [];
var timer = 0;
var primario = 0;
/*for (var w = 0; w < links.length; w++) {
//console.log("LINK " + links[i] + " VETTO " + sel_vect);
if (links[w][0] == sel_vect) {
primario = 1;
break;
}
else {
primario = 0;
}
}*/
var l = tLinks.length;
for (var i = l-1; i > -1; i--) {
//timer = i;
//console.log("LINK " + links[i] + " VETTO " + sel_vect);
/*if (primario == 1) {
if (tLinks[i][0] == sel_vect) {
tempLinks.push(tLinks[i]);
console.log("QUI!!",tLinks[i],i,linksCopy);
if (linksCopy.indexOf(tLinks[i][1]) == -1) {
console.log("AGGIUNGO " + links[i][1] + " "+ i);
linksCopy.push(tLinks[i][1]);
}
tLinks.splice(i, 1);
}
if (i == 0 && linksCopy.length == 0) {
console.log("ESCI");
break;
}
```

```
if (i == 0) {
sel_vect = linksCopy[0];
//console.log("VETTO ", sel_vect);
linksCopy.splice(0, 1);
i = tLinks.length -1;
}
//console.log("PRIMARIO 1", i, links.length - 1, linksCopy.length);
}*/
if (primario == 0) {
if (tLinks[i][3] == sel_vect) {
tempLinks.push(tLinks[i]);
indici.push(i);
}
if (tLinks[i][3] == sel_vect) {
//tempLinks.push(linkOrdinati[i]);
console.log(("#p" + i));
document.getElementById("p" + i).getAttribute("style");
var ts = document.getElementById("p" + i).getAttribute("style").split("stroke-opacity")[0];
//, "stroke:" + col + "; fill:none; stroke-opacity:1; stroke-width:" + linkOrdinati[i][2] + ";");
document.getElementById("p" + i).setAttribute("style", ts + "stroke-opacity: 1; stroke-width:" + linkOrdinati[i][2] + ";");
//$("#p" + i).unbind("mouseenter");
$("#p" + i).unbind("mouseleave");
//$("#p" + i).mouseenter(pMe);
$("#p" + i).mouseleave(function () {
for (var o = 0; o < indici.length; o++) {
var temps = document.getElementById("p" + indici[o]).getAttribute("style").split("stroke-opacity")[0];
//, "stroke:" + col + "; fill:none; stroke-opacity:1; stroke-width:" + linkOrdinati[i][2] + ";");
document.getElementById("p" + indici[o]).setAttribute("style", temps + "stroke-opacity: 1; stroke-width:" + linkOrdinati[indici[o]][2] + ";");
}
$("#infoLink").hide();
$("#infoLink p").html("");
});
}
else {
document.getElementById("p" + i).getAttribute("style");
var ts = document.getElementById("p" + i).getAttribute("style").split("stroke-opacity")[0];
document.getElementById("p" + i).setAttribute("style", ts + "stroke-opacity: 0.1; stroke-width:" + linkOrdinati[i][2] + ";");
$("#p" + i).unbind("mouseenter");
$("#p" + i).unbind("mouseleave");
}
/* if (tLinks[i][0] == sel_vect ) {
tempLinks.push(tLinks[i]);
//console.log("QUI!!", tLinks[i], i, linksCopy);
if (linksCopy.indexOf(tLinks[i][1]) == -1) {
//console.log("AGGIUNGO " + links[i][1]);
linksCopy.push(tLinks[i][1]);
}
tLinks.splice(i, 1);
}
if (i == 0 && linksCopy.length == 0) {
console.log("ESCI link trovati", tempLinks);
break;
}
if (i == 0) {
sel_vect = linksCopy[0];
linksCopy.splice(0, 1);
i = tLinks.length - 1;
}*/
}
```

```
}
/*for (var j = tLinks.length -1; j > -1; j--) {
if (tLinks[j][3] == vetto) {
tempLinks.push(tLinks[j]);
tLinks.splice(j, 1);
}
}*/
}
}
else if (vetto == val.options[val.selectedIndex].value || val.options[val.selectedIndex].value=="") {
selezionato = 0;
console.log("deseleziona");
// $(this).css("border", "1px solid rgb(211, 211, 211)");
sel_vect = "";
vetto = "";
eliminaLink();
lineeIn = [];
lineeOut = [];
trovato = 0;
shiftOut = 0;
shiftIn = 0;
riempiSpessori();
creaLinks();
rigaSeparazione();
}
/*for (var a = 0; a < tempLinks.length; a++) {
console.log("ecco",tempLinks[a][0],tempLinks[a][1],tempLinks[a][2],tempLinks[a][3]);
}*/
/*timer = 0;
eliminaLink();
lineeIn = [];
lineeOut = [];
trovato = 0;
shiftOut = 0;
shiftIn = 0;
//console.log("templinks",tempLinks.length);
riempiSpessori();
creaLinks();
rigaSeparazione();*/
}
function linkNodo(e) {
for (var j = 0; j < linkOrdinati.length; j++) {
$("#p"+j).unbind("mouseenter");
$("#p" + j).unbind("mouseleave");
}
var indici2 = [];
selezionato = 1;
var tmpNodo = e.target.getAttribute("id");
for(var i=0;i<linkOrdinati.length;i++){
if (linkOrdinati[i][0] == tmpNodo || linkOrdinati[i][1] == tmpNodo) {
//tempLinks.push(linkOrdinati[i]);
indici2.push(i);
document.getElementById("p"+i).getAttribute("style");
var ts = document.getElementById("p" + i).getAttribute("style").split("stroke-opacity")[0];
//, "stroke:" + col + "; fill:none; stroke-opacity:1; stroke-width:" + linkOrdinati[i][2] + ";");
document.getElementById("p" + i).setAttribute("style", ts + "stroke-opacity: 1; stroke-width:" + linkOrdinati[i][2] + ";");
}
else {
document.getElementById("p" + i).getAttribute("style");
var ts = document.getElementById("p" + i).getAttribute("style").split("stroke-opacity")[0];
```

```
document.getElementById("p" + i).setAttribute("style", ts + "stroke-opacity: 0.1; stroke-width:" + linkOrdinati[i][2] + ";");
}
}
for (var a = 0; a < indici2.length; a++) {
$("#p" + indici2[a]).mouseenter(pMe);
$("#p" + indici2[a]).mouseleave(function () {
for (var o = 0; o < indici2.length; o++) {
var temps = document.getElementById("p" + indici2[o]).getAttribute("style").split("stroke-opacity")[0];
//, "stroke:" + col + "; fill:none; stroke-opacity:1; stroke-width:" + linkOrdinati[i][2] + ";");
document.getElementById("p" + indici2[o]).setAttribute("style", temps + "stroke-opacity: 1; stroke-width:" + linkOrdinati[indici2[o]][2] + ";");
}
$("#infoLink").hide();
$("#infoLink p").html("");
});
}
/*eliminaLink();
lineeIn = [];
lineeOut = [];
trovato = 0;
shiftOut = 0;
shiftIn = 0;
riempiSpessori();
creaLinks();*/
}
var tmpFam="";
function selFamiglia() {
var val = document.getElementById("numF");
//console.log(selezionato, tmpFam, val.options[val.selectedIndex].value);
if (val.options[val.selectedIndex].value == "") {
console.log("DESELEZIONO");
selezionato = 0;
eliminaLink();
lineeIn = [];
lineeOut = [];
trovato = 0;
shiftOut = 0;
shiftIn = 0;
riempiSpessori();
creaLinks();
rigaSeparazione();
}
else {
console.log("SELEZIONO");
selezionato = 1;
tmpFam = val.options[val.selectedIndex].value;
console.log(tmpFam);
for (var i = 0; i < posizioni.length; i++) {
for (var j = 0; j < linkOrdinati.length; j++) {
if (linkOrdinati[j][3] == posizioni[i][0] && posizioni[i][1] == tmpFam) {
tempLinks.push(linkOrdinati[j]);
}
}
}
eliminaLink();
lineeIn = [];
lineeOut = [];
trovato = 0;
shiftOut = 0;
shiftIn = 0;
riempiSpessori();
```

```
creaLinks();
rigaSeparazione();
}
}
//RICHIESTE DB
function getVettori() {
//vettori = [["gas", "vettore"], ["natural gas", "vettore"], ["heat", "vettore"], ["coal", "vettore"], ["electricity", "vettore"], ["solar", "vettore"]]
xmlhttp.open("GET", "query/vettori.php", true);
xmlhttp.onreadystatechange = function () {
if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
//console.log("completato!");
var risposta = JSON.parse(xmlhttp.responseText);
//console.log("VETTORI ",risposta);
for (var i = 0; i < risposta.length; i++) {
vettori.push([risposta[i].name, "vettori"]);
}
//nodi = nodi.concat(vettori);
//console.log("vettori",vettori.length);
getBilancio();
}
else {
//console.log("waiting...");
}
}
xmlhttp.send();
}
function getBilancio() {
//vettori = [["gas", "vettore"], ["natural gas", "vettore"], ["heat", "vettore"], ["coal", "vettore"], ["electricity", "vettore"], ["solar", "vettore"]]
xmlhttp.open("GET", "query/bilancio.php", true);
xmlhttp.onreadystatechange = function () {
if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
//console.log("completato!");
var risposta = JSON.parse(xmlhttp.responseText);
//console.log("BILANCIO ", risposta);
for (var i = 0; i < risposta.length; i++) {
if (risposta[i].position == "10") {
bilancio.push([ risposta[i].name , "primary sources"]);
}
if (risposta[i].position == "21") {
bilancio2.push([risposta[i].name , "secondary sources"]);
}
if (risposta[i].position == "22") {
bilancio3.push([risposta[i].name , "secondary_sources"]);
}
if (risposta[i].position == "23") {
bilancio4.push([risposta[i].name , "tertiary sources"]);
}
}
nodi = nodi.concat(bilancio);
nodi = nodi.concat(bilancio2);
nodi = nodi.concat(bilancio3);
nodi = nodi.concat(bilancio4);
//console.log(nodi);
//getBilancio2();
getLinkMeter();
}
else {
//console.log("waiting...");
}
}
```

```
xmlhttp.send();
}
function getLinkMeter() {
//vettori = [["gas", "vettore"], ["natural gas", "vettore"], ["heat", "vettore"], ["coal", "vettore"], ["electricity", "vettore"], ["solar", "vettore"]]
xmlhttp.open("GET", "query/link_primacolonna.php", true);
xmlhttp.onreadystatechange = function () {
if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
//console.log("completato!", xmlhttp.responseText);
var risposta = JSON.parse(xmlhttp.responseText);
//console.log("LINK METER ", risposta);
for (var i = 0; i < risposta.length; i++) {
if (risposta[i].quantita != 0) {
if (parseFloat(risposta[i].quantita) >= fattore) {
links.push([risposta[i].tipo, risposta[i].name, risposta[i].quantita / fattore, risposta[i].name]);
}
else {
links.push([risposta[i].tipo, risposta[i].name, 1, risposta[i].name]);
}
}
}
//console.log(nodi);
getTrasformazioni();
}
else {
//console.log("waiting...");
}
}
xmlhttp.send();
}
function getTrasformazioni() {
//var trasformazioni = [["nuclear", "trasformazione"], ["regasification", "trasformazione"], ["refinery", "trasformazione"], ["gasification", "trasformazione"], ["CHP", "trasformazione"], ["Hydropower", "trasformazione"], ["Solar Thermal", "trasformazione"]];
xmlhttp.open("GET", "query/trasformazioni.php", true);
xmlhttp.onreadystatechange = function () {
if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
//console.log("completato!");
var risposta = JSON.parse(xmlhttp.responseText);
//console.log("TRASFORMAZIONI ", risposta);
//var trasformazioni = [];
for (var i = 0; i < risposta.length; i++) {
if (risposta[i].position == "21") {
trasformazioni.push([risposta[i].nome, "mechanical transformations"]);
}
if (risposta[i].position == "22") {
trasformazioni2.push([risposta[i].nome, "chemical transformations"]);
}
if (risposta[i].position == "23") {
trasformazioni3.push([risposta[i].nome, "energy conversions"]);
}
}
nodi = nodi.concat(trasformazioni);
nodi = nodi.concat(trasformazioni2);
nodi = nodi.concat(trasformazioni3);
//console.log(nodi);
getT_input();
}
else {
//console.log("waiting...");
}
```

```
}
xmlhttp.send();
}
function getT_input() {
//vettori = [["gas", "vettore"], ["natural gas", "vettore"], ["heat", "vettore"], ["coal", "vettore"], ["electricity", "vettore"], ["solar", "vettore"]]
xmlhttp.open("GET", "query/trasformazioni_input.php", true);
xmlhttp.onreadystatechange = function () {
if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
//console.log("completato!", xmlhttp.responseText);
var risposta = JSON.parse(xmlhttp.responseText);
//console.log("TRASFORMAZIONI INPUT", risposta);
for (var i = 0; i < risposta.length; i++) {
if (parseFloat(risposta[i].prod_q) == 0) {
if (parseFloat(risposta[i].cons_q) >= fattore) {
links.push([risposta[i].name, risposta[i].trasformazione, risposta[i].cons_q / fattore, risposta[i].name]);
}
else {
links.push([risposta[i].name, risposta[i].trasformazione, 1 , risposta[i].name]);
}
}
if (parseFloat(risposta[i].cons_q) == 0) {
if (parseFloat(risposta[i].prod_q) >= fattore) {
links.push([risposta[i].trasformazione, risposta[i].name, risposta[i].prod_q / fattore, risposta[i].name]);
}
else {
links.push([risposta[i].trasformazione, risposta[i].name, 1, risposta[i].name]);
}
}
}
//nodi = nodi.concat(bilancio2);
//console.log(nodi);
getmeter();
}
else {
//console.log("waiting...");
}
}
xmlhttp.send();
}
function getmeter() {
//vettori = [["gas", "vettore"], ["natural gas", "vettore"], ["heat", "vettore"], ["coal", "vettore"], ["electricity", "vettore"], ["solar", "vettore"]]
xmlhttp.open("GET", "query/link_meter.php", true);
xmlhttp.onreadystatechange = function () {
if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
//console.log("completato!", xmlhttp.responseText);
var risposta = JSON.parse(xmlhttp.responseText);
//console.log("TRASFORMAZIONI INPUT", risposta);
for (var i = 0; i < risposta.length; i++) {
if (risposta[i].quantita != 0) {
if (parseFloat(risposta[i].quantita) >= fattore) {
links.push([risposta[i].name, "final user", risposta[i].quantita / fattore, risposta[i].name]);
}
else {
links.push([risposta[i].name, "final user", 1, risposta[i].name]);
}
}
}
//nodi = nodi.concat(bilancio2);
//console.log(nodi);
```

```
getPosizioni();
}
else {
//console.log("waiting...");
}
}
xmlhttp.send();
}
function getPosizioni() {
//vettori = [["gas", "vettore"], ["natural gas", "vettore"], ["heat", "vettore"], ["coal", "vettore"], ["electricity", "vettore"], ["solar", "vettore"]]
xmlhttp.open("GET", "query/posizioni.php", true);
xmlhttp.onreadystatechange = function () {
if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
//console.log("completato!", xmlhttp.responseText);
var risposta = JSON.parse(xmlhttp.responseText);
//console.log("TRASFORMAZIONI INPUT", risposta);
for (var i = 0; i < risposta.length; i++) {
posizioni.push([risposta[i].name, risposta[i].fk_id_famiglia, risposta[i].ordine]);
lastF = risposta[i].fk_id_famiglia;
}
console.log("ULTIMA FAMIGLIA",lastF);
posizioni.unshift(["final user", 0, 1]);
posizioni.unshift(["storage", 0, 4]);
posizioni.unshift(["export", 0, 3]);
posizioni.unshift(["import", 0, 2]);
posizioni.unshift(["indigenous production", 0, 1]);
nodi = nodi.concat(meter);
//nodi = nodi.concat(bilancio2);
console.log("POSIZONI", posizioni);
getUses();
}
else {
//console.log("waiting...");
}
}
xmlhttp.send();
}
var posizioni1 = [];
var posizioni2 = [];
var posizioni3 = [];
function getUses() {
//vettori = [["gas", "vettore"], ["natural gas", "vettore"], ["heat", "vettore"], ["coal", "vettore"], ["electricity", "vettore"], ["solar", "vettore"]]
xmlhttp.open("GET", "query/uses.php", true);
xmlhttp.onreadystatechange = function () {
if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
//console.log("completato!", xmlhttp.responseText);
var risposta = JSON.parse(xmlhttp.responseText);
//console.log("TRASFORMAZIONI INPUT", risposta);
for (var i = 0; i < risposta.length; i++) {
uses.push([risposta[i].uso, "meter by uses"]);
lastMeter.push(["meter_" + risposta[i].uso, "ultimate uses"]);
posizioni1.push(risposta[i].uso);
posizioni2.push(lastMeter[i][0]);
}
nodi = nodi.concat(uses);
getLinkFinal();
}
else {
//console.log("waiting...");
```

```
}
}
xmlhttp.send();
}
function getLinkFinal() {
//vettori = [["gas", "vettore"], ["natural gas", "vettore"], ["heat", "vettore"], ["coal", "vettore"], ["electricity", "vettore"], ["solar", "vettore"]]
xmlhttp.open("GET", "query/link_meter_final.php", true);
xmlhttp.onreadystatechange = function () {
if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
//console.log("completato!", xmlhttp.responseText);
var risposta = JSON.parse(xmlhttp.responseText);
//console.log("TRASFORMAZIONI INPUT", risposta);
for (var i = 0; i < risposta.length; i++) {
if (risposta[i].quantita != 0) {
if (parseFloat(risposta[i].quantita) >= fattore) {
links.push(["final user", risposta[i].uso, risposta[i].quantita / fattore, risposta[i].name]);
}
else {
links.push(["final user", risposta[i].uso, 1, risposta[i].name]);
}
}
}
//nodi = nodi.concat(bilancio2);
//console.log(nodi);
getnodiFinal();
}
else {
//console.log("waiting...");
}
}
xmlhttp.send();
}
function getnodiFinal() {
//vettori = [["gas", "vettore"], ["natural gas", "vettore"], ["heat", "vettore"], ["coal", "vettore"], ["electricity", "vettore"], ["solar", "vettore"]]
xmlhttp.open("GET", "query/nodi_final_transformation.php", true);
xmlhttp.onreadystatechange = function () {
if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
//console.log("completato!", xmlhttp.responseText);
var risposta = JSON.parse(xmlhttp.responseText);
//console.log("TRASFORMAZIONI INPUT", risposta);
for (var i = 0; i < risposta.length; i++) {
nodiFinal.push([risposta[i].nome+"_"+risposta[i].uso, "final uses"]);
posizioni3.push(risposta[i].nome+"_"+risposta[i].uso);
}
nodi = nodi.concat(nodiFinal);
nodi = nodi.concat(lastMeter);
//console.log(nodi);
getLinkFinalProduction();
}
else {
//console.log("waiting...");
}
}
xmlhttp.send();
}
function getLinkFinalProduction() {
//vettori = [["gas", "vettore"], ["natural gas", "vettore"], ["heat", "vettore"], ["coal", "vettore"], ["electricity", "vettore"], ["solar", "vettore"]]
xmlhttp.open("GET", "query/link_final_production.php", true);
```

```
xmlhttp.onreadystatechange = function () {
if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
//console.log("completato!", xmlhttp.responseText);
var risposta = JSON.parse(xmlhttp.responseText);
//console.log("TRASFORMAZIONI INPUT", risposta);
for (var i = 0; i < risposta.length; i++) {
if (risposta[i].quantita != 0) {
if (parseFloat(risposta[i].quantita) >= fattore) {
links.push([risposta[i].nome + "_" + risposta[i].uso, "meter_" + risposta[i].uso, risposta[i].quantita / fattore, risposta[i].name]);
}
else {
links.push([risposta[i].nome + "_" + risposta[i].uso, "meter_" + risposta[i].uso, 1, risposta[i].name]);
}
}
}
//nodi = nodi.concat(bilancio2);
//console.log(nodi);
getLinkFinalGridSupply();
}
else {
//console.log("waiting...");
}
}
xmlhttp.send();
}
function getLinkFinalGridSupply() {
//vettori = [["gas", "vettore"], ["natural gas", "vettore"], ["heat", "vettore"], ["coal", "vettore"], ["electricity", "vettore"], ["solar", "vettore"]]
xmlhttp.open("GET", "query/final_grid_supply_meter.php", true);
xmlhttp.onreadystatechange = function () {
if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
//console.log("completato!", xmlhttp.responseText);
var risposta = JSON.parse(xmlhttp.responseText);
//console.log("TRASFORMAZIONI INPUT", risposta);
for (var i = 0; i < risposta.length; i++) {
if (risposta[i].quantita != 0) {
if (parseFloat(risposta[i].direct_cons) >= fattore) {
links.push([risposta[i].uso, "meter_" + risposta[i].uso, risposta[i].direct_cons / fattore, risposta[i].name]);
}
else {
links.push([risposta[i].uso, "meter_" + risposta[i].uso, 1, risposta[i].name]);
}
}
}
//nodi = nodi.concat(bilancio2);
//console.log(nodi);
getLinkFinalTransformation();
}
else {
//console.log("waiting...");
}
}
xmlhttp.send();
}
function getLinkFinalTransformation() {
//vettori = [["gas", "vettore"], ["natural gas", "vettore"], ["heat", "vettore"], ["coal", "vettore"], ["electricity", "vettore"], ["solar", "vettore"]]
xmlhttp.open("GET", "query/link_final_transformation.php", true);
xmlhttp.onreadystatechange = function () {
if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
//console.log("completato!", xmlhttp.responseText);
```

```
var risposta = JSON.parse(xmlhttp.responseText);
//console.log("TRASFORMAZIONI INPUT", risposta);
for (var i = 0; i < risposta.length; i++) {
if (risposta[i].quantita != 0) {
if (parseFloat(risposta[i].quantita) >= fattore) {
links.push([risposta[i].uso, risposta[i].nome + "_" + risposta[i].uso, risposta[i].quantita / fattore, risposta[i].name]);
}
else {
links.push([risposta[i].uso, risposta[i].nome + "_" + risposta[i].uso, 1, risposta[i].name]);
}
}
}
console.log(posizioni1, posizioni2, posizioni3);
var q = 0;
for (var w = 0; w < posizioni1.length; w++) {
posizioni.push([posizioni1[w], parseInt(lastF) + 1 + w, 1]);
for (var ww = 0; ww < posizioni2.length; ww++) {
if (posizioni2[ww].split("_")[1] == posizioni1[w]) { posizioni.push([posizioni2[ww], parseInt(lastF) + 1 + w, 2]); }
}
for (var www = 0; www < posizioni3.length; www++) {
if (posizioni3[www].split("_")[1] == posizioni1[w]) { posizioni.push([posizioni3[www], parseInt(lastF) + 1 + w, 3 + q]); q++; }
}
q = 0;
}
//nodi = nodi.concat(bilancio2);
//console.log(nodi);
init();
}
else {
//console.log("waiting...");
}
}
xmlhttp.send();
}

function blocchi() {
for (var i = 0; i < colonne.length; i++) {
var cerchio = document.createElementNS('http://www.w3.org/2000/svg', 'circle');
cerchio.setAttribute("id", colonne[i]);
cerchio.setAttribute("cx", (i * (screenW / colonne.length)) + 20);
cerchio.setAttribute("cy", 80);
cerchio.setAttribute("r", 10);
cerchio.setAttribute("style", "fill:url(#icona_meno); opacity:0");
cerchio.addEventListener("click", riduci, false);
svg.appendChild(cerchio);
var cerchio = document.createElementNS('http://www.w3.org/2000/svg', 'circle');
cerchio.setAttribute("id", colonne[i] + "_espandi");
cerchio.setAttribute("cx", (i * (screenW / colonne.length)) + 20);
cerchio.setAttribute("cy", 40);
cerchio.setAttribute("r", 10);
cerchio.setAttribute("opacity", 0);
cerchio.setAttribute("style", "fill:url(#icona_piu); opacity:0");
cerchio.addEventListener("click", espandi, false);
svg.appendChild(cerchio);
//.setAttribute("x",i*80);
var testo = document.createElementNS('http://www.w3.org/2000/svg', 'text');
testo.setAttribute("id", "t"+i);
testo.setAttribute("x", (i * (screenW / colonne.length)) + 30);
testo.setAttribute("y", 65);
```

```
testo.setAttribute("font-weight", "bold");
var textNode = document.createTextNode(colonne[i].toUpperCase());
testo.appendChild(textNode);
//svg.appendChild(testo);
}
}
function getAltezze() {
var oldC = 0;
var pos = [];
for (var k = 0; k < colonne.length; k++) {
pos[k] = 0;
}
for (var i = 0; i < nodiOrdinati.length; i++) {
for (var k = 0; k < colonne.length; k++) {
if (colonne[k] == nodiOrdinati[i][1]) {
var rif = colonne[k];
if (k > 0) {
var colore = colori[(parseInt(k) + parseInt(pos[0]))];
//quota = 250;
//console.log("COLORE "+colore);
}
else {
var colore = colori[i];
//quota = 750;
}
if (oldC != k) {
//console.log("KAPPA " + k);
}
break;
}
}
oldC = k;
//pos[k]++;
var blocchi = document.getElementsByTagName("circle");
for (var w = 0; w < blocchi.length; w++) {
if (blocchi[w].getAttribute("id") == rif) { var blocco = blocchi[w]; }
}
var h = 1;
var u = 0;
var e = 0;
for (var j = 0; j < linkOrdinati.length; j++) {
if (linkOrdinati[j][0] == nodiOrdinati[i][0]) {
//console.log("altezza");
e = e + parseFloat(linkOrdinati[j][2]);
}
if (linkOrdinati[j][1] == nodiOrdinati[i][0]) {
//console.log("altezza");
u = u + parseFloat(linkOrdinati[j][2]);
}
}
h = Math.max(e, u);
//if (h > 0 && h <= 1) { h = 1; }
altezze.push([nodiOrdinati[i][0], h, blocco.id]);
}
//altezze2.push(altezze[0], altezze[1], altezze[2], altezze[3], altezze[71]);
for (var i = 0; i < posizioni.length; i++) {
var n = posizioni[i][0];
for (var j = 0; j < altezze.length; j++) {
if (n == altezze[j][0]) {
```

```
altezze2.push(altezze[j]);
}
}
}
console.log("ALTEZZE", altezze2);
var p=0;
var tmpArray = [];
var max = 0;
for (var i = 0; i < posizioni.length; i++) {
var n = posizioni[i][0];
if (p != parseInt(posizioni[i][1])) {
//console.log("P", p, parseInt(posizioni[i][1]));
max = 0;
for (var q = 1; q < tmpArray.length; q++) { max += tmpArray[q]; }
altezzeSub[p] = tmpArray;
famiglie[p] = max;
p = parseInt(posizioni[i][1]);
tmpArray = [];
max = 0;
}
for (var j = 0; j < altezze2.length; j++) {
if (n == altezze2[j][0]) {
//console.log("BLOCCO", n, altezze2[j][1]);
//if (p == 15) { alert();}
if (tmpArray[parseInt(posizioni[i][2])] != undefined) {
//console.log("UGUALI", Math.max(tmpArray[posizioni[i][2]], altezze2[j][1]), tmpArray[posizioni[i][2]], altezze2[j][1]);
tmpArray[parseInt(posizioni[i][2])] = Math.max(tmpArray[posizioni[i][2]], altezze2[j][1]);
}
else {
tmpArray[parseInt(posizioni[i][2])] = altezze2[j][1];
}
//altezze2.push(altezze[j]);
}
}
if ((i == (posizioni.length - 1))) {
//console.log("P", p, parseInt(posizioni[i][1]));
max = 0;
for (var q = 1; q < tmpArray.length; q++) { max += tmpArray[q]; }
altezzeSub[p] = tmpArray;
famiglie[p] = max;
p = parseInt(posizioni[i][1]);
tmpArray = [];
max = 0;
}
}
console.log("ALTEZZE SUB", altezzeSub);
/* var column;
var row;
var oldColumn;
var oldRow;
for (var q = 0; q < posizioni.length; q++) {
for (var z = 0; z < altezze2.length; z++) {
column = altezze2[z][2];
row = posizioni[q][1];
if (posizioni[q][0] == altezze2[z][0]) {
if (famiglie[posizioni[q][1]] != undefined) {
if (column == oldColumn) {
famiglie[posizioni[q][1]] = famiglie[posizioni[q][1]] + altezze2[z][1];
//console.log("somma",famiglie[posizioni[q][1]]);
```

```
}
else {
famiglie[posizioni[q][1]] = Math.max(famiglie[posizioni[q][1]], altezze2[z][1]);
//console.log("max", Math.max(famiglie[posizioni[q][1]], altezze2[z][1]), famiglie[posizioni[q][1]], altezze2[z][1]);
}
}
else {
famiglie[posizioni[q][1]] = altezze2[z][1];
}
}
oldColumn = altezze2[z][2];
oldRow = posizioni[q][1];
}
}*/
//famiglie[0] = altezze2[0][1] + altezze2[1][1] + altezze2[2][1] + altezze2[3][1] + altezze2[4][1];
console.log("FAMIGLIE", famiglie, famiglie.length);
}
function ordina() {
//nodiOrdinati.push(nodi[0], nodi[1], nodi[2], nodi[3], nodi[71]);
for (var i = 0; i < posizioni.length; i++) {
var n = posizioni[i][0];
for (var j = 0; j < nodi.length; j++) {
if (n == nodi[j][0]) {
nodiOrdinati.push(nodi[j]);
}
}
}
for (var a = 0; a < nodiOrdinati.length; a++) {
var m = nodiOrdinati[a][0];
for (var b= 0; b < links.length; b++) {
if (m == links[b][0]) {
linkOrdinati.push(links[b]);
}
}
}
for (var v = 0; v < colonne.length; v++) {
if (v == 1 || v == 3 || v == 5 || v == 7) {
spessori.push([colonne[v], (100+widthB)]);
} else {
spessori.push([colonne[v], (100+width)]);
}
tempspesOut.push([colonne[v], 0]);
tempspesIn.push([colonne[v], 0]);
}
console.log("nodiOrdinati", nodiOrdinati);
console.log("linkOrdinati", linkOrdinati, linkOrdinati.length);
console.log("SPESSORi", spessori);
}
function creaNodi() {
/* for (var s = 0; s < links.length; s++) {
for (var d = 0; d < nodiOrdinati.length; d++) {
if (links[s][0] == nodiOrdinati[d][0]) {
for (var z = 0; z < tempspes.length; z++) {
if (nodiOrdinati[d][1] == tempspes[z][0]) { tempspes[z][1] += (links[s][2] + (links[s][2] / 2)); }
}
}
if (links[s][1] == nodiOrdinati[d][0]) {
for (var z = 0; z < tempspes.length; z++) {
if (nodiOrdinati[d][1] == tempspes[z][0]) { tempspes[z][1] += (links[s][2] + (links[s][2] / 2)); }
```

```
}
}
}
}
console.log("TEMP SPESS", tempspes);*/
var shift = 0;
var oldC = 0;
var h = 1;
var quota = [];
var pos = [];
var subpos;
var index;
for (var k = 0; k < colonne.length; k++) {
pos[k] = 0;
quota[k] = [];
}
for (var i = 0; i < nodiOrdinati.length; i++) {
var rect = document.createElementNS('http://www.w3.org/2000/svg', 'rect');
rect.setAttribute("id", nodiOrdinati[i][0]);
rect.setAttribute("class", "nodorect");
/* for (var k = 0; k < colonne.length;k++){
if (colonne[k] == nodiOrdinati[i][1]) {
var rif = colonne[k];
if (k > 0) {
var colore = colori[(parseInt(k) + parseInt(pos[0]))];
//quota = 250;
//console.log("COLORE "+colore);
}
else {
var colore = colori[i];
//quota = 750;
}
/*if (oldC != k) {
//shift = 0;
offset = 0;
//quota = 0;
//h = 0;
console.log("CAMBIO COLONNA ", nodiOrdinati[i][0]);
}
break;
}
}
oldC = k;*/
/*for (var k = 0; k < vettori.length;k++){
if (vettori[k][0] == nodiOrdinati[i][0]) {
console.log(vettori[k][0], nodiOrdinati[i][0], colori[k]);
var colore = colori[k];
var largh = widthB;
break;
}
else {
var colore = "#c0c0c0";
var largh = width;
}
}*/
for (var k = 0; k < posizioni.length; k++) {
if (posizioni[k][0] == nodiOrdinati[i][0]) {
//console.log(vettori[k][0], nodiOrdinati[i][0], colori[k]);
var colore = colFam[posizioni[k][1]];
for (var kk = 0; kk < vettori.length; kk++) {
```

```
if (vettori[kk][0] == nodiOrdinati[i][0]) {
var largh = widthB;
break;
}
else {
var largh = width;
}
}
break;
}
else {
var colore = "#c0c0c0";
var largh = width;
}
}
//pos[k]++;
var blocchi = document.getElementsByTagName("circle");
var xp = 50;
for (var w = 0; w < spessori.length; w++) {
var txt = document.getElementById("t" + w);
if (w == 0) {
//txt.setAttribute("x", 100);
}
else {
xp += (spessori[w - 1][1] + largh);
//txt.setAttribute("x", xp);
}
if (nodiOrdinati[i][1] == "gross consumption") { var xposition = 100; }
else if (spessori[w][0] == nodiOrdinati[i][1]) {
var xposition=100;
for (var r = 0; r < w; r++) {
xposition += spessori[r][1];
}
}
}
var u = 0;
var e = 0;
for (var j = 0; j < linkOrdinati.length; j++) {
if (linkOrdinati[j][0] == nodiOrdinati[i][0]) {
//console.log("altezza");
e = e + parseFloat(linkOrdinati[j][2]);
}
if (linkOrdinati[j][1] == nodiOrdinati[i][0]) {
//console.log("altezza");
u = u + parseFloat(linkOrdinati[j][2]);
}
}
h = Math.max(e,u);
//if (h > 0 && h <= 1) { h = 1; }
//altezze.push([nodi[i][0], h, blocco.id]);
/*for (var j = 0; j < altezze2.length; j++) {
if (altezze2[j][0] == nodiOrdinati[i][0]) {
h = altezze2[j][1];
}
}*/
for (var t = 0; t < posizioni.length; t++) {
if (posizioni[t][0] == nodiOrdinati[i][0]) {
//console.log("TROVO");
index = parseInt(posizioni[t][1]);
subpos = parseInt(posizioni[t][2]);
```

```
for (var q = 1; q < subpos; q++) {
shift += (parseFloat(altezzeSub[index][q])*2);
}
for (var o = 0; o < index; o++) {
//shift += parseFloat(famiglie[o]);
shift += (vShift * altezzeSub[o].length);
for (var q = 1; q < altezzeSub[o].length; q++) {
shift += (parseFloat(altezzeSub[o][q])*2);
}
if (posizioni[t][1] == lastF && o == index - 1) {
var meno = shift;
console.log("ULTIMA POSIZIONE", shift, h);
}
}
// console.log(posizioni[t][0], shift, index, subpos);
if (posizioni[t][1] > parseInt(lastF)) {
console.log("QUI", posizioni[t][1]);
shift = shift - meno;
}
break;
}
}
var offset = 0;
/*if (quota[oldC][index] != undefined) {
offset = quota[oldC][index];
}*/
for (var t = 1; t < subpos; t++) {
offset += altezzeSub[index][t];
}
rect.setAttribute("x", xposition);
rect.setAttribute("y", shift + 50);
//console.log(i, parseInt(blocco.getAttribute("cy")) * (pos[k] + 1));
rect.setAttribute("width", largh);
rect.setAttribute("height", h);
//rect.setAttribute("rx", 3);
//rect.setAttribute("ry", 3);
rect.setAttribute("stroke", colore);
if (h < 1) {
rect.setAttribute("opacity", 0);
}
else {
rect.setAttribute("opacity", 0.9);
}
rect.setAttribute("stroke-width", 1);
rect.setAttribute("fill", colore);
rect.addEventListener("click", info, false);
//rect.addEventListener("mouseover", info, false);
//rect.addEventListener("mouseout", close, false);
//console.log("VALORE: ", (h - 50) / 2);
//rect.setAttribute("style", "fill:url(#icona_casa)");
svg.appendChild(rect);
var testo = document.createElementNS('http://www.w3.org/2000/svg', 'text');
testo.setAttribute("class", "txt");
testo.setAttribute("text-anchor", "start");
testo.setAttribute("style", "font-family: Arial; font-size: 10px;");
testo.setAttribute("x", xposition);
testo.setAttribute("y", shift + 0 - 5);
if (h < 1) {
//console.log("altezza zero");
testo.setAttribute("opacity", 0);
```

```
}
var textNode = document.createTextNode(nodiOrdinati[i][0].toUpperCase());
testo.appendChild(textNode);
//svg.appendChild(testo);
pos[k]++;
shift = 0;
if (quota[oldC][index] != undefined) {
quota[oldC][index] = parseFloat(quota[oldC][index]) + h;
}
else {
quota[oldC][index] = h;
}
}
$(".nodorect")
.mouseenter(function () {
var nome = this.getAttribute("id");
var testo3 = "<b>"+nome+"</b>";
$("#infoRect").css({ "width": "" + (55 + (nome.length * 7)) + "px" });
//$("#info").animate({ top: "0" }, 1000, "linear");
$("#infoRect").css("left", event.clientX);
$("#infoRect").css("top", event.clientY);
$("#infoRect").show();
$("#infoRect img").attr("src", "icone/indigenous_production.png");
//$("#infoRect img").attr("src", "icone/" + nome.replace(" ", "_") + ".png");
$("#infoRect img").css("float", "left");
$("#infoRect p").css("float","right");
$("#infoRect p").html(testo3);
//console.log("icone/" + nome.replace(" ", "_") + ".png");
})
.mouseleave(function () {
$("#infoRect").hide();
$("#infoRect p").html("");
});
//console.log("QUOTA", quota);
}
function creaLinks() {
console.log("LINK ", tempLinks.length);
if (selezionato == 1) {
//console.log(tempLinks);
linksBKup = linkOrdinati; linkOrdinati = tempLinks;
console.log("SPESSORI", spessori, "TMP SPES OUT", tempspesOut, "TMP SPES IN", tempspesIn);
}
if (selezionato == 0) {
console.log("creo tutto", linksBKup);
links = linksBKup; tempLinks = [];
console.log("SPESSORI",spessori,"TMP SPES OUT",tempspesOut,"TMP SPES IN", tempspesIn);
}
var trovato = 0;
var shiftOut = 0;
var shiftIn = 0;
for (var i = 0; i < linkOrdinati.length; i++) {
var rect = document.getElementsByTagName("rect");
for (var j = 0; j < rect.length; j++) {
if (linkOrdinati[i][0] == rect[j].getAttribute("id")) {
//console.log("trovo source");
for (var k = 0; k < lineeOut.length; k++) {
if (lineeOut[k][0] == linkOrdinati[i][0]) {
//console.log("esiste "+links[i][0]);
shiftOut = lineeOut[k][1];
```

```
lineeOut[k][1] = parseInt(lineeOut[k][1]) + parseInt(linkOrdinati[i][2]);
trovato = 1;
}
}
if (trovato == 0) {
//console.log("aggiungo " + links[i][0]);
lineeOut.push([linkOrdinati[i][0], linkOrdinati[i][2]]);
}
var x0 = parseInt(rect[j].getAttribute("x")) + parseInt(rect[j].getAttribute("width"));
var y0 = parseFloat(rect[j].getAttribute("y")) + (parseFloat(linkOrdinati[i][2]) / 2) + parseInt(shiftOut);
var ww=parseInt(rect[j].getAttribute("width"));
}
trovato = 0;
if (linkOrdinati[i][1] == rect[j].getAttribute("id")) {
//console.log("trovo target");
for (var w = 0; w < lineeIn.length; w++) {
if (lineeIn[w][0] == linkOrdinati[i][1]) {
//console.log("esiste " + links[i][1]);
shiftIn = lineeIn[w][1];
lineeIn[w][1] = parseInt(lineeIn[w][1]) + parseInt(linkOrdinati[i][2]);
trovato = 1;
}
}
if (trovato == 0) {
//console.log("aggiungo " + links[i][1]);
lineeIn.push([linkOrdinati[i][1], linkOrdinati[i][2]]);
}
var x1 = rect[j].getAttribute("x");
var y1 = parseFloat(rect[j].getAttribute("y")) + (parseFloat(linkOrdinati[i][2]) / 2) + parseInt(shiftIn);
}
trovato = 0;
}
for (var v = 0; v < nodiOrdinati.length; v++) {
if (linkOrdinati[i][0]==nodiOrdinati[v][0]) {
var colP = nodiOrdinati[v][1];
colP=colonne.indexOf(colP);
}
if (linkOrdinati[i][1] == nodiOrdinati[v][0]) {
var colA = nodiOrdinati[v][1];
colA = colonne.indexOf(colA);
}
}
//console.log(colP, colA);
//VECCHI LINK
/*if (parseInt(x0) < parseInt(x1)) {
//var cx0 = x0 + ((screenW / colonne.length) - 50) / 2;
//var cx1 = x1 - ((screenW / colonne.length) - 50) / 2;
var cx0 = x0 + ((x1-x0) / 2);
var cx1 = x1 - ((x1-x0) / 2);
var cy0 = y0;
var cy1 = y1;
}
if (parseInt(x1) < parseInt(x0)) {
var cx0 = x0;
var cx1 = x1;
if (parseInt(y0) < parseInt(y1)) {
var cy0 = y1 - ((y1 - y0) / 2);
var cy1 = y0 + ((y1 - y0) / 2);
}
else {
```

```
var cy0 = y1+ ((y0 - y1) / 2);
var cy1 = y0-((y0 - y1) / 2);
}
}*/
//curve
//var path = "M" + x0 + "," + y0 + " C" + cx0 + "," + cy0 + " " + cx1 + "," + cy1 + " " + x1 + "," + y1;
//spezzate
//var path = "M" + x0 + "," + y0 + " L" + cx0 + "," + cy0 + " L" + cx1 + "," + cy1 + " L" + x1 + "," + y1;
//PROVA NUOVI LINK
var nodoP = tempspesOut[colP][1];
var nodoA = tempspesIn[colA][1];
if ((colA - colP) == 1) {
if (y0 < y1) {
var cx3 = x0 + spessori[colP][1] - (linkOrdinati[i][2] / 2) - tempspesIn[colP][1] - 50 - ww;
var cx1 = cx3;
//var cx2 = cx3;
var cy1 = y0;
var cy3 = y1;
var path = "M" + x0 + "," + y0 + " L" + cx1 + "," + cy1 + " L" + cx3 + "," + cy3 + " L" + x1 + "," + y1;
tempspesIn[colP][1] += (linkOrdinati[i][2]);
}
else {
var cx3 = x0 + (linkOrdinati[i][2] / 2) + tempspesOut[colP][1] + 50 ;
var cx1 = cx3;
//var cx2 = cx3;
var cy1 = y0;
var cy3 = y1;
var path = "M" + x0 + "," + y0 + " L" + cx1 + "," + cy1 + " L" + cx3 + "," + cy3 + " L" + x1 + "," + y1;
tempspesOut[colP][1] += (linkOrdinati[i][2]);
}
}
else if(y0 < y1) {
//console.log(linkOrdinati[i], colonne[colP], colonne[colA-1] );
var cx0 = x0 + spessori[colP][1] - (linkOrdinati[i][2] / 2) - tempspesIn[colP][1] - 50 - ww;
var cx1 = cx0;
var cx3 = x1 - (linkOrdinati[i][2] / 2) - tempspesIn[colA-1][1] - 50;
var cx2 = cx3;
var cy0 = y0;
var cy3 = y1;
for (var p = 0; p < posizioni.length; p++) {
if (linkOrdinati[i][0] == posizioni[p][0]) { var cy1 = altezzeSub[parseInt(posizioni[p][1])][parseInt(posizioni[p][2])] + cy0; }
if (linkOrdinati[i][1] == posizioni[p][0]) { var cy2 = altezzeSub[parseInt(posizioni[p][1])][parseInt(posizioni[p][2])] + cy3; }
//console.log(posizioni[p][0], cy1, cy2, posizioni[p][1], posizioni[p][2]);
}
if (cy1 > cy2) { cy1 = cy2; }
else { cy2 = cy1; }
var path = "M" + x0 + "," + y0 + " L" + cx0 + "," + cy0 + " L" + cx1 + "," + cy1 + " L" + cx2 + "," + cy2 + " L" + cx3 + "," + cy3 + " L" + x1 + "," + y1;
//tempspesOut[colP][1] += (linkOrdinati[i][2]);
tempspesIn[colP][1] += (linkOrdinati[i][2]);
tempspesIn[colA-1][1] += (linkOrdinati[i][2]);
}
else if (y0 > y1) {
var cx0 = x0 + (linkOrdinati[i][2] / 2) + tempspesOut[colP][1] + 50;
var cx1 = cx0;
//console.log(x1, spessori[colA - 1][1] ,(linkOrdinati[i][2] / 2) , tempspesOut[colA - 1][1]);
if (colA - 1 == 1 || colA - 1 == 3 || colA - 1 == 5 || colA - 1 == 7) { var wi = widthB; } else { var wi = width;}
var cx3 = x1 - spessori[colA-1][1] + (linkOrdinati[i][2] / 2) + tempspesOut[colA-1][1] + 50 + wi;
var cx2 = cx3;
var cy0 = y0;
```

```
var cy3 = y1;
for (var p = 0; p < posizioni.length; p++) {
if (linkOrdinati[i][0] == posizioni[p][0]) { var cy1 = altezzeSub[parseInt(posizioni[p][1])][parseInt(posizioni[p][2])] + cy0; }
if (linkOrdinati[i][1] == posizioni[p][0]) { var cy2 = altezzeSub[parseInt(posizioni[p][1])][parseInt(posizioni[p][2])] + cy3; }
//console.log(posizioni[p][0], cy1, cy2, posizioni[p][1], posizioni[p][2]);
}
if (cy1 > cy2) { cy1 = cy2; }
else { cy2 = cy1; }
var path = "M" + x0 + "," + y0 + " L" + cx0 + "," + cy0 + " L" + cx1 + "," + cy1 + " L" + cx2 + "," + cy2 + " L" + cx3 + "," + cy3 + " L" + x1 + "," + y1;
tempspesOut[colP][1] += (linkOrdinati[i][2]);
tempspesOut[colA - 1][1] += (linkOrdinati[i][2]);
}
/*
if ((colA - colP) == 1) {
var cx3 = x0 + spessori[colP][1] - (linkOrdinati[i][2] / 2) - tempspesIn[colP][1];
var cx1 = cx3;
//var cx2 = cx3;
var cy1 = y0;
var cy3 = y1;
var path = "M" + x0 + "," + y0 + " L" + cx1 + "," + cy1 + " L" + cx3 + "," + cy3 + " L" + x1 + "," + y1;
}
else if (y0 > y1) {
var cx0 = x0 + spessori[colP][1] - (linkOrdinati[i][2] / 2) - tempspesIn[colP][1];
var cx1 = cx0;
var cx3 = x1 - (linkOrdinati[i][2] / 2) - nodoA;
var cx2 = cx3;
var cy0 = y0;
var cy3 = y1;
for (var p = 0; p < posizioni.length; p++) {
if (linkOrdinati[i][0] == posizioni[p][0]) { var cy1 = altezzeSub[parseInt(posizioni[p][1])][parseInt(posizioni[p][2])] + cy0; }
if (linkOrdinati[i][1] == posizioni[p][0]) { var cy2 = altezzeSub[parseInt(posizioni[p][1])][parseInt(posizioni[p][2])] + cy3; }
//console.log(posizioni[p][0], cy1, cy2, posizioni[p][1], posizioni[p][2]);
}
if (cy1 > cy2) { cy1 = cy2; }
else { cy2 = cy1; }
var path = "M" + x0 + "," + y0 + " L" + cx0 + "," + cy0 + " L" + cx1 + "," + cy1 + " L" + cx2 + "," + cy2 + " L" + cx3 + "," + cy3 + " L" + x1 + "," + y1;
}
*/
//var path = "M" + x0 + "," + y0 + " L" + cx0 + "," + cy0 + " L" + cx1 + "," + cy1 + " L" + x1 + "," + y1;
var link = document.createElementNS('http://www.w3.org/2000/svg', 'path');
link.setAttribute("id", "p"+i);
link.setAttribute("d", path);
for (var y = 0; y < vettori.length; y++) {
if (linkOrdinati[i][0] == vettori[y][0]) { link.setAttribute("title", "consumption to: " + linkOrdinati[i][1] + ""); }
if (linkOrdinati[i][1] == vettori[y][0]) {
if (linkOrdinati[i][0] == "indigenous production" || linkOrdinati[i][0] == "import" || linkOrdinati[i][0] == "export" || linkOrdinati[i][0] == "storage") {
link.setAttribute("title", "");
}
else {
link.setAttribute("title", "production from: " + linkOrdinati[i][0] + "");
}
}
else {
link.setAttribute("title", "consumption to: " + linkOrdinati[i][1] + "");
}
}
/* for (var z = 0; z < vettori.length; z++) {
if (linkOrdinati[i][3] == vettori[z][0]) {
var col = colori[z];
```

```
var tmp_class = vettori[z][0];
link.setAttribute("class", vettori[z][0]);
//console.log("colore "+col+ " index "+z);
}
}*/
for (var z = 0; z < posizioni.length; z++) {
if (linkOrdinati[i][3] == posizioni[z][0]) {
//console.log("fam", posizioni[z][1], "vettore", linkOrdinati[i][3], colFam[posizioni[z][1]]);
var col = coloriLink[posizioni[z][1]];
var tmp_class = linkOrdinati[i][3];
link.setAttribute("class", linkOrdinati[i][3]);
//console.log("colore "+col+ " index "+z);
break;
}
else {
var col = "black";
}
}
link.setAttribute("stroke-linecap", "butt");
/*if (y0 < 160 || y1 < 160) {
if (tmp_class == sel_vect) {
link.setAttribute("style", "stroke:" + col + "; fill:none; stroke-opacity:1; stroke-width:" + links[i][2] + ";");
}
else {
link.setAttribute("style", "stroke:" + col + "; fill:none; stroke-opacity:0.1; stroke-width:" + links[i][2] + ";");
}
}*/
if (x0 == xcol || x1 == xcol || x0 == (xcol + ww) || x1 == (xcol + ww)) {
if (tmp_class == sel_vect) {
link.setAttribute("style", "stroke:" + col + "; fill:none; stroke-opacity:1; stroke-width:" + linkOrdinati[i][2] + ";");
}
else {
link.setAttribute("style", "stroke:" + col + "; fill:none; stroke-opacity:1; stroke-width:" + linkOrdinati[i][2] + ";");
}
}
else {
if (tmp_class == sel_vect) {
link.setAttribute("style", "stroke:" + col + "; fill:none; stroke-opacity:1; stroke-width:" + linkOrdinati[i][2] + ";");
}
else {
link.setAttribute("style", "stroke:" + col + "; fill:none; stroke-opacity:1; stroke-width:" + linkOrdinati[i][2] + ";");
}
}
$("g").append(link);
shiftOut = 0;
shiftIn = 0;
/*
for (var y = 0; y < nodiOrdinati.length; y++) {
if (linkOrdinati[i][0] == nodiOrdinati[y][0]) {
for (var d = 0; d < tempspesOut.length; d++) {
if (nodiOrdinati[y][1] == tempspesOut[d][0]) {
if ((colA - colP) != 1) {
tempspesOut[d][1] += (linkOrdinati[i][2]);
}
//console.log(nodiOrdinati[y][1], tempspesOut[d][1]);
}
}
}
if (linkOrdinati[i][1] == nodiOrdinati[y][0]) {
for (var d = 0; d < tempspesIn.length; d++) {
```

```
if (nodiOrdinati[y][1] == tempspesIn[d][0]) {
tempspesIn[d][1] += (linkOrdinati[i][2]);
//console.log(nodiOrdinati[y][1], tempspesIn[d][1]);
}
}
}
}*/
}
$("path")
.mouseenter(pMe)
.mouseleave(pMl);
linkOrdinati = linksBKup; tempLinks = [];
tempspesIn = [];
tempspesOut = [];
spessori = [];
//console.log("TEMP SPESSORI", tempspesOut, tempspesIn);
//console.log("lineeIn: "+lineeIn);
//console.log("lineeOut: " + lineeOut);
}
function pMe() {
$("path").css('stroke-opacity', 0.1);
$(this).css('stroke-opacity', 1);
var percorso = this.getAttribute("title");
var flusso = this.getAttribute("class");
var spessore = $(this).css("stroke-width");
var testo2 = "<b>SOURCE:</b> " + flusso + " <br>" + percorso + "<br>TOTAL: " + ((parseFloat(spessore.split("px")[0]) * fattore).toFixed(3));
//console.log(flusso, spessore);
if (flusso.length > spessore.length) { var l = flusso.length; }
else { var l = spessore.length; }
$("#infoLink").css({ "width": "" + (55 + (l * 7)) + "px" });
//$("#info").animate({ top: "0" }, 1000, "linear");
$("#infoLink").css("left", event.clientX);
$("#infoLink").css("top", event.clientY);
$("#infoLink").show();
$("#infoLink p").html(testo2);
}
function pMl() {
$("path").css('stroke-opacity', 1);
$("#infoLink").hide();
$("#infoLink p").html("");
}
function riempiSpessori() {
for (var i = 0; i < linkOrdinati.length; i++) {
for (var y = 0; y < nodiOrdinati.length; y++) {
if (linkOrdinati[i][0] == nodiOrdinati[y][0]) {
for (var d = 0; d < spessori.length; d++) {
if (nodiOrdinati[y][1] == spessori[d][0]) { var indice1 = colonne.indexOf(nodiOrdinati[y][1]); }
}
}
if (linkOrdinati[i][1] == nodiOrdinati[y][0]) {
for (var dd = 0; dd < spessori.length; dd++) {
if (nodiOrdinati[y][1] == spessori[dd][0]) { var indice2 = colonne.indexOf(nodiOrdinati[y][1]); }
}
}
}
if (indice2 - indice1 == 1) { spessori[indice1][1] += linkOrdinati[i][2]; }
else {
spessori[indice1][1] += linkOrdinati[i][2];
spessori[indice2 - 1][1] += linkOrdinati[i][2];
```

```
}
}
//console.log("SPESSORI 1", spessori);
}

function riduci(e) {
if (selezionato == 1) {
$("input").css("border", "1px solid rgb(211, 211, 211)");
selezionato = 0;
sel_vect = "";
vetto = "";
}
e.target.setAttribute("display", "none");
e.target.setAttribute("opacity", 0);
var esp=document.getElementById(e.target.getAttribute("id")+"_espandi");
esp.setAttribute("opacity", 1);
esp.setAttribute("display", "block");
var xColonna = parseInt(e.target.getAttribute("cx"));
xcol = parseInt(e.target.getAttribute("cx"));
var rect = document.getElementsByClassName("rect");
var testi = document.getElementsByClassName("txt");
for (var j = 0; j < rect.length; j++) {
if (rect[j].getAttribute("x") == xColonna) {
rect[j].setAttribute("y", parseInt(e.target.getAttribute("cy")));
rect[j].setAttribute("opacity", 0.1);
rect[j].removeEventListener('mouseover', info, false);
rect[j].removeEventListener('mouseout', close, false);
}
//console.log("TESTO X " + testi[j].getAttribute("x") +" COLONNA X "+ xColonna);
if (testi[j].getAttribute("x") == xColonna) {
//console.log("QUI!!");
//testi[j].setAttribute("y", parseInt(e.target.getAttribute("cy")) + 20);
testi[j].setAttribute("opacity", 0);
}
}
eliminaLink();
lineeIn = [];
lineeOut = [];
trovato = 0;
shiftOut = 0;
shiftIn = 0;
creaLinks();
}

function espandi(e) {
if (selezionato == 1) {
$("input").css("border", "1px solid rgb(211, 211, 211)");
selezionato = 0;
}
e.target.setAttribute("display", "none");
e.target.setAttribute("opacity", 0);
lineeIn = [];
lineeOut = [];
trovato = 0;
shiftOut = 0;
shiftIn = 0;
var sh = 0;
var shT = 0;
var esp = document.getElementById(e.target.getAttribute("id").split("_")[0]); //id pulsande riduci
esp.setAttribute("opacity", 1);
esp.setAttribute("display", "block");
var xColonna = parseInt(e.target.getAttribute("cx"));
```

```
eliminaLink();
var testi = document.getElementsByClassName("txt");
var rect = document.getElementsByClassName("rect");
var pos = 0;
var posT = 0;
var primo = 0;
var quota;
for (var j = 0; j < rect.length; j++) {
if (e.target.getAttribute("id") == "gross consumption_espandi") { console.log("SONO QUI"); var quota = 790; }
else { var quota = 290; }
if (rect[j].getAttribute("x") == xColonna) {
rect[j].setAttribute("y", ((parseInt(e.target.getAttribute("cy")) + quota) * (pos + 1)) + sh );
rect[j].setAttribute("opacity", 0.9);
rect[j].addEventListener('mouseover', info, false);
rect[j].addEventListener('mouseout', close, false);
pos = pos + 1;
//sh = sh + parseInt(rect[j].getAttribute("height"));
sh = 0;
}
if (testi[j].getAttribute("x") == xColonna) {
testi[j].setAttribute("y", ((parseInt(e.target.getAttribute("cy")) + quota) * (posT + 1)) - 5 + shT);
for (var w = 0; w < rect.length; w++) {
testi[j].setAttribute("opacity", 1);
if (rect[w].getAttribute("height") == 0 && rect[w].getAttribute("id").toUpperCase() == testi[j].innerHTML.toUpperCase()) {
testi[j].setAttribute("opacity", 0);
break;
}
}
posT = posT + 1;
//shT = shT + parseInt(rect[j].getAttribute("height"));
shT = 0;
}
}
creaLinks();
//console.log(tmp);
}
function eliminaLink() {
for (var v = 0; v < colonne.length; v++) {
if (v == 1 || v == 3 || v == 5 || v == 7) {
spessori.push([colonne[v], (100 + widthB)]);
} else {
spessori.push([colonne[v], (100 + width)]);
}
tempspesOut.push([colonne[v], 0]);
tempspesIn.push([colonne[v], 0]);
}
var links = document.getElementsByTagName("path");
console.log("elimino links ");
for (var i = links.length - 1; i > -1; i--) {
//console.log("i "+i);
//console.log("elimino", links[i]);
$(links[i]).remove();
}
}
/*$("#slider").slider({
min: 1,
max: 100,
value: 100,
animate: true,
```

```
slide: function (event, ui) {
var svg = document.getElementsByTagName("g")[0];
for (var e = 0; e < matrix.length; e++) {
matrix[e] *= (ui.value / 100);
}
matrix[4] += (1 - (ui.value / 100)) * window.innerWidth / 2;
matrix[5] += (1 - (ui.value / 100)) * (window.innerHeight-190) / 2;
console.log(matrix);
var matrix2 = "matrix(" + matrix.join(' ') + ")";
svg.setAttributeNS(null, "transform", matrix2);
}
});*/
function zoom(scale) {
console.log("SCALA");
var svg = document.getElementsByTagName("g")[0];
for (var e = 0; e < matrix.length; e++) {
matrix[e] *= scale;
}
matrix[4] += (1 - scale) * window.innerWidth / 2;
matrix[5] += (1 - scale) * (window.innerHeight - 190) / 2;
console.log(matrix);
var matrix2 = "matrix(" + matrix.join(' ') + ")";
svg.setAttributeNS(null, "transform", matrix2);
}
document.getElementById("in").addEventListener('click', function () { zoom(1.25); }, false);
document.getElementById("out").addEventListener('click', function () { zoom(0.8); }, false);
var premuto = false;
$("#up").mousedown(function () {
premuto = true;
var svg = document.getElementsByTagName("g")[0];
setInterval(function () {
if (premuto) {
matrix[5] += 10;
svg.setAttribute("transform", "matrix(" + matrix.join(' ') + ")");
}
}, 100);
});
$("#up").mouseup(function () {
premuto = false;
});
var premuto1 = false;
$("#down").mousedown(function () {
premuto1 = true;
var svg = document.getElementsByTagName("g")[0];
setInterval(function () {
if (premuto1) {
matrix[5] -= 10;
svg.setAttribute("transform", "matrix(" + matrix.join(' ') + ")");
}
}, 100);
})
.mouseup(function () {
premuto1 = false;
});
var premuto2 = false;
$("#right").mousedown(function () {
premuto2 = true;
var svg = document.getElementsByTagName("g")[0];
setInterval(function () {
```

```
if (premuto2) {
matrix[4] -= 10;
svg.setAttribute("transform", "matrix(" + matrix.join(' ') + ")");
}
}, 100);
});
$("#right").mouseup(function () {
premuto2 = false;
});
var premuto3 = false;
$("#left").mousedown(function () {
premuto3 = true;
var svg = document.getElementsByTagName("g")[0];
setInterval(function () {
if (premuto3) {
matrix[4] += 10;
svg.setAttribute("transform", "matrix(" + matrix.join(' ') + ")");
}
}, 100);
});
$("#left").mouseup(function () {
premuto3 = false;
});
var nodo;
var tmpColor;
function info(e) {
linkNodo(e);
nodo = e.target.getAttribute("id");
tmpColor = e.target.getAttribute("stroke");
e.target.setAttribute("stroke", "yellow");
e.target.setAttribute("stroke-width", 3);
var blocchi=document.getElementsByTagName("rect");
for (var j = 0; j < blocchi.length; j++){
blocchi[j].removeEventListener('click', info, false);
}
var ingressi = [];
var uscite = [];
var ext = "";
var ins = "";
var tot="";
for (var j = 0; j < altezze2.length; j++) {
if (altezze2[j][0] == nodo) {
tot=altezze2[j][1];
}
}
var t = "<b>SECTION:</b> " + nodo + "<br>TOTAL: " + tot.toFixed(3);
var entra = 0;
var esce = 0;
for (var i = 0; i < linkOrdinati.length; i++) {
if (linkOrdinati[i][0] == nodo) {
uscite.push("<br>" + linkOrdinati[i][1] + " valore: " + (linkOrdinati[i][2] * fattore).toFixed(3));
esce += linkOrdinati[i][2] * fattore;
//ext = "<br><b>PRODUCTION:</b> " + uscite + " ";
ext += "<tr><td>" + linkOrdinati[i][3] + "</td><td>" + (linkOrdinati[i][2] * fattore).toFixed(3) + "</td><td>" + linkOrdinati[i][1] + "</td></tr>";
}
if (linkOrdinati[i][1] == nodo) {
ingressi.push("<br>" + linkOrdinati[i][0] + " valore: " + (linkOrdinati[i][2] * fattore).toFixed(3));
entra += linkOrdinati[i][2] * fattore;
//ins = "<br><b>CONSUMPTION:</b> " + ingressi + " ";
ins += "<tr><td>" + linkOrdinati[i][3] + "</td><td>" + (linkOrdinati[i][2] * fattore).toFixed(3) + "</td><td>" + linkOrdinati[i][0] + "</td></tr>";
```

```
}
}
var testo = "<table class='tabella'><tr>CONSUMPTION: " + entra.toFixed(3) + "</tr> <tr> <th>SOURCE</th> <th>AMOUNT</th> <th>FROM</th></tr>";
var testo2 = "<table class='tabella'><tr>PRODUCTION: " + esce.toFixed(3) + "</tr> <tr> <th>SOURCE</th> <th>AMOUNT</th> <th>TO</th></tr>";
//$("#info").css({ "height": "" + (200 + (17 * (ingressi.length + uscite.length))) + "px" });
//$("#info").animate({ top: "0" }, 1000, "linear");
$("#info").show();
if (entra != 0 && esce != 0) {
var loss = ((esce / entra) * 100).toFixed(3) + "%";
$(".positivo").html(Math.round((esce / entra) * 100)+ "%");
$(".negativo").html(Math.round(100 - ((esce / entra) * 100)) + "%");
//torta();
document.getElementById("efficency").setAttribute("height", "" + Math.round((esce / entra) * 270) + "px");
document.getElementById("efficency").setAttribute("y", "" + 270-Math.round((esce / entra) * 270)+ "");
document.getElementById("losses").setAttribute("height", "" + Math.round(270-((esce / entra) * 270)) + "px");
document.getElementById("losses").setAttribute("y", "" + 270-Math.round(270-((esce / entra) * 270)) + "");
//$("#info").css({ "height": "" + (280 + (17 * (ingressi.length + uscite.length))) + "px" });
$("#info p").html(t + "<br><br>" + testo + ins + "</table><br>" + testo2 + ext + "</table>");
$("#holder").show();
$("#info").css({"width":"675px"});
}
if (entra == 0) {
$("#info p").html(t + "<br><br>" + testo2 + ext + "</table><br>");
$("#holder").hide();
$("#info").css({ "width": "300px" });
}
if (esce == 0) {
$("#info p").html(t + "<br><br>" + testo + ins + "</table><br>");
$("#holder").hide();
$("#info").css({ "width": "300px" });
}
}
function infoLink(e) {
var flusso;
flusso = e.target.getAttribute("class");
var spessore = e.target.getAttribute("stroke-width");
var testo2 = "<b>SOURCE:</b> " + flusso + "<br>TOTAL: " + spessore;
$("#info").css({ "height": "100px" });
//$("#info").animate({ top: "0" }, 1000, "linear");
$("#info").show();
$("#info p").html(testo2);
}
function close() {
selezionato = 0;
eliminaLink();
lineeIn = [];
lineeOut = [];
trovato = 0;
shiftOut = 0;
shiftIn = 0;
riempiSpessori();
creaLinks();
var offset = parseInt($("#info").css("height").split("px")[0]);
//$("#info").animate({ top: "-" + (offset + 30) + "px" }, 1000, "linear");
$("#info").hide();
//$("#holder svg").remove();
$("#info p").html("");
document.getElementById(nodo).setAttribute("stroke", tmpColor);
document.getElementById(nodo).setAttribute("stroke-width", 1);
var blocchi = document.getElementsByClassName("nodorect");
```

```
for (var j = 0; j < blocchi.length; j++) {
blocchi[j].addEventListener('click', info, false);
//blocchi[j].addEventListener("mouseover", info, false);
//blocchi[j].addEventListener("mouseout", close, false);
}
}
function change() {
var val = document.getElementById("numF");
if (val.options[val.selectedIndex].value == "g1") {
tipo = 0;
} else if (val2.options[val2.selectedIndex].value == "g2") {
tipo = 1;
}
var val = document.getElementById("Select1");
if (val.options[val.selectedIndex].value == "SolarRad") { valore = "SolarRad"; etichetta = "SolarRad (W/m²)"; }
if (val.options[val.selectedIndex].value == "TempOut") { valore = "TempOut"; etichetta = "Temperature (ºC)"; }
if (val.options[val.selectedIndex].value == "HumOut") { valore = "HumOut"; etichetta = "Humidity (%)"; }
if (val.options[val.selectedIndex].value == "Pressure") { valore = "Pressure"; etichetta = "Pressure (mbar)"; }
if (val.options[val.selectedIndex].value == "WindDir") { valore = "WindDir"; etichetta = "WindDir (gradi)"; }
if (val.options[val.selectedIndex].value == "WindSpeed") { valore = "WindSpeed"; etichetta = "WindSpeed (mph)"; }
if (val.options[val.selectedIndex].value == "UV") { valore = "UV"; etichetta = "UV (MEDs)"; }
if (val.options[val.selectedIndex].value == "RainRate") { valore = "RainRate"; etichetta = "RainRate (in/hr)"; }
}

});
</script>
<body >
<div id="bottoni">
<b>SOURCE</b>
<select id="numV">
</select>
<input id="vai" class="myButton" type="button" value="OK"/>
</div>
<div id="bottoniF">
<b>FAMILY</b>
<select id="numF">
</select>
<input id="invio" class="myButton" type="button" value="OK"/>
</div>
<div id="controller">
<div id="up" class="control"></div>
<div id="left" class="control"></div>
<div id="sliderCont">
<div id="in" ></div>
<div id="out"></div>
</div>
<div id="right" class="control"></div>
<div id="down" class="control"></div>
</div>
<svg>
<defs>
<pattern id="icona_meno" x="0" y="0" width="20" height="20">
<image xlink:href="icona_meno.jpg" x="0" y="0" width="20" height="20"/>
</pattern>
<pattern id="icona_piu" x="0" y="0" width="20" height="20">
<image xlink:href="icona_piu.jpg" x="0" y="0" width="20" height="20"/>
</pattern>
<pattern id="icona_casa" x="0" y="0" width="50" height="50" >
<rect width="50" height="100" style="fill:white;stroke-width:1;stroke:rgb(0,0,0)" />
<image xlink:href="casa.jpg" x="0" y="0" width="50" height="50"/>
```

```
</pattern>
<filter id="f1" x="0" y="0" width="200%" height="200%">
<feOffset result="offOut" in="SourceAlpha" dx="3" dy="3" />
<feGaussianBlur result="blurOut" in="offOut" stdDeviation="2" />
<feBlend in="SourceGraphic" in2="blurOut" mode="normal" />
</filter>
</defs>
<g ></g>
</svg>
<div id="info">
<div id="1" style="float:left">
<button id="close" class="myButton"><b>X</b></button>
<p></p>
<a href="spreadjs.html">edit data in this section</a>
</div>
<div id="holder">
<b>LOSS PERCENTAGE</b>
<svg width="200px" height="300px">
<rect id="efficency" width="40px" x="45" fill="#99FF99" filter="url(#f1)" rx="5" ry="5"></rect>
<text x="35" y="270" style="text-anchor: end; writing-mode: tb; glyph-orientation-vertical: -90;">EFFICENCY</text>
<text class="positivo" x="55" y="290"></text>
<rect id="losses" width="40px" x="130" fill="#FF6666" filter="url(#f1)" rx="5" ry="5"></rect>
<text x="120" y="270" style="text-anchor: end; writing-mode: tb; glyph-orientation-vertical: -90;">LOSSES</text>
<text class="negativo" x="140" y="290"></text>
</svg>
</div>
</div>
<div id="infoFC">
<p></p>
</div>
<div id="infoLink">
<p></p>
</div>
<div id="infoRect">
<img src="" width="40px" height="40px"/>
<p></p>
</div>
</body>
</html>
```

# Appendice – Curriculum scientifico del gruppo di lavoro

**Alberto Poggio**
Nato a Torino il 18 luglio 1971. Laureato in Ingegneria Elettrica, Dottore di Ricerca in Energetica e Master in Ingegneria della Sicurezza e Analisi dei Rischi, presso il Politecnico di Torino.
Ricercatore in Sistemi per l'Energia e l'Ambiente presso il Dipartimento Energia del Politecnico di Torino, dove insegna Fondamenti di Macchine e Impianti di Cogenerazione nel Corso di Laurea in Ingegneria Energetica. Responsabile di progetti e contratti di ricerca finanziati da aziende, enti locali e gestori di servizi energetici.
I suoi principali campi di ricerca sono l'utilizzazione energetica di biomasse e dei rifiuti, gli impianti di cogenerazione e i sistemi di teleriscaldamento, la pianificazione e la statistica energetica locale.
Assistente del Responsabile per l'Uso Razionale dell'Energia del Politecnico di Torino, per la gestione delle forniture energetiche.
Esperto della Fondazione Ambiente "Teobaldo Fenoglio" (Torino).

**Giulio Cerino Abdin**
Nato a Borgosesia (VC) il 30 settembre 1985. Laureato in Ingegneria Energetica e Nucleare presso il Politecnico di Torino.
Assegnista di ricerca in Sistemi per l'Energia e l'Ambiente presso il Dipartimento Energia del Politecnico di Torino, dove frequenta il secondo anno del Dottorato di Ricerca in Energetica sul tema "Local Planning of biomass-to-energy systems" e collabora in progetti e contratti di ricerca finanziati da enti locali e società di servizi energetici.
I suoi campi di ricerca sono relativi alla pianificazione e all'utilizzazione energetica delle biomasse, agli impianti di cogenerazione e ai sistemi di teleriscaldamento.

**Andrea Crocetta**
Nato a Pinerolo (To) il 12 settembre 1978. Laureato in Scienze Biologiche, indirizzo ecologico, presso l'Università degli Studi di Torino. Master Europeo in Tecniche per la progettazione e la valutazione ambientale, presso il Politecnico di Torino.
Assegnista di ricerca in Sistemi per l'Energia e l'Ambiente presso il Dipartimento Energia del Politecnico di Torino, dove collabora in progetti e contratti di ricerca finanziati da enti locali e società di servizi energetici.
Si occupa di utilizzazione energetica e sostenibilità ambientale delle biomasse, pianificazione energetica.

**Luca Degiorgis**
Nato a Torino l' 8 luglio 1973. Laureato in Ingegneria Meccanica, Dottore di Ricerca in Energetica, presso il Politecnico di Torino.
Assegnista di ricerca in Sistemi per l'Energia e l'Ambiente presso il Dipartimento Energia del Politecnico di Torino, dove collabora in progetti e contratti di ricerca finanziati da enti locali e società di servizi energetici.
Si occupa di solare termico, utilizzazione energetica delle biomasse. Si è occupato di produzione idrogeno.
Collaboratore del Responsabile per l'Uso Razionale dell'Energia del Politecnico di Torino, per gli interventi di risparmio energetico e monitoraggio dei consumi.
Già collaboratore presso l'Istituto per le Piante da Legno e l'Ambiente (Torino), nell'Unità di Ricerca Energie Rinnovabili, dove si è occupato di progettazione di filiere energetiche forestali.

**Michel Noussan**
Nato ad Aosta il 22 marzo 1985. Laureato in Ingegneria Energetica e Nucleare presso il Politecnico di Torino. Assegnista di ricerca in Sistemi per l'Energia e l'Ambiente presso il Dipartimento Energia del Politecnico di Torino, dove frequenta il secondo anno del Dottorato di Ricerca in Energetica sul tema "Simulation models for biomass-fired power plants" e collabora in progetti e contratti di ricerca finanziati da enti locali e società di servizi energetici.
Il suo campo di ricerca è relativo alla simulazione degli impianti produzione energetica da biomasse solide e dei sistemi di teleriscaldamento, alla statistica energetica locale.

**Maurizio Fantino**
Nato a Cuneo nel 1976. Ha conseguito la laurea in Ingegneria delle Telecomunicazioni (110/110) nel 2001 presso il Politecnico di Torino con la tesi dal titolo: "Studio e Simulazione di Turbo Codici per la trasmissione di segnali audio e video con il sistema UMTS" svolta al centro ricerche e innovazione tecnologica della RAI.
Maurizio Fantino ha anche conseguito un dottorato di ricerca in ingegneria Elettronica e delle Comunicazioni, sempre presso il Politecnico di Torino nell'ambito della scuola di alta eccellenza interpolitecnica nel 2006. Il programma di dottorato è stato supportato da LABEN, una delle maggiori aziende italiane operanti nell'ambito della navigazione satellitare.
Tra il dicembre 2001 e il marzo 2003 ha lavorato al centro ricerche TILAB (Telecom Italia LAB) nel gruppo "rete di accesso" occupandosi di architetture di rete.
A partire dal Gennaio 2006 è un ricercatore dell'Istituto Superiore Mario Boella. Membro del Navigation and Simulation Group (NavSAS) fino all'inizio del 2012 ha lavorato attivamente nel campo della navigazione satellitare come responsabile della unità "GNSS Core Algorithms".
Oggi è responsabile del programma strategico "Smart Energy", un programma che opera trasversalmente rispetto alle aree di ricerca dell'Istituto e volto a esplorare le potenzialità delle Information and Communication Technologies a supporto di un impiego più efficiente ed efficace delle risorse nostre risorse energetiche.

**Giuseppe Caragnano**
Si laurea in ingegneria delle Telecomunicazioni nel ottobre del 2006. Nel 2008 lavora presso il laboratorio di ricerca bancaria applicata presso ISMB. Ha maturato competenze su sistemi Linux, database SQL e diversi linguaggi di programmazione. Dal 2010 lavora presso l'unità IS4AC e si occupa di infrastrutture di calcolo per l'elaborazione di grandi quantità di dati, in particolare studiando soluzioni basate su grid e cloud computing e su cluster di database, con accento su schedulazione dei processi e gestione delle risorse di calcolo. Egli ha maturato competenze su tecnologie di Cloud Computing in particolare sulla progettazione di soluzioni di private cloud e sviluppo di software su piattaforma Azure di Microsoft.

**Mario Chiesa**
Si è laureato in Ingegneria Elettronica presso il Politecnico di Torino ed ha conseguito il Master in Interaction Design presso l'Interaction Design Institute di Ivrea. Dal 2003 lavora presso l'Istituto Superiore Mario Boella (ISMB). Dal Luglio 2013 è responsabile dell'Unità di Ricerca "Creativity and Design Innovation" (CDI), attiva negli ambiti della Creatività, dell'Interaction Design e della User Experience. Come ricercatore e designer, è impegnato nella definizione e implementazione di strategie e soluzioni ICT per l'innovazione di prodotti, servizi e processi, attraverso la progettazione dell'interazione e dell'esperienza dell'utente, dell'architettura dell'informazione, la gestione della conoscenza, il multimedia, la visualizzazione di dati. I domini di applicazione sono svariati: dall'IoT - Internet delle cose, alle città intelligenti, dai musei virtuali all'automotive, alla comunicazione. Si occupa inoltre delle proposte per nuovi progetti di ricerca regionali ed europei, della gestione di progetti e dei loro aspetti organizzativi e gestionali.

**Francesco Posca**
Ha ottenuto la laurea specialistica nel 2011 in Ingegneria del Cinema e dei Mezzi di Comunicazione presso il Politecnico di torino. Frequentando corsi quali “Ambienti Multimediali” e “Realtà Virtuale” acquisisce competenze inerenti la programmazione web e la grafica. Nel 2011 inizia a collaborare come freelance per progetti di grafica, dall'animazione motion graphics 2D, alla modellazione di oggetti 3D, e come web designer, presso PMI del territorio. Da Giugno 2013 lavora per la Creativity and Design Innovation Unit, all'interno dell'Area di Ricerca Innovation Development dell’Istituto Superiore Mario Boella, occupandosi di modellazione 3D, visualizzazione di dati e interfacce utente per il web.

**Pietro Ruiu**
Ha ottenuto la laurea magistrale in Ingegneria delle Telecomunicazioni al Politecnico di Torino nel 2006. Dal 2013 ricopre il ruolo di responsabile dell’Unità di Ricerca IS4AC (Infrastructures and Systems for Advanced Computing) e si occupa del coordinamento di attività di ricerca sulle infrastrutture di calcolo per grandi quantità di dati, che coinvolgono tecnologie come il Cloud Computing, il Grid Computing, l’High Performance Computing (HPC), la virtualizzazione. Alcune delle tematiche di interesse riguardano l’automazione del cloud, la migrazione delle applicazioni in cloud, l’interoperabilità delle piattaforme, il cloud computing per applicazioni e-science e il green computing. Ha partecipato alla stesura e alla gestione di numerosi progetti di ricerca nazionali ed europei in collaborazione con partner internazionali, acquisendo competenze sui principali strumenti di virtualizzazione e cloud computing in particolare open source.