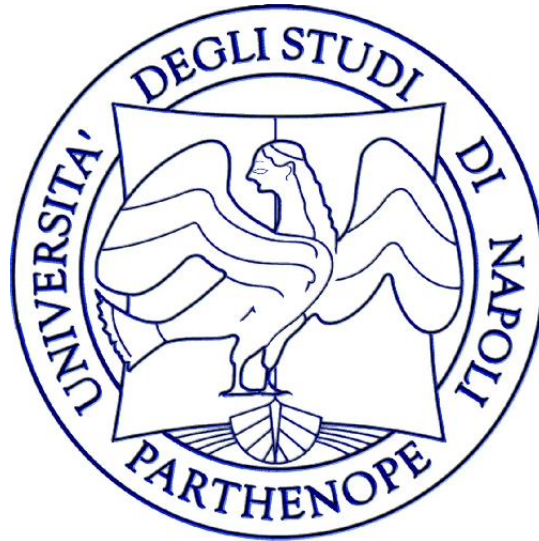


UNIVERSITÀ DEGLI STUDI DI NAPOLI “PARTHENOPE”
FACOLTÀ DI SCIENZE E TECNOLOGIE
CORSO DI LAUREA IN INFORMATICA



PROGETTO DI RETI DI CALCOLATORI

Università

DOCENTE

Emanuel Di Nardo

STUDENTI

Rocco Del Prete - 0124002592

Antonio Capasso - 0124002624

Luca Amoroso - 0124002512

Anno Accademico 2023-2024

Indice

1	Descrizione del progetto e dell'architettura	2
1.1	Componenti coinvolte	3
2	Manuale utente	5
2.1	Server universitario	6
2.1.1	Compilazione ed esecuzione	6
2.2	Segreteria	7
2.2.1	Compilazione ed esecuzione	7
2.3	Studiante	8
2.3.1	Compilazione ed esecuzione	8
3	Database	9
3.1	Diagramma EE/R	9
3.2	Studiante	10
3.3	Prenotazione	11
3.4	Appello	12
3.5	Esame	13

Traccia

Scrivere un'applicazione client/server parallelo per gestire gli esami universitari.

- **Segreteria:**

- Inserisce gli esami sul server dell'università (salvare in un file o conservare in memoria il dato)
- Inoltra la richiesta di prenotazione degli studenti al server universitario
- Fornisce allo studente le date degli esami disponibili per l'esame scelto dallo studente

- **Studente:**

- Chiede alla segreteria se ci siano esami disponibili per un corso
- Invia una richiesta di prenotazione di un esame alla segreteria

- **Server universitario:**

- Riceve l'aggiunta di nuovi esami
- Riceve la prenotazione di un esame

Il server universitario ad ogni richiesta di prenotazione invia alla segreteria il numero di prenotazione progressivo assegnato allo studente e la segreteria a sua volta lo inoltra allo studente

Se la segreteria non risponde alla richiesta dello studente questo deve ritentare la connessione per 3 volte. Se le richieste continuano a fallire allora aspetta un tempo random e ritenta. Simulare un timeout della segreteria in modo da arrivare a testare l'attesa random.

Capitolo 1

Descrizione del progetto e dell'architettura

Di seguito viene proposto un diagramma per descrivere l'architettura utilizzata nel progetto.

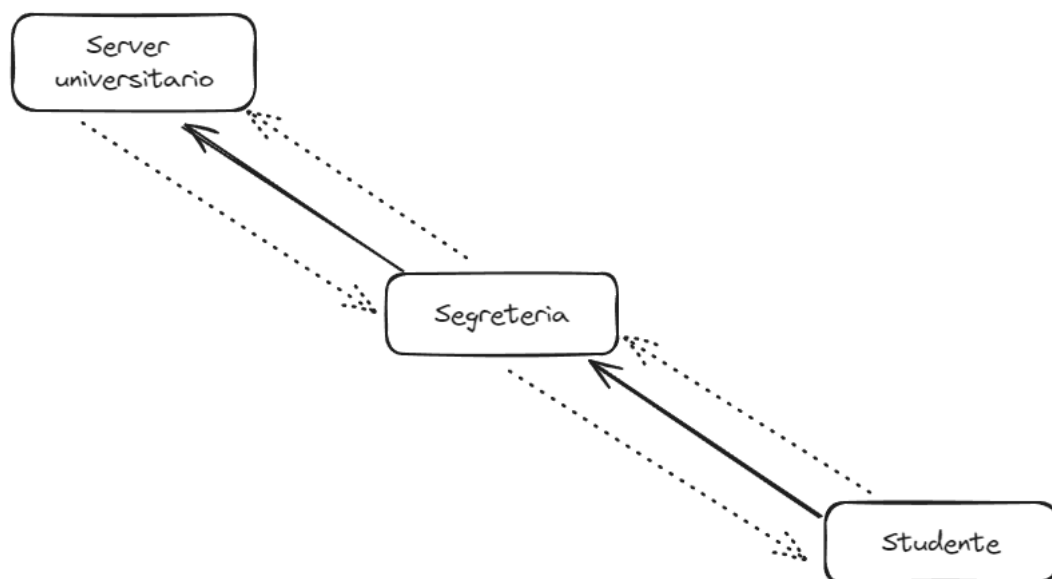


Figura 1.1: Diagramma architettura

Andando a descrivere il diagramma:

- le frecce con linea continua indicano il collegamento tra le componenti;
- le frecce con linea tratteggiata indicano i versi di comunicazione.

1.1 Componenti coinvolte

Da come si evince dalla figura, le componenti coinvolte sono:

- **Server universitario:** questa componente funge da server per la segreteria e ha i compiti di:
 - aggiungere una prenotazione per un determinato esame: tale richiesta viene effettuata dallo studente alla segreteria, la quale procede ad inoltrarla al server universitario. Inoltre, il server universitario invia alla segreteria il numero di prenotazione progressivo, il quale viene inoltrato allo studente;
 - aggiungere un appello per un determinato esame: tale richiesta viene effettuata dalla segreteria.
- **Segreteria:** questa componente funge da server per lo studente e da client per il server universitario, con compiti di:
 - inviare al server universitario l'appello di un esame da aggiungere;
 - inoltrare la richiesta di prenotazione ad un appello, effettuata da uno studente, al server universitario;
 - fornire allo studente le date di tutti gli appelli d'esame disponibili per il suo corso di studi;
 - fornire allo studente le date degli appelli di un esame specificato dallo studente.
- **Studente:** questa componente funge da client per la segreteria, con compiti di:
 - richiedere alla segreteria di accedere alla sua area riservata, attraverso l'inserimento di matricola e password (login);
 - richiedere alla segreteria tutti gli appelli d'esame disponibili per il suo corso di studi;
 - richiedere alla segreteria gli appelli di un esame da egli specificato;
 - inviare alla segreteria una richiesta di prenotazione ad un determinato appello d'esame.

CAPITOLO 1. DESCRIZIONE DEL PROGETTO E DELL'ARCHITETTURA

Se durante le richieste effettuate la segreteria smette di rispondere, lo studente ritenta la connessione per tre volte, dopodiché, se continuano a fallire, attende un tempo casuale e ritenta.

Capitolo 2

Manuale utente

Di seguito vengono proposte le istruzioni per l'uso dell'applicativo.

Come primo passo, bisogna aprire il progetto posizionandosi all'interno della cartella `src`

2.1 Server universitario

Si vanno a descrivere le istruzioni per compilare ed eseguire il codice relativo alla componente del server universitario.

2.1.1 Compilazione ed esecuzione

Per poter eseguire la compilazione, si esegue il comando:

```
gcc server.c -o server -lmysqlclient
```

che genererà un file chiamato `server`, eseguibile con il comando:

```
./server
```

L'opzione `-lmysqlclient` indica al compilatore di collegare (link) il programma con la libreria `mysqlclient`, che permette di connettersi ed interfacciarsi col database.

2.2 Segreteria

Si vanno a descrivere le istruzioni per compilare ed eseguire il codice relativo alla componente della segreteria.

2.2.1 Compilazione ed esecuzione

Per poter eseguire la compilazione, si esegue il comando:

```
gcc segreteria.c -o segreteria -lmysqlclient
```

che genererà un file chiamato `segreteria`, eseguibile con il comando:

```
./segreteria <IP address>
```

L'opzione `-lmysqlclient` indica al compilatore di collegare (link) il programma con la libreria `mysqlclient`, che permette di connettersi ed interfacciarsi col database.

Inoltre, bisogna anche specificare l'indirizzo IP, che in fase di test può essere `127.0.0.1` (localhost).

2.3 Studente

Si vanno a descrivere le istruzioni per compilare ed eseguire il codice relativo alla componente dello studente.

2.3.1 Compilazione ed esecuzione

Per poter eseguire la compilazione, si esegue il comando:

```
gcc studente.c -o studente
```

che genererà un file chiamato **studente**, eseguibile con il comando:

```
./studente <IP address>
```

Da come si evince bisogna anche specificare l'indirizzo IP, che in fase di test può essere 127.0.0.1 (localhost).

Capitolo 3

Database

3.1 Diagramma EE/R

Di seguito viene mostrato il diagramma EE/R del database utilizzato.

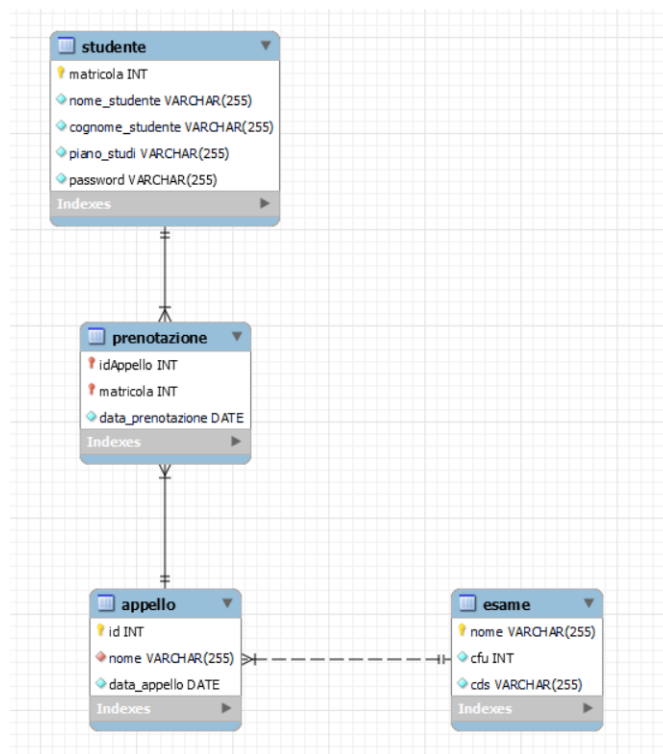


Figura 3.1: Diagramma database

3.2 Studente

La tabella **studente** contiene i seguenti campi:

- **matricola**: campo che rappresenta la matricola dello studente; è la chiave primaria della tabella;
- **nome_studente**: campo che rappresenta il nome dello studente;
- **cognome_studente**: campo che rappresenta il cognome dello studente;
- **piano_studi**: campo che rappresenta il piano di studi dello studente;
- **password**: campo che rappresenta la password dello studente, crittografata in SHA-2.

3.3 Prenotazione

La tabella **prenotazione** contiene i seguenti campi:

- **idAppello**: identificativo dell'appello a cui lo studente si è prenotato; è chiave esterna e fa riferimento al campo **id** della tabella **appello**;
- **matricola**: matricola dello studente che ha effettuato la prenotazione; è chiave esterna e fa riferimento al campo **matricola** della tabella **studente**;
- **data_prenotazione**: data in cui lo studente ha effettuato la prenotazione.

La coppia (**idAppello**, **matricola**) è la chiave primaria di questa tabella.

3.4 Appello

La tabella **appello** contiene i seguenti campi:

- **id**: identificativo dell'appello; è la chiave primaria della tabella;
- **nome**: nome dell'esame; è chiave esterna e fa riferimento al campo **nome** della tabella **esame**;
- **data_appello**: data dell'appello.

3.5 Esame

La tabella **esame** contiene i seguenti campi:

- **nome**: nome dell'esame; è chiave primaria della tabella;
- **cfu**: CFU dell'esame;
- **cds**: corso di studi al quale appartiene l'esame.