



**Politecnico
di Torino**

Laurea Magistrale in Ingegneria Informatica

Dal testo al codice: uno strumento per incoraggiare l'analisi degli esercizi di programmazione

Relatori:

Luigi De Russis
Juan Pablo Sáenz
Fulvio Corno

Candidato:

Andrea Bruno

Obiettivi della tesi

Supporto degli studenti di informatica nell'analisi e decomposizione degli esercizi di programmazione



Individuare bisogni e abitudini degli studenti:
Analisi della letteratura e Needfinding



Progettare, implementare e validare
un'opportuna applicazione web

Analisi della letteratura

Gli studenti non analizzano i loro esercizi prima di programmare:

Hanno difficoltà

Impiegano
troppo tempo

Non ci pensano

Vogliono solo risolvere errori
di sintassi

Pensano che denoti scarse
capacità di programmazione

Needfinding



Questionario



Intervista

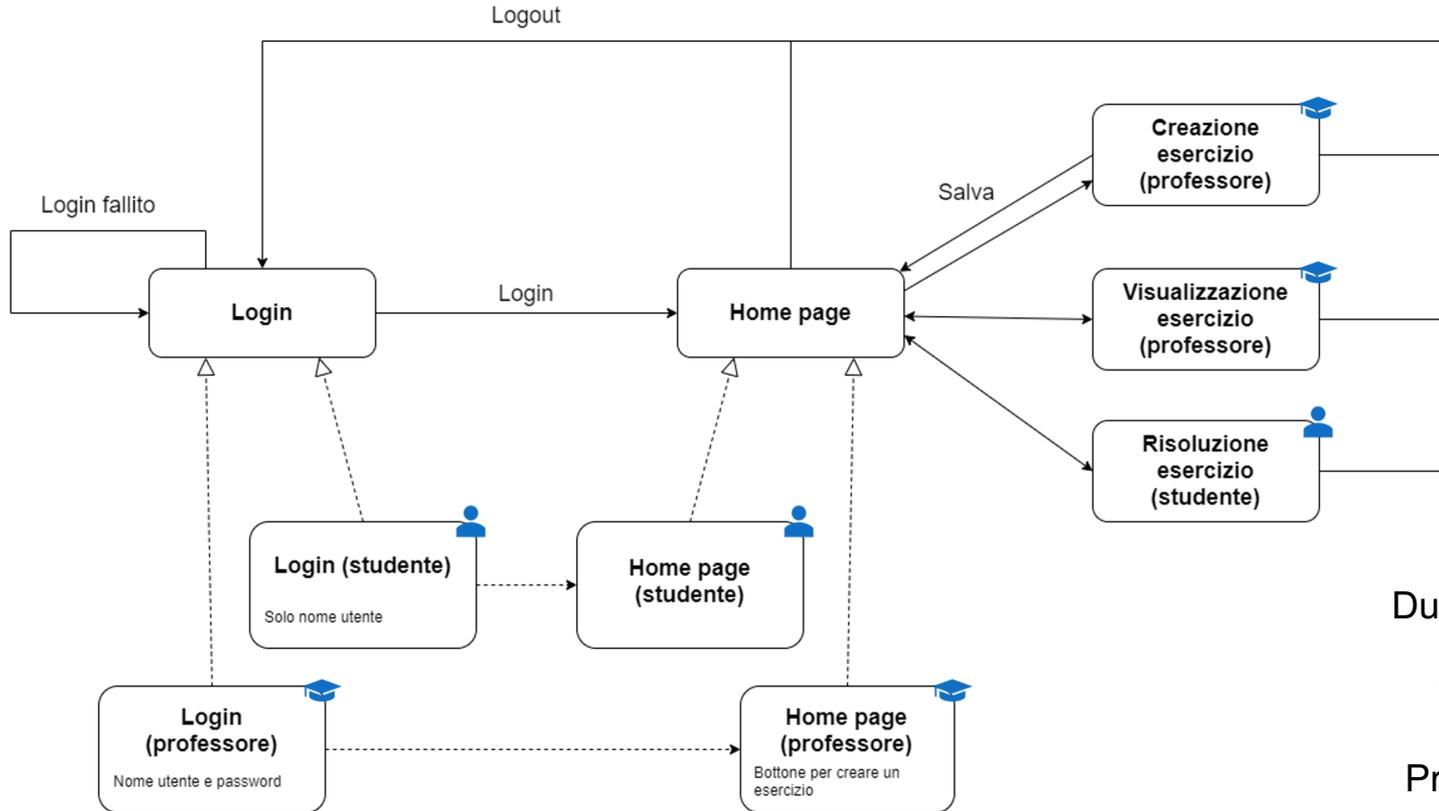


Disegno di un'interfaccia

Alcuni risultati:

- Chi non analizza il problema tende ad **alternare fasi di lettura e di implementazione**
- Chi non analizza l'esercizio tende a dover **tornare indietro a modificare il programma** quando si accorge di aver sbagliato approccio

Interazioni all'interno dell'interfaccia



Due tipi di utenti:

Studente 

Professore 

Risoluzione dell'esercizio

Annotazioni sul testo
del problema

Suddivisione del testo
in sottoproblemi

Processo di analisi
relativamente veloce

The screenshot shows the 'Text to Code' web application interface. The title is 'Calendarizzazione'. The main content area is titled 'Testo dell'esercizio' and contains the following text:

In un programma di calendarizzazione (scheduling) di eventi si deve verificare se due appuntamenti della stessa giornata si sovrappongono. Per semplicità, ipotizziamo che gli appuntamenti inizino sempre a un'ora esatta (senza minuti), ma

- controlla che usino l'**orario militare** (cioè con le ore che vanno da 0 a 23).

Lo pseudocodice seguente descrive un algoritmo che determina se l'appuntamento che inizia all'ora **start1** e termina all'ora **end1** si sovrappone all'appuntamento che inizia all'ora **start2** e termina all'ora **end2**.

- Se $start1 > start2$

```
s = start1
```

Altrimenti

```
s = start2
```

The interface also features a navigation bar with 'INIZIA A PROGRAMMARE', 'SALVA', 'FILE', 'SOLUZIONE', and 'ESEGUI' buttons. A 'CREA SEZIONE' button is located above the text editor. The text editor includes 'Bold' and 'Underline' options and a 'CREA SEZIONE' button.

Risoluzione dell'esercizio

La scrittura del codice è disponibile solo dopo aver analizzato il testo dell'esercizio

Analisi del problema prima di iniziare a scrivere il programma

Testo del problema accanto al codice

Text to Code Home page

Calendarizzazione

SALVA FILE SOLUZIONE ESEGUI ▶

Testo dell'esercizio

Bold Underline CREA SEZIONE

In un programma di calendarizzazione (scheduling) di eventi si deve verificare se due appuntamenti della stessa giornata si sovrappongono. Per semplicità, ipotizziamo che gli appuntamenti inizino sempre a un'ora esatta (senza minuti), ma

- controlla che usino l'orario militare (cioè con le ore che vanno da 0 a 23).

Lo pseudocodice seguente descrive un algoritmo che determina se l'appuntamento che inizia all'ora **start1** e termina all'ora **end1** si sovrappone all'appuntamento che inizia all'ora **start2** e termina all'ora **end2**.

- Se $start1 > start2$

```
s = start1
```

Altrimenti

```
s = start2
```

Codice provvisorio

Scegli una sezione

```
1- if start1 < start2:
2-     s = start2
3- else:
4-     s = start1
```

```
1- if end2 < end1:
2-     e = end2
3- else:
4-     e = end1
```

Codice definitivo

```
1- Scrivi qui i tuoi import
2- def isOverlap(start1, end1, start2, end2):
3-
4-
5-
6-
7-
8-
9-
10-
11-
12-
13-
14-
15-
16-
17-
18-
19-
20-
21-
22-
23-
24-
25-
26-
27-
28-
29-
30-
31-
32-
33-
34-
35-
36-
37-
38-
39-
40-
41-
42-
43-
44-
45-
46-
47-
48-
49-
50-
51-
52-
53-
54-
55-
56-
57-
58-
59-
60-
61-
62-
63-
64-
65-
66-
67-
68-
69-
70-
71-
72-
73-
74-
75-
76-
77-
78-
79-
80-
81-
82-
83-
84-
85-
86-
87-
88-
89-
90-
91-
92-
93-
94-
95-
96-
97-
98-
99-
100-
```

```
1- for hour in [start1, start2, end1, end2]:
2-     if hour < 0 or hour > 23:
3-         return "Input non in formato militare"
4-
5- if start1 > start2:
6-     s = start1
7- else:
8-     s = start2
9-
10- if end1 < end2:
11-     e = end1
12- else:
13-     e = end2
14-
15- if s < e:
16-     return "Sovrapposizione"
17- else:
18-     return "Nessuna sovrapposizione"
19-
20- print(isOverlap(10, 12, 11, 13))
```

Funzionalità aggiuntive

Supporto nella scelta di algoritmi e strutture dati

Gli appuntamenti 10-12 e 11-13 si sovrappongono?

! Domanda obbligatoria

Sì

No

Soluzione corretta!

RISPONDI

Commento del professore

La sovrapposizione avviene nell'intervallo 11-12

6 10-12 e 11-13

7 10-11 e 12-13

Domande del professore

Supporto da parte del professore

Soluzione proposta

In un programma di calendarizzazione (scheduling) di eventi si deve verificare se due appuntamenti della stessa giornata si sovrappongono. Per semplicità, ipotizziamo che gli appuntamenti inizino sempre a un'ora esatta (senza minuti), ma

- 1 controlla che usino l'orario militare (cioè con le ore che vanno da 0 a 23).
- 2 Lo pseudocodice (i quattro valori devono essere interi e compresi tra 0 e 23) determina se l'appuntamento che inizia all'ora start1 e termina all'ora end1 si sovrappone all'appuntamento che inizia all'ora start2 e termina all'ora end2.
- 3 Se $start1 > start2$
- 4 Se $end1 < end2$

Altrimenti

$s = start1$

Altrimenti

$s = start2$

$e = end1$

Altrimenti

Consigli di suddivisione del testo

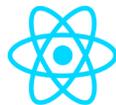
Tecnologie utilizzate

Applicazione web programmata in JavaScript

JS

Frontend

React



Comunicazione tramite
API REST

Backend

Express

express

NeDB (implementazione
in-memory di mongoDB)

 NeDB

 mongoDB.

Test di usabilità



Svolgimento dei task



Domande conclusive



Questionario SUS

Media questionari SUS: 80.3 (superiore alla soglia di 68)

Conclusioni

Gli obiettivi della tesi sono stati raggiunti

Limitazioni

- Periodo di svolgimento di osservazioni e valutazioni
 - Durante il **needfinding** il corso di Informatica si era già concluso
 - Durante la **verifica** il corso di Informatica era appena iniziato

Sviluppi futuri

- Nuove osservazioni e test dell'interfaccia durante il corso di Informatica
- Applicazione dei risultati ottenuti dalla valutazione
- Utilizzo dell'applicazione nei corsi di Informatica

Grazie per l'attenzione