



tesi di laurea

## **Realizzazione in Flex di un'applicazione web per l'accesso a libri di testo in formato digitale**

Anno Accademico 2010/2011

**relatore**

Ch.mo prof. Porfirio Tramontana

**candidato**

Adriano D'Onofrio

Matr. 534003277

## Obiettivi:

- creare un'applicazione che permetta di ricercare e visualizzare una serie di testi scolastici digitalizzati, organizzati per materie, con l'ausilio delle nuove tecnologie del web 2.0 e precisamente attraverso l'utilizzo del framework Flex

## Analisi dei requisiti:

- ricerca testi attraverso una lista che visualizzi le copertine dei libri
- ricerca testi attraverso una lista che permetta la selezione in base all'appartenenza del libro ad una specifica materia, argomento e sottoargomento
- visualizzazione dei libri in un lettore di documenti realizzato come applicazione web che metta a disposizione le principali operazioni che si effettuano durante la lettura di un documento (cambio pagina, zoom, ricerca parola ecc.)





## Tecnologie e linguaggi

### Flex

- Framework open source e gratuito per lo sviluppo di applicazioni RIA
- si presta alla creazione di programmi oltre che per il web anche per il desktop e i dispositivi mobili
- supporta il data-binding, la progettazione di un'architettura ad eventi, e una serie di componenti con metodi incorporati
- è una tecnologia ibrida comprendente i linguaggi ActionScript ed MXML

### ActionScript 3.0

- linguaggio di scripting basato su ECMAScript
- linguaggio ad alto livello e orientato agli oggetti che rende possibile la gestione dati e l'interattività
- linguaggio interpretato (o pseudocompilato)
- la sua esecuzione è affidata all'AVM (ActionScript Virtual Machine)

### MXML (Macromedia Flex Markup Language)

- linguaggio di markup dichiarativo basato su XML
- usato per predisporre l'interfaccia di un'applicazione e per definire gli aspetti non visuali come l'accesso a dati in remoto, i legami tra i dati e i comportamenti

### MySQL

- sistema di gestione per database relazionali basato sul linguaggio SQL
- si occupa della strutturazione e della gestione a basso livello dei dati, in modo da velocizzarne l'accesso, la modifica e l'inserimento di nuovi elementi

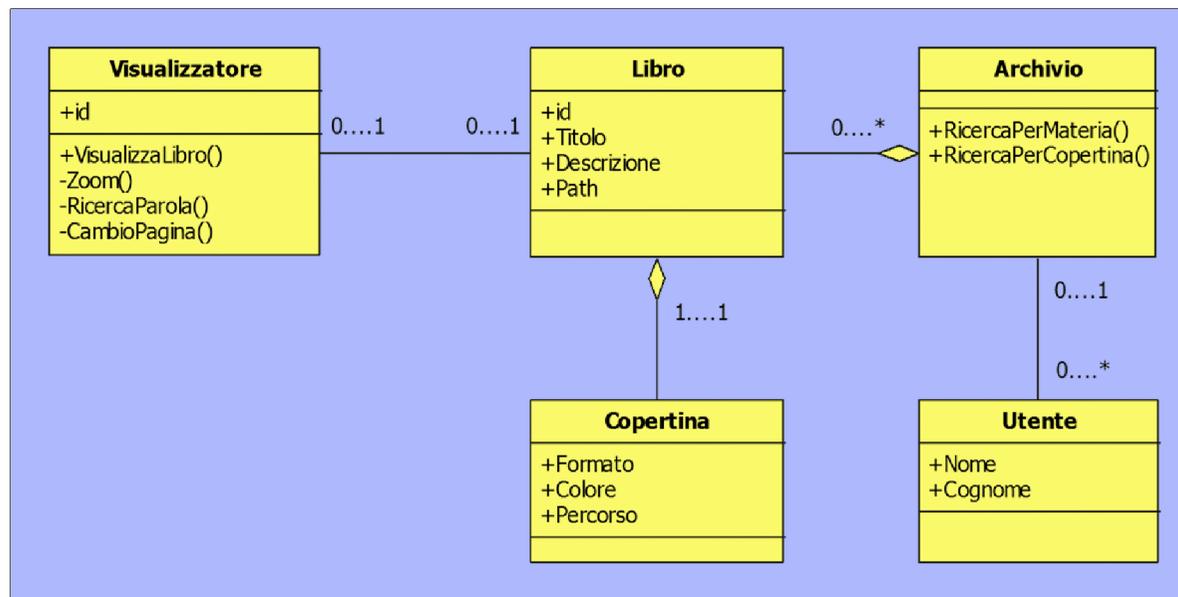
### AsSql

- driver ActionScript 3.0
- crea una connessione socket tra l'applicazione Flex dell'utente e il database nella rete



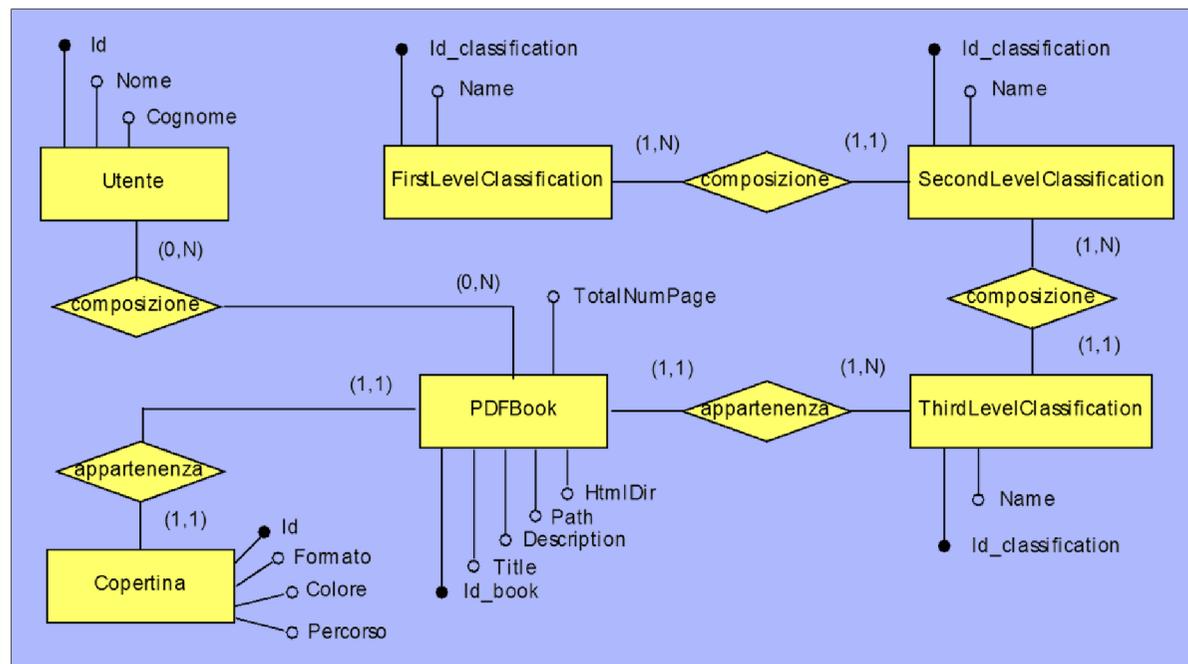
## Progettazione

- Fase in cui si decidono le modalità di passaggio da "che cosa" deve essere realizzato a "come" deve essere realizzato
- il diagramma delle classi di progetto aiuta nella comprensione di come è fatta l'applicazione sviluppata
- cinque entità e quattro relazioni individuate di cui due di associazione e due di aggregazione



## I dati e il modello concettuale

- Necessità di visualizzare una serie di informazioni quali immagini e testo
- tre livelli di classificazione previsti per i testi
- cinque entità e cinque associazioni individuate



## La struttura dell'interfaccia utente

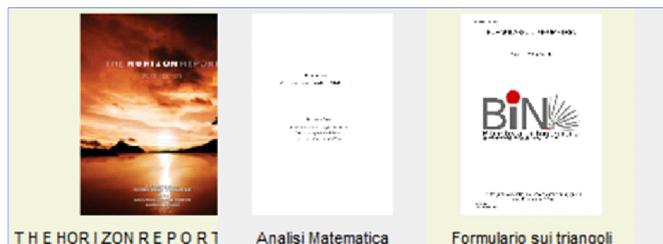
L'interazione dell'utente con la macchina è garantita da un'interfaccia costituita dai seguenti elementi:

### Intestazione

- contiene il logo dell'applicazione

### Lista orizzontale per la visualizzazione delle copertine

- mostra l'elenco delle copertine con i relativi titoli dei libri presenti sul server
- popolata attraverso dati esterni recuperabili con delle interrogazioni
- permette la visualizzazione di un libro tramite un click sulla copertina



### Lista verticale per la ricerca dei libri

- permette la ricerca di un libro in maniera progressiva
- la navigazione tra le categorie di appartenenza dei libri determina l'aggiornamento della lista orizzontale
- scelto il sottoargomento permette la visualizzazione di un libro attraverso un click sul titolo
- il bottone sottostante consente la navigazione a ritroso

### Visualizzatore documenti

- consente la visualizzazione del libro vero e proprio
- il progetto FlexPaper fornisce un componente già pronto integrabile come libreria esterna
- FlexPaper = visualizzatore web di documenti open source con pulsanti per il cambio pagina, lo zoom, la stampa, l'adattamento delle pagine e strumenti per la ricerca di parole e la selezione del testo



Realizzazione in Flex di un'applicazione web per l'accesso a libri di testo in formato digitale

The screenshot displays the ISCHOOL PORTAL interface. At the top, the logo for CRIAI and ISCHOOL PORTAL are visible. Below the header is a horizontal carousel of document thumbnails. The thumbnails include:

- provapdfmg
- provaimgs2109
- Formulario sui triangoli
- provafont
- test pdf Adobe Creator
- test pdf Foxit Creator
- test

On the left side, there is a sidebar titled "Scegli la materia" (Choose the subject) with a list of subjects: Matematica, Italiano, Geografia, Fisica, storia, Inglese, and Inglese 2. A "Ritorna" (Return) button is located at the bottom of this sidebar.

The main content area shows a digital viewer of a book page. The page title is "Il rifugio segreto" by Pino Assandri and Elena Mutti, with the subtitle "Leggere i Classici". The page is displayed at a zoom level of 57%. The viewer includes a toolbar with various navigation and utility icons, and the FlexPaper logo is visible in the bottom right corner.



## La logica dell'applicazione

### OnAppComplete()

- invocata quando l'applicazione è completamente inizializzata o indirettamente alla pressione del pulsante con etichetta ritorna
- aggiorna le due liste dell'interfaccia con le copertine di tutti libri e con l'elenco di tutte le materie presenti nel DB
- con il primo servizio di connessione al DB esegue una proiezione della relazione (tabella) `firstlevelclassification` su tutti i suoi attributi

```
service.send("SELECT * FROM firstlevelclassification")
```

- con il secondo servizio esegue una proiezione della relazione `pdfbook` rispetto agli attributi "title", "id\_book", "htmlDir" e "path"

```
service2.send("SELECT title,id_book,htmlDir,path FROM  
pdfbook")
```

### CaricaListaArgomenti()

- invocata dopo la selezione di una materia o indirettamente alla pressione del pulsante con etichetta ritorna
- la procedura fa utilizzo di due variabili di tipo stringa `CondizioneSelezioneQuery` e `SalvaSelezioneMaterie`
- `flag=true` (selezione materia) ⇒ viene assegnato alle due variabili il valore della chiave primaria della materia selezionata
- `flag=false` (pressione pulsante) ⇒ viene assegnato alla variabile `CondizioneSelezioneQuery` il valore della chiave primaria della materia che era stata precedentemente memorizzata in `SalvaSelezioneMaterie`
- il valore del parametro formale booleano "flag" contribuisce alla distinzione di due casi differenti

- dopo il costrutto di selezione vengono eseguite le due interrogazioni necessarie all'aggiornamento delle liste

```
service.send("SELECT * FROM secondlevelclassification  
where parent =" + CondizioneSelezioneQuery)
```

```
service2.send("SELECT title,id_book,htmlDir,path  
FROM secondlevelclassification,  
thirdlevelclassification,pdfbook  
where secondlevelclassification.parent=  
"+CondizioneSelezioneQuery+" AND  
secondlevelclassification.id_classification=  
thirdlevelclassification.parent"+" AND  
pdfbook.thirdlevelclassification=  
thirdlevelclassification.id_classification")
```



Realizzazione in Flex di un'applicazione web per l'accesso a libri di testo in formato digitale

## CaricaListaSottoArgomenti()

- invocata dopo la selezione di un argomento o indirettamente alla pressione del pulsante con etichetta ritorna
- presenta lo stesso principio di funzionamento della procedura prima illustrata.
- la differenza sta nel fatto che le due liste vengono aggiornate con l'elenco delle copertine dei libri e con l'elenco dei sottoargomenti relativi all'argomento selezionato
- due interrogazioni eseguite dopo il costrutto if-then-else

```
service.send("SELECT * FROM secondlevelclassification  
where parent =" + CondizioneSelezioneQuery)
```

```
service2.send("SELECT title,id_book,htmlDir,path  
FROM thirdlevelclassification,pdfbook  
where thirdlevelclassification.parent=  
"+ CondizioneSelezioneQuery+" AND  
pdfbook.thirdlevelclassification=  
thirdlevelclassification.id_classification")
```

## CaricaListaLibri()

- invocata dopo la selezione di un sottoargomento
- la procedura fa utilizzo di un'unica variabile `CondizioneSelezioneQuery` a cui assegna il valore della chiave primaria del sottoargomento selezionato
- aggiorna poi le due liste dell'interfaccia con l'elenco delle copertine dei libri e con l'elenco di tutti i titoli dei libri relativi al sottoargomento selezionato

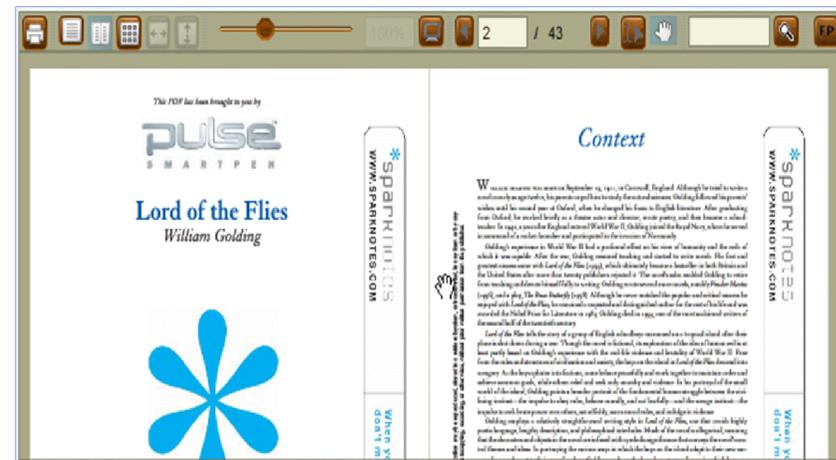
## TornaIndietro()

- invocata direttamente alla pressione del bottone con etichetta "Ritorna" è caratterizzata da una struttura condizionale nidificata
- le procedure finora analizzate oscurano la lista visibile nel loro istante di esecuzione per mostrarne un'altra aggiornata
- la funzione in esame verifica quale sia la lista visibile al momento della sua esecuzione per poi richiamare la routine più appropriata che consenta di procedere all'indietro nella selezione delle categorie di appartenenza dei libri

Realizzazione in Flex di un'applicazione web per l'accesso a libri di testo in formato digitale

## VisualizzaLibroInFlexPaper()

- invocata dopo la selezione di un libro dalla lista verticale o dopo la selezione della copertina dalla lista orizzontale
- costruisce l'url necessario ad eseguire una richiesta http per il recupero del libro di testo digitale nel formato swf
- tre metodi per la manipolazione di stringhe adoperati per la determinazione dell'url che viene assegnato al percorso sorgente del visualizzatore di documenti tramite una variabile "Bindable"





## Conclusioni e sviluppi futuri

- Il progetto si proponeva di sviluppare un applicativo di base che incentivasse gli studenti alla consultazione di materiale didattico digitale attraverso le nuove tecnologie del web
- i risultati raggiunti con lo sviluppo dell'applicazione potrebbero essere integrati con altre caratteristiche funzionali utili al miglioramento della didattica nelle scuole
- si potrebbero aggiungere funzionalità per la memorizzazione di annotazioni, funzionalità per la consultazione di materiale integrativo come slides e appunti, test per valutare le conoscenze acquisite dallo studente e anche strumenti per la visualizzazione di contenuti multimediali come video e immagini