



A.A. 2022-2023



ID Gruppo: LSO_2122_23



Valentino Bocchetti - N86003405



Dario Morace -



Lucia Brando -

Indice

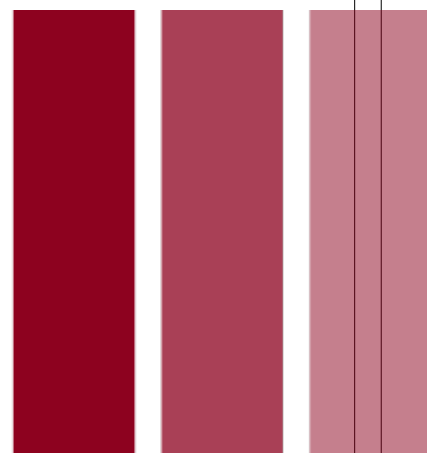
1	Revisioni	3
2	Presentazione	3
3	Guida al Server	3
3.1	Funzionalità	3
3.2	Scelte implementative	4
3.3	Tecnologie e strumenti utilizzati	4
3.4	Memorizzazione dei dati	4
4	Guida al Client	4
4.1	Primo avvio	4
4.2	Post registrazione	4
4.3	Memorizzazione delle informazioni	4
4.4	Modelli di Dominio	4
5	Protocollo applicativo	4
6	Dettagli implementativi	5
6.1	Server	5
6.2	Client	5
7	Codice sorgente sviluppato	5
8	Contributori	5
9	Ringraziamenti	5

1 Revisioni

2 Presentazione

INFOPOINT

Educate Yourself



InfoPoint® è un progetto che nasce per offrire un supporto ai visitatori del museo.

Questo progetto si concretizza in 2 componenti ben definite:

- ▶ Un backend scritto in C per la gestione dei dati, hostato¹ su una macchina virtuale offerta da Azure;²
- ▶ Una applicazione **Android**, scritta in **Java** che fa da client;

3 Guida al Server

3.1 Funzionalità

Il Sistema, deve offrire, una serie di funzionalità:

- ▶ Possibilità di connessione concorrente;
- ▶ Possibilità di potersi registrare alla piattaforma;³
- ▶ Possibilità di usufruire dei contenuti in base alla tipologia di utente, in modo da permettere un focus diverso in base alle sue caratteristiche;⁴

¹ Indica un servizio di rete che consiste nell'allocare su un server web delle pagine web di un sito web o di un'applicazione web, rendendolo così accessibile dalla rete Internet e ai suoi utenti.

² Per maggiori informazioni visitare il seguente [sito](#).

³ Le credenziali vengono salvate facendo uso di un Database, che risulta molto più affidabile di un semplice file di testo.

⁴ Ricordiamo che il bacino degli utenti che possono fare uso del sistema può variare da scolaresche, famiglie o esperti.

3.2 Scelte implementative

Seguendo il concetto del *DIVIDE ET IMPERA*⁵ si è scelto di spezzare le varie funzionalità che vengono messe a disposizione per rendere il codice facilmente manutenibile ed evitare lo stato di codice monolitico.⁶

3.3 Tecnologie e strumenti utilizzati

Per una migliore gestione del Sistema, si è fatto uso di una serie di strumenti.

Durante lo sviluppo si è fatto uso dell'utility [cmake](#),⁷ tool modulare che permette la generazione di un Makefile⁸

3.4 Memorizzazione dei dati

4 Guida al Client

4.1 Primo avvio

4.2 Post registrazione

4.3 Memorizzazione delle informazioni

4.4 Modelli di Dominio

Class Diagram

Sequence Diagram

5 Protocollo applicativo

Come già indicato in precedenza abbiamo preferito il protocollo TCP rispetto al protocollo UDP, per la presenza di un controllo della congestione e affidabilità in termini di **invio/ricezione** di dati.⁹

Lo sviluppo dell'applicativo è stato inizialmente verticalizzato sulla creazione dello scheletro del Server, per avere un primo approccio nudo e crudo allo scambio di messaggi via **socket**.

Per avere un programma robusto e manutenibile si è fatto largo uso delle **good practices** che questo tipo di comunicazione richiede. In particolare:

⁵Metodologia per la risoluzione di problemi → Il problema viene diviso in sottoproblemi più semplici e si continua fino a ottenere problemi facilmente risolvibili. Combinando le soluzioni ottenute si risolve il problema originario.

⁶Che risulta notoriamente più difficile da gestire e modificare nel tempo.

⁷Per maggiori informazioni visitare il seguente [sito](#).

⁸Che contiene tutte le direttive utilizzate dall'utility make.

⁹Ricordiamo infatti che UDP non ha garanzie sulla trasmissione dei pacchetti, seguendo la logica di best-effort.

- ▶ La connessione viene aperta solo nel momento in cui devono essere **inviati/ricevuti** dati (Si evita in questo modo di tenere aperte connessioni in momenti in cui queste non vengono sfruttate);
- ▶ Si effettuano controlli di raggiungibilità del server lato client;¹⁰
- ▶ Vengono controllati i dati **inviati/ricevuti** sempre prima di compiere operazioni che possano minare il corretto funzionamento di **Server** e **Client**;¹¹
- ▶ Vengono effettuati controlli e gestione degli stati di tutte le operazioni lato **Server**.

6 Dettagli implementativi

6.1 Server

6.2 Client

7 Codice sorgente sviluppato

Il codice sorgente prodotto durante lo sviluppo di *InfoPoint*[®] è disponibile sulla piattaforma *GitHub*, che ne ha permesso anche il versionamento.

Di seguito riportiamo un link per il [download](#)¹²

8 Contributori

9 Ringraziamenti

Ringraziamo la professoressa [Alessandra Rossi](#) per lo splendido corso, che ci ha permesso di conoscere nuove interessanti tecnologie e del supporto offertoci durante e dopo le lezioni.

¹⁰Non ha senso infatti tenere aperta una connessione se non utilizzata, anzi si rischia anche di causare interruzione di servizio dovuti a timeout improvvisi.

¹¹Questo avviene anche attraverso un particolare pattern di costruzione dei dati.

¹²Potrebbe non essere accessibile a tutti (il repository è per privacy privato).