

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
Santo Femiano - 0124002426
Salvatore Santaniello - 0124002091

Anno Accademico 2023-2024

Indice

1	Traccia	1
2	Descrizione del progetto	3
2.1	Server-Segreteria-Studenti	5
3	Manuale utente	6
3.1	Server universitario	6
3.1.1	Compilazione ed esecuzione	6
3.2	Segreteria	7
3.2.1	Compilazione ed esecuzione	7
3.3	Studente	8
3.3.1	Compilazione ed esecuzione	8
3.4	Alternativa	8
3.4.1	Compilazione ed esecuzione	8
4	Database	9
4.1	Diagramma EE/R	9
4.2	Exams	9
4.3	Exam Sessions	11
4.4	Booking	12

Capitolo 1

Traccia

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

- **Server universitario:**

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

- **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

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

Capitolo 2

Descrizione del progetto

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

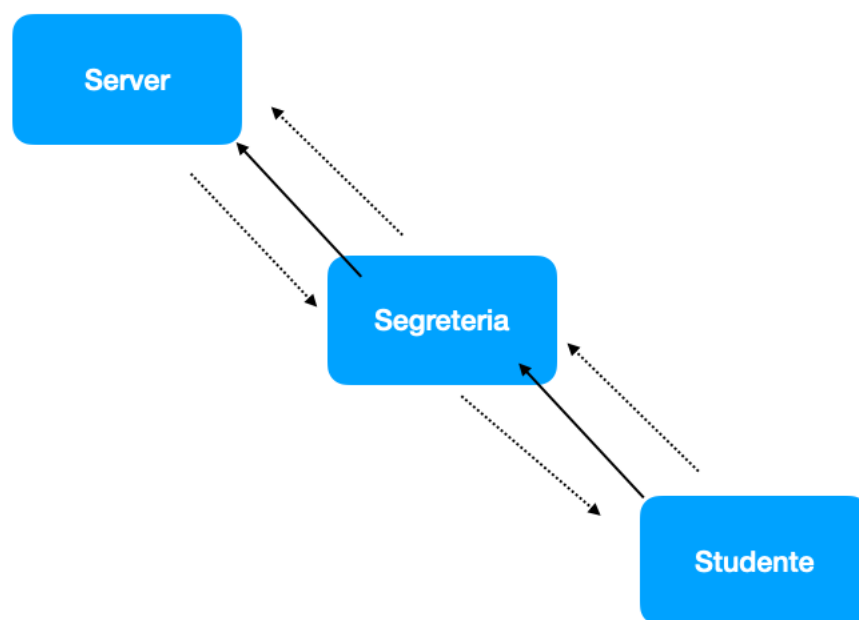


Figura 2.1: Diagramma architettura

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.

2.1 Server-Segreteria-Studenti

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 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 3

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`

3.1 Server universitario

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

3.1.1 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.

3.2 Segreteria

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

3.2.1 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).

3.3 Studente

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

3.3.1 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).

3.4 Alternativa

Utilizzo di Docker tramite Clion , utilizzando i container e il bridge.

3.4.1 Compilazione ed esecuzione

Per ogni Server/client avviare tramite container. Nel caso di Segreteria e Studenti indicare nelle impostazione di avvio l'indirizzo esatto del container relativo associato allo stesso gruppo di Network (Bridge).

Capitolo 4

Database

4.1 Diagramma EE/R

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

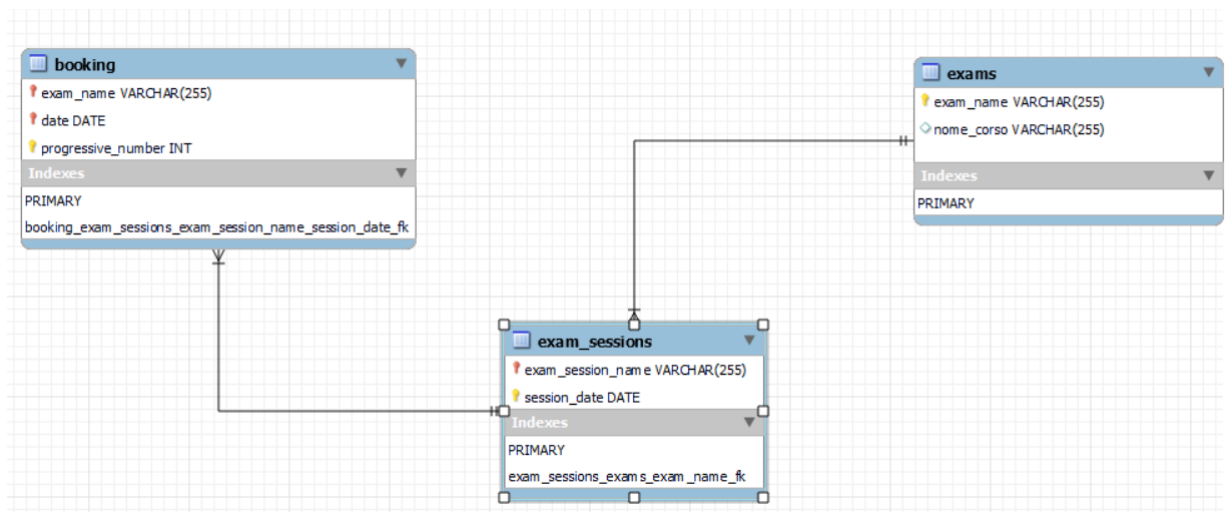


Figura 4.1: Diagramma database

4.2 Exams

La tabella Exams contiene i seguenti campi:

- **Exam Name:**[Primary Key] campo che rappresenta il nome dell'esame

- **Nome Corso:** campo che rappresenta il nome del corso

4.3 Exam Sessions

La tabella `Exam Sessions` contiene i seguenti campi:

- **Exam Session Name:** campo che rappresenta il nome dell'appello di quella sessione.
- **Session Date:** campo che rappresenta la data dell'appello di quella sessione.

La [Primary Key] è la combinazione tra i campi `ExamSessionName` e `SessionDate`, inoltre c'è una [Foreign Key] su `ExamSessionName` per referenza `exam-name` di `exams`.

4.4 Booking

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

- **Exam Name:** identificativo del nome dell'esame [appello].
- **Date:** data dell'appello.
- **Progressive Number:** rappresenta il numero progressivo della prenotazione a quello specifico appello.

La [Primary Key] è la combinazione tra i campi Exam Name , Date e Progressive Number inoltre c'è una [Foreign Key] su Exam Name e Date per referenza Exam Session Name e Exam Session Date.