

RELAZIONE ELABORATO SISTEMI MULTIMEDIALI

Anno 2014/2015

Eventi in Romagna

GRUPPO 7:

SAMANTHA BANDINI

ANNA GIULIA LEONI

MATTEO VITALI

Specifiche elaborato

Realizzare una trasformazione XSLT che prenda in input un file XML e restituisca una pagina HTML5+CSS3+Javascript.

Il file XML di input deve contenere i dati relativi a diversi eventi e il file HTML di output deve permettere agli utenti di navigare gli eventi, visualizzandoli tutti oppure selezionando alcuni parametri per filtrarli.

Le caratteristiche obbligatorie degli eventi sono: il titolo, la data (oppure data di inizio e data di fine dell'evento), l'ora di inizio dell'evento, il tipo di evento (ad esempio: mostra fotografica, rassegna cinematografica, rappresentazione teatrale, spettacolo, concerto, conferenza, seminario, eccetera), il luogo dell'evento (città + indirizzo esatto), una descrizione dettagliata dell'evento, il link al sito Web dell'evento (se disponibile). Queste caratteristiche dovranno essere utilizzare per creare la DTD e la relativa istanza XML con tutti gli eventi.

L'HTML di output deve permettere all'utente di modificare l'ordine di visualizzazione degli eventi (in ordine cronologico crescente oppure decrescente, raggruppati per tipo oppure per città).

Nell'HTML di output l'utente deve poter visualizzare oppure nascondere i dettagli dell'evento, deve poter visualizzare la mappa di GoogleMaps sulla base dell'indirizzo dell'evento, deve poter filtrare gli eventi sulla base di più parametri contemporaneamente (ad esempio: luogo, data e tipo).

Dovranno essere consegnati: il codice XSLT, XML + DTD e HTML5+CSS3+Javascript.

Scelte implementative

Nella realizzazione dell'elaborato siamo partiti dalla creazione dell'XML per aiutarci a capire quali sarebbero stati i tag che avrebbero composto il nostro evento e, una volta costruito un evento tipo e assegnato a questo dei valori, siamo passati alla creazione della DTD.

La prima pagina che si apre lanciando il progetto è index.html, ottenuta tramite una trasformazione xslt da index.xsl.

index.html permette di visualizzare tutti gli eventi nell'ordine in cui sono stati inseriti nell'XML. E' possibile tornare in ogni momento a questa visualizzazione degli eventi attraverso la pressione dell'header della pagina.

Per ogni evento vengono visualizzate le informazioni principali (nome, tipo, data e città in cui si svolge).

Se l'utente è interessato all'evento, può premere su "Visualizza dettagli" alla cui pressione verrà lanciato uno script javascript [manageDetails.js] che si occuperà di rendere visibili altre informazioni dell'evento per il quale ha richiesto i dettagli (orario di inizio, indirizzo, descrizione, sito web).

Verrà inizializzata ogni volta alla pressione anche la mappa di Google Maps con un map marker all'indirizzo dell'evento [maps.js].

Al contrario "Nascondi dettagli" lo farà tornare alla visualizzazione delle sole informazioni principali dell'evento.

Gli utenti hanno la possibilità di interagire modificando la visualizzazione degli eventi in due modi.

Il primo modo consiste in un semplice ordinamento o raggruppamento per data, città o tipo: in questo modo gli eventi verranno comunque mostrati tutti ma in ordine diverso rispetto a quello di inserimento.

E' possibile effettuare questi ordinamenti grazie al menu orizzontale posizionato subito sotto all'header della pagina.

La seconda modalità consiste nel filtrare gli eventi da visualizzare, quindi andando a rendere invisibili tutti gli eventi a cui l'utente non è interessato.

E' possibile applicare i filtri su province, date e tipi di evento grazie alla form presente alla destra del container degli eventi.

Queste due modalità possono essere combinate, cioè è data la possibilità di applicare filtri anche nelle pagine ordinate in modo diverso da quello di default.

DTD e XML

```
<!ELEMENT event-list (event)+>
<!ELEMENT event (title|period|type|location|description|webSite?|image)+>
<!ATTLIST event
  id CDATA #REQUIRED
  >
<!ELEMENT title (#PCDATA)>
<!ELEMENT period (start|end)+>
<!ELEMENT start (date|time)+>
<!ELEMENT date (year|month|day)+>
<!ELEMENT year (#PCDATA)>
<!ELEMENT month (#PCDATA)>
<!ELEMENT day (#PCDATA)>
<!ELEMENT time (hour|minute)+>
<!ELEMENT hour (#PCDATA)>
<!ELEMENT minute (#PCDATA)>
<!ELEMENT end (date)+>
<!ELEMENT type (#PCDATA)>
<!ELEMENT location (address|number|city|district|map)+>
<!ELEMENT address (#PCDATA)>
<!ELEMENT number (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT district (#PCDATA)>
<!ELEMENT map (lat|long)+>
<!ELEMENT lat (#PCDATA)>
<!ELEMENT long (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT webSite (#PCDATA)>
<!ELEMENT image (#PCDATA)>

<!ENTITY CC "Concerto">
<!ENTITY CF "Conferenza">
<!ENTITY MO "Mostra">
<!ENTITY FI "Fiera">
<!ENTITY SA "Sagra">
<!ENTITY SP "Spettacolo">

<!ENTITY RA "RA">
<!ENTITY FC "FC">
<!ENTITY RN "RN">
```

```
<!DOCTYPE event-list SYSTEM "events.dtd">
<event-list>

  <event id="1">
    <title>Linea 77 in concerto</title>
    <period>
      <start>
        <date>
          <year>2015</year>
          <month>07</month>
          <day>31</day>
        </date>
        <time>
          <hour>21</hour>
          <minute>00</minute>
        </time>
      </start>
      <end>
        <date>
          <year>2015</year>
          <month>07</month>
          <day>31</day>
        </date>
      </end>
    </period>
    <type>&CC;</type>
    <location>
      <address>Viale Europa</address>
      <number>1</number>
      <city>Ravenna</city>
      <district>&RA;</district>
      <map>
        <lat>44.421875</lat>
        <long>12.229149</long>
      </map>
    </location>
    <description>Descrizione </description>
    <webSite>http://www.linea77.com/</webSite>
    <image>http://www.linea77.com/wp-content/uploads/2014/02/Linea77-940x470.jpg</image>
  </event>

  Altri eventi...

</event-list>
```

Come è possibile notare l'XML è conforme alla DTD.

Abbiamo deciso di inserire nella DTD, oltre agli elementi richiesti dalle specifiche dell'elaborato, altri elementi come latitudine, longitudine e un'immagine per l'evento. Le coordinate geografiche dell'indirizzo dell'evento ci servono per posizionare il map marker della mappa di Google Maps. Abbiamo utilizzato anche delle entità per quanto riguarda i tipi e le province per evitare errori di battitura nell'XML che causerebbero inconsistenza.

Realizzazione dell'ordinamento per data e dei raggruppamenti.

Subito sotto l'header delle pagine html è presente una nav che permette agli utenti del sito di visualizzare gli eventi ordinandoli per data decrescente o crescente, per città oppure per tipo.

La pressione su ognuno di questi link rimanda rispettivamente a 4 pagine html realizzate attraverso 4 trasformazioni xsl (eventsDateAsc.html, eventsDateDesc.html, groupByCity.html e groupByType.html).

Per fare questo abbiamo semplicemente aggiunto al codice dell'index.xsl un sort sulle date, per ordine crescente, poi decrescente (per quanto riguarda i primi due html), un controllo sulla provincia in cui si svolge l'evento (per il terzo html) e un controllo sul tipo dell'evento (per il quarto html).

Alla prima pressione di "Ordina per data" dall'index.html gli eventi vengono ordinati in maniera decrescente, alla seconda pressione in maniera crescente, poi decrescente di nuovo e così via.

Alla pressione di "Raggruppa per città" tutti gli eventi che si svolgono all'interno della stessa provincia (ad esempio: Ravenna) vengono raggruppati uno sotto l'altro e viene visualizzata la provincia a cui fanno riferimento.

Alla pressione di "Raggruppa per tipo" tutti gli eventi che hanno lo stesso tipo (ad esempio: Concerto, Mostra, Spettacolo...) vengono raggruppati uno sotto l'altro e viene visualizzato il tipo a cui fanno riferimento.

Realizzazione dei filtri.

Sulla destra delle pagine html invece è data la possibilità agli utenti del sito di filtrare gli eventi sulla base di più parametri contemporaneamente (provincia, data, tipo) anche appartenenti allo stesso tipo (più province, più tipi di evento). Ad esempio l'utente può richiedere di visualizzare i concerti e le mostre che si svolgeranno in provincia di Ravenna e Rimini in un certo range di tempo.

I filtri possono essere applicati da ogni pagina html del sito e sono stati inseriti tutti all'interno di una form:

- `<input type="checkbox"/>` per quanto riguarda le province e i tipi di evento
- `<input type="date"/>` per quanto riguarda le date

Gli unici browser che supportano una visualizzazione corretta dell'`<input type="date"/>` sono Google Chrome e Opera.

La discriminazione sulla data può essere fatta scegliendo una data singola o un intervallo di date. Se l'utente sceglie una singola data dovrà selezionare la stessa data sia nell'input relativo a "Dal" sia in quello relativo ad "Al". Nel caso in cui un utente volesse vedere tutti gli eventi presenti entro una certa data oppure tutti gli eventi successivi ad una data gli basterà inserire rispettivamente la data di fine (Al) oppure la data di inizio (Dal).

Alla pressione del pulsante di submit "Applica filtri" viene lanciato uno script javascript [filter.js] che funziona a grandi linee in questo modo:

- I. Vengono recuperati tutti i filtri inseriti dall'utente
- II. Gli eventi vengono discriminati prima sulla base della provincia
- III. Il risultato della discriminazione sulla provincia è l'input per la discriminazione sulla base del tipo di evento
- IV. Il risultato della discriminazione sul tipo di evento è l'input per la discriminazione sulla base della data
- V. Il risultato della discriminazione sulla data sono gli eventi da far visualizzare all'utente.

Possibili ottimizzazioni

Rendere il filtro per la data accessibile da tutti i browser

Come detto prima, l'input di tipo "date" è visibile solamente da Google Chrome e Opera.

Abbiamo scelto di utilizzarlo ugualmente perché è molto semplice, intuitivo e veloce per l'utente: fornisce infatti un calendario attraverso cui è possibile scegliere la data e in più controlla che il formato della data sia corretto e, nel caso in cui venga inserita una data non completa (ad esempio l'utente inserisce solo il giorno ma non il mese e nemmeno l'anno) questa viene inviata alla form come vuota, per cui dà la possibilità di risparmiare molti controlli.

Per rendere possibile la visualizzazione da altri browser basterebbe utilizzare un altro input, ad esempio un campo di testo, e poi controllarne la formattazione con delle regular expression e controllare che la data inserita sia una data valida (cose che l'input "date" fa in automatico).

Ottimizzazione per mobile

Per quanto riguarda i dispositivi mobile si potrebbe ottimizzare questo sito per la visualizzazione da mobile, realizzando una web app, oppure si potrebbe pensare di sviluppare un'applicazione vera e propria.

Nel secondo caso ci si troverebbe a dover decidere se sviluppare applicazioni dedicate (app native per ogni sistema operativo) oppure se realizzare un'app ibrida utilizzando dei framework come Cordova che ci aiutano a sviluppare applicazioni multiplatforma con supporto nativo.

Nel caso in cui si scegliesse di ottimizzare il sito per visualizzazione da mobile, sarebbe necessario impostare il layout in modo che sia portatile per tutti i diversi dispositivi.

Alcune ottimizzazioni potrebbero essere:

- Utilizzare percentuali invece di dimensioni fisse, in modo che la visualizzazione si adatti allo schermo del dispositivo.
- Nel caso in cui la visualizzazione venga effettuata da mobile si potrebbero mostrare solamente le immagini degli eventi e visualizzare le informazioni solamente alla pressione sull'immagine.