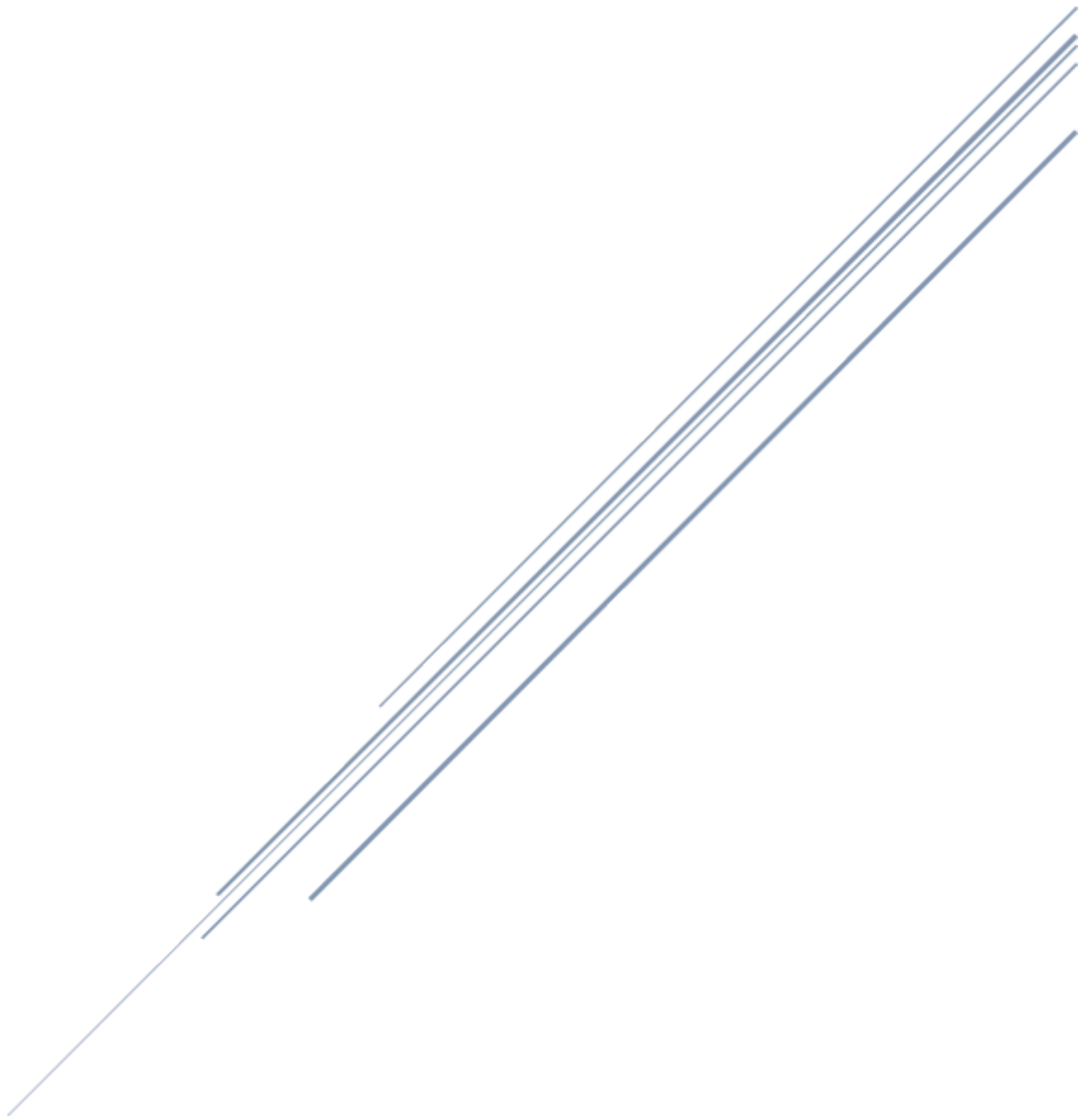


# SOLVE.IT

Documento di progetto



Gruppo T053  
Ingegneria del Software



## Sommario

Obiettivo del progetto	3
Macro-funzionalità del sistema	3
Requisiti funzionali	4
Requisiti non funzionali	6
Front-end	7
Back-end	8

## Obiettivo del progetto

Il progetto Solve.it consiste nella realizzazione di una Web-app accessibile a studenti e docenti dell'università di Trento per mettere a disposizione un servizio automatico di esercitazione e di condivisione di materiale didattico. Il focus dell'applicazione è quello di offrire, agli studenti che accedono al servizio, l'opportunità di esercitarsi con del materiale inerente ai corsi di studio seguiti, nella maniera più efficiente possibile stando allo stesso tempo al passo con le proposte del docente. Il sistema Solve.it si focalizza su esercitazioni di tipologia "problem solving".

## Macro-funzionalità del sistema

1. Il materiale e le esercitazioni pubblicate su Solve.it sono suddivisi in corsi
2. Per ogni corso creato i docenti possono caricare delle esercitazioni e del materiale didattico (video, pdf, documenti, ecc.)
3. Gli studenti che accedono alla piattaforma possono iscriversi a qualunque dei corsi proposti dai docenti e usufruire del materiale caricato e delle esercitazioni pubblicate
4. Solve.it raccoglie delle statistiche sugli esiti delle esercitazioni, che vengono poi mostrate sia agli studenti (statistiche personali) che ai docenti (statistiche sulla singola esercitazione)
5. Gli studenti iscritti ad un corso possono usufruire delle esercitazioni passate per prepararsi all'esame

## Requisiti funzionali

1. Il sistema deve permettere allo studente di svolgere esercitazioni che sono composte da:
  - 1.1. Un testo (in cui viene descritto il problema da risolvere)
  - 1.2. Una tipologia di risposta
  - 1.3. Una soluzione
  - 1.4. Una spiegazione della risoluzione del problema
2. Il sistema deve permettere al docente di creare esercitazioni di cui deve specificare:
  - 2.1. il testo
  - 2.2. la tipologia di risposta
  - 2.3. la soluzione
  - 2.4. la spiegazione della procedura di svolgimento dell'esercitazione.
3. Il sistema deve supportare come formato di risposta per un'esercitazione uno tra i seguenti:
  - 3.1. Risposta numerica
  - 3.2. Crocette a risposta singola
  - 3.3. Crocette a risposta multipla
4. Il sistema deve temporizzare la pubblicazione delle esercitazioni come segue:
  - 4.1. Ogni esercitazione deve avere una data di pubblicazione
  - 4.2. Ogni esercitazione deve avere una data di terminazione (alla quale viene pubblicata la spiegazione della procedura di risoluzione)
5. La descrizione della procedura di risoluzione di una data esercitazione può essere espressa in formato testuale o allegando un relativo documento
6. Il docente deve essere vincolato dal sistema a caricare per ogni esercitazione la spiegazione della procedura di risoluzione
7. Il sistema deve permettere al docente di specificare gli attributi di ogni contenuto che carica, ossia:
  - 7.1. Visibilità
  - 7.2. Nel caso delle esercitazioni se questa è "d'esame": un'esercitazione "d'esame" è un'esercitazione che il docente reputa essere compatibile con la tipologia di esercizi dell'esame del proprio corso. Questa caratterizzazione è necessaria per la generazione delle simulazioni d'esame (vedi requisito funzionale 22)
  - 7.3. Gli attributi di un contenuto pubblicato in un corso possono essere modificati durante o in seguito alla creazione
8. Il sistema deve permettere agli studenti di sottomettere risposte alle esercitazioni
  - 8.1. Agli studenti deve essere concesso di sottomettere un numero arbitrario di soluzioni, ma il sistema deve considerare valida esclusivamente l'ultima risposta specificata
  - 8.2. Il sistema deve valutare in modo automatico le risposte alle esercitazioni fornite dagli studenti in trentesimi
9. Il sistema deve permettere ai docenti di creare nuovi corsi
  - 9.1. Il docente deve poter specificare un abstract per il corso
  - 9.2. Il docente deve specificare il titolo del corso creato

10. Il sistema deve permettere ai docenti di visualizzare l'elenco dei propri corsi
11. Per ogni proprio corso il docente può:
  - 11.1. Visualizzare contenuti pubblicati
  - 11.2. Visualizzare gli attributi di ciascun contenuto
12. Il sistema deve permettere ai docenti di eliminare contenuti pubblicati
13. Il sistema deve permettere al docente di caricare in un corso delle risorse, quali:
  - 13.1. Documenti
  - 13.2. Immagini
  - 13.3. Video
  - 13.4. Audio
14. Tutti gli utenti devono accedere al sistema tramite le proprie credenziali dell'università di Trento
15. Il sistema, a sign-in dello studente effettuato, deve mostrare:
  - 15.1. Tutti i corsi a cui lo studente è iscritto
  - 15.2. L'elenco di tutti i corsi a cui lo studente non è iscritto attualmente
16. Quando lo studente seleziona un corso a cui è iscritto dal relativo elenco, la piattaforma deve mostrare:
  - 16.1. L'elenco dei materiali didattici del corso
  - 16.2. L'elenco delle esercitazioni suddivise in esercitazioni in corso ed esercitazioni già terminate
  - 16.3. L'elenco delle scadenze delle esercitazioni in corso
17. Il sistema deve fornire ad ogni studente una schermata per visualizzare tutti i corsi a cui è iscritto
18. La piattaforma deve mostrare agli utenti i dettagli associati al corso (docente e abstract del corso)
19. La piattaforma deve permettere agli studenti di effettuare una ricerca dagli elenchi di corsi in base ai seguenti criteri:
  - 19.1. Nome del docente
  - 19.2. Nome del corso
20. La piattaforma deve permettere ai docenti di visualizzare l'elenco dei voti per una data esercitazione di un dato corso
  - 20.1. Il sistema deve permettere ai docenti di vedere il numero di studenti che hanno consegnato la risposta ad una certa esercitazione
  - 20.2. Il sistema deve mostrare al docente delle statistiche riassuntive dei voti
21. La piattaforma deve permettere agli studenti di visualizzare per ogni corso i voti di ciascuna esercitazione consegnata
  - 21.1. Il sistema deve mostrare allo studente delle statistiche riassuntive dei propri voti ricevuti
22. Il sistema deve permettere agli studenti di effettuare simulazioni d'esame
  - 22.1. Il sistema deve selezionare a random delle esercitazioni tra quelle marcate come "d'esame" in un numero specificato dall'utente che costituiranno gli esercizi della simulazione d'esame
  - 22.2. Lo studente deve specificare la durata della simulazione prima dell'avvio della stessa

22.3. Il sistema al termine dell'esercitazione deve calcolare in automatico il punteggio e mostrarlo allo studente. Il punteggio della simulazione è calcolato come media ponderata, corretta in trentesimi, dei punteggi ottenuti dallo studente per la consegna di ogni singola esercitazione nella simulazione

## Requisiti non funzionali

1. L'accesso al sistema (meccanismi di sign-in e sign-up) deve avvenire in maniera sicura
  - 1.1. Comunicazioni sicure tra front-end e back-end: tutte le comunicazioni e gli scambi di dati tra front-end e back-end devono avvenire tramite protocollo https
  - 1.2. Durata della sessione utente: se un utente front-end non è attivo (ossia non scambia né messaggi né dati con il server) la sessione scade dopo 20 minuti
  - 1.3. A sign-in effettuato una sessione di attività del front-end deve essere identificata da un token temporaneo riconosciuto dal back-end, che dunque lo deve conservare durante tutto il suo arco di attivazione
  - 1.4. Tutte le richieste https che il client effettua verso il server devono essere marcate con il token della sessione
  - 1.5. Il back-end può fornire risposte alle richieste https dei client solo se il token fornito nella richiesta è valido
  - 1.6. Un token è valido se è presente tra i token attivi noti al back-end ed il tempo trascorso dall'ultima richiesta ricevuta con lo stesso token è inferiore al tempo limite stabilito al requisito non funzionale 1.2.
  - 1.7. Se il token di sessione non è valido il back-end informa il front-end che la sessione non è attiva ed è necessario effettuare il sign-in
2. Al meccanismo di specifica della risposta di un esercizio da parte di un utente loggato vengono applicati i seguenti vincoli:
  - 2.1. La verifica del punteggio della risposta è effettuata a back-end
  - 2.2. Non si può effettuare submitting di risposte a meno di 30 secondi una dall'altra
  - 2.3. Ogni volta che viene effettuato un submit nel database il punteggio viene sovrascritto
3. Usabilità: l'utente è in grado di usare il sistema senza la necessità di essere formato sul suo funzionamento
  - 3.1. Il numero di elementi dell'interfaccia utente front-end di tipologia differente (due voci cliccabili di un elenco di corsi non sono considerati tali) presenti a schermo sono limitati a 10
4. Privacy: il sistema deve proteggere i dati privati di ogni utente dagli altri utenti del sistema
  - 4.1. il sistema non deve prelevare all'utente più delle informazioni essenziali per la sua identificazione all'interno del sistema stesso, ossia la mail dell'ateneo di Trento e il ruolo (docente o studente) all'interno dell'università
  - 4.2. Le statistiche sulle esercitazioni degli studenti visibili ai docenti sono anonime
5. Compatibilità: il sistema supporta l'utilizzo da computer, smartphone e tablet, purché siano dotati di uno tra i seguenti browser: Chrome (versione 105 o successiva), Firefox (versione 104 o successiva) e Microsoft Edge (versione 105 o successiva)



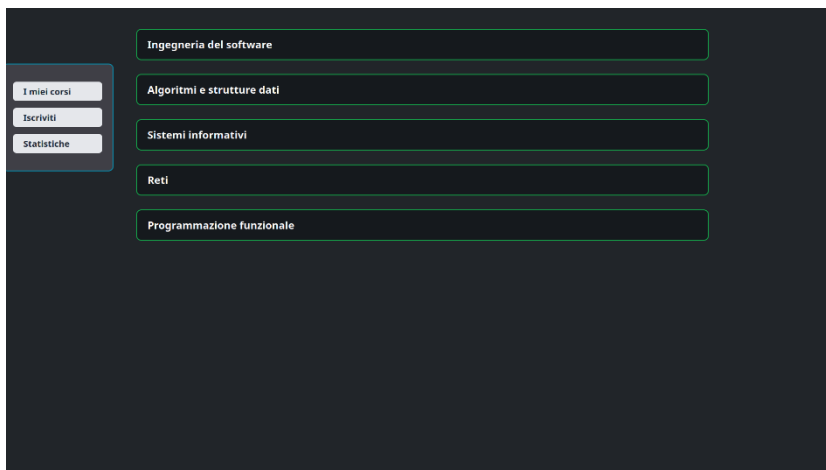
## Front-end

### 1. Schermata home della piattaforma

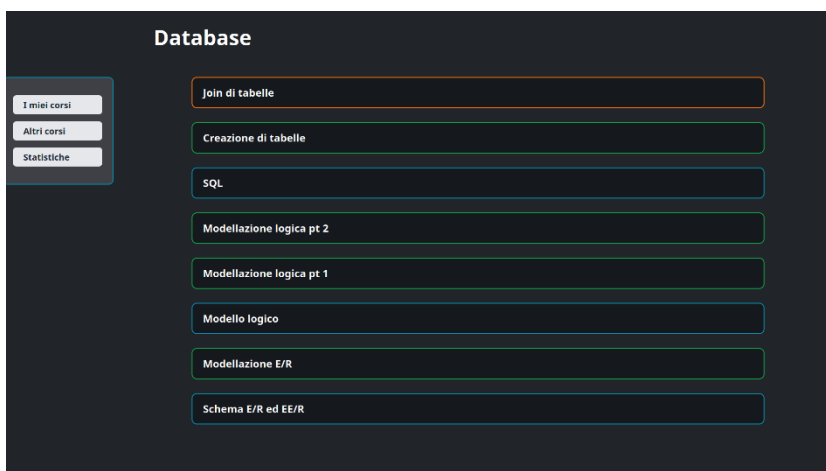


### 2. Schermate dell'interfaccia studente

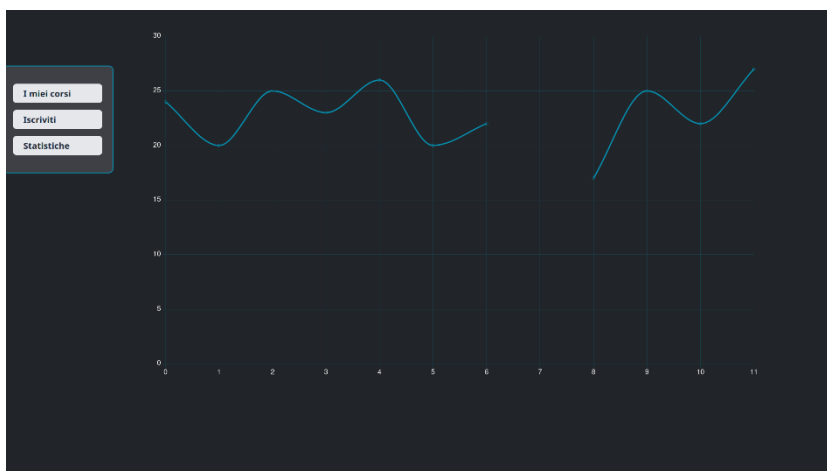
#### 2.1. Visualizzazione dell'elenco dei corsi



#### 2.2. Visualizzazione del materiale di un corso

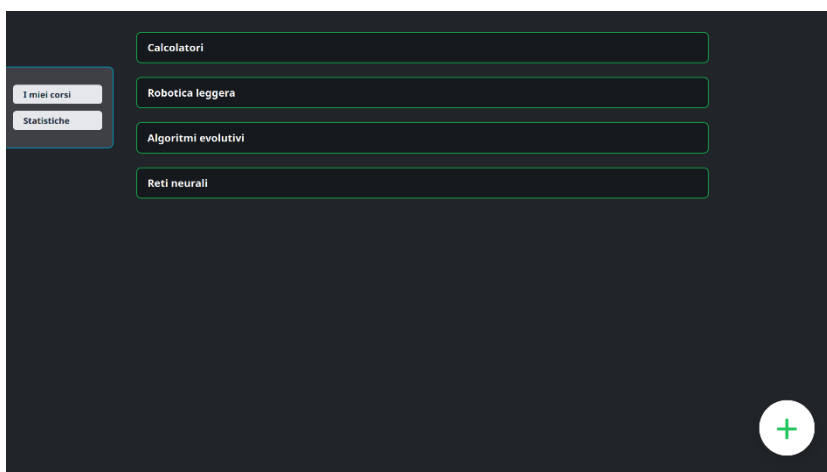


## 2.3. Statistiche sulle esercitazioni completate



## 3. Schermate dell'interfaccia docente

### 3.1. Visualizzazione dell'elenco dei propri corsi



### 3.2. Fill-in delle informazioni essenziali di un corso alla sua creazione (come

specificato dai requisiti funzionali 7.1 e 7.2)

3.3.

A screenshot of a web interface for creating a new course. On the left, there is a sidebar with two buttons: "I miei corsi" and "Statistiche". The main area contains a form with two text input fields: "Titolo del corso" and "Descrizione del corso". Below the form is a green button labeled "Crea corso".

## Back-end

Solve.it necessita delle seguenti API esterne per il suo funzionamento

- UNITN: le API di Unitn saranno utilizzate per la gestione dell'accesso al sito
- WOLFRAM: le API di Wolfram saranno utilizzate per interpretare risposte esplicitate in forma di espressioni matematiche
- MONGODB: useremo Mongodb per salvare tutti i dati relativi agli utenti, le sessioni attive, gli accessi, e i dati necessari all'applicazione per funzionare

