

RELAZIONE PROGETTO DI TELEMATICA

Andrea Bertogalli - Nicolò Tombini

2021-2022



Sommario

[Introduzione 3](#_Toc106394150)

[struttura del progetto 3](#_Toc106394151)

[Frontend 3](#_Toc106394152)

[Backend 4](#_Toc106394153)

[Database 4](#_Toc106394154)

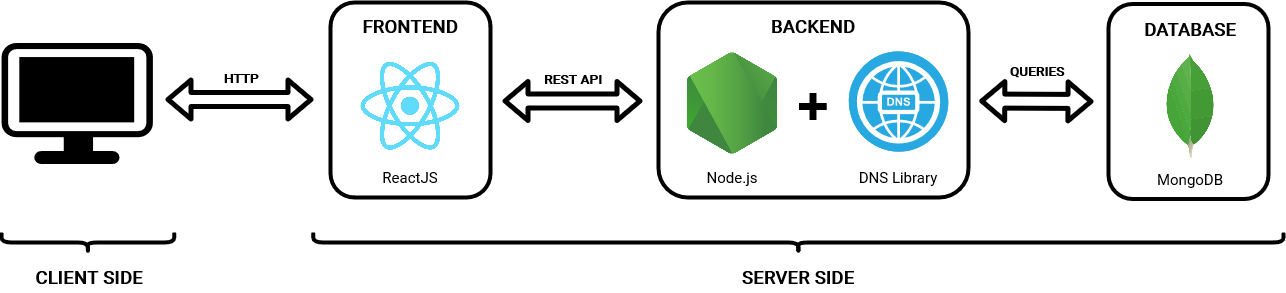
[Guida utente 4](#_Toc106394155)

[Conclusioni 4](#_Toc106394156)

# Introduzione

Per quanto riguarda l’attività di progetto per il corso di Telematica si è deciso di sviluppare un Resolver DNS utilizzabile mediante delle REST API quindi mediante HTTP, inoltre per rendere più user-friendly il progetto si è voluta sviluppare una GUI accessibile mediante il browser.

# struttura del progetto



Lo schema riportato mostra in maniera generale la struttura del progetto, tutto il progetto è stato sviluppato lato server il client (browser) si limita ad accedere alla web-application e interagire con il backend attraverso di essa. Lato server distinguiamo tre macro-blocchi:

* Frontend
* Backend
* Database

È bene specificare che frontend e backend sono due processi server, due server NodeJS, distinti.

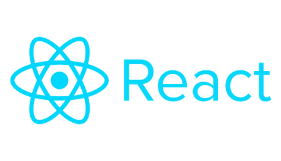
## Frontend

Per frontend si intende tutto quello che è visualizzabile sottoforma di GUI da parte dell’utente, in particolare per lo sviluppo del frontend si è preferito utilizzare [ReactJS](https://it.reactjs.org/).

*React è una libreria open-source, front-end, JavaScript per la creazione di interfacce utente. È mantenuto da Meta e da una comunità di singoli sviluppatori e aziende.*

React permette appunto di sviluppare interfacce grafiche che siano molto efficienti, questo deriva dal fatto che React utilizza un sofisticato meccanismo di sincronizzazione tra un DOM virtuale e quello reale e permette quindi di aggiornare la pagina senza doverla ricaricare ma solamente aggiornando i vari pezzi che vanno aggiornati. Inoltre il frontend girando su NodeJS fa si che si possano integrare una grande varietà di moduli, i principali moduli che sono stati utilizzati, oltre a react ovviamente, sono:

* **Axios:** modulo estremamente utile che permette di effettuare in maniera estremamente efficace qualsiasi richiesta http e largamente utilizzato con React.
* **Material-ui:** questa libreria sviluppata da Google nel 2014 è attualmente la libreria appunto usata nei Frontend Google, mette a disposizione una vasta gamma di componenti grafici integrabili con React.
* **React-router-dom:** questo modulo è stato di estrema importanza per quanto riguarda la gestione degli accessi alle pagine e ai componenti, integrandosi con il metodo di autenticazione sviluppato.

Sono stati usati anche altri moduli di minor importanza quali: js-sha256 utile per generare hash sha256, vari set di icone ecc…

## Backend

Il backend ha varie funzionalità, tra queste distinguiamo ovviamente quella di Resolver DNS, quella di server HTTP che mette a disposizione delle API REST per l’utilizzo del Resolver DNS in maniera estremamente semplice e controllata e infine quella di accesso e gestione della base di dati No-SQL.

Per lo sviluppo sono state utilizzate le seguenti tecnologie:

* **Express:** Express.js è un [framework per applicazioni web](https://it.wikipedia.org/wiki/Framework_per_applicazioni_web) per [Node.js](https://it.wikipedia.org/wiki/Node.js) serve per creare web applications e [API](https://it.wikipedia.org/wiki/Application_programming_interface) ed ormai definito il server framework standard de facto per [Node.js](https://it.wikipedia.org/wiki/Node.js).
* **Nodemon:** un tool molto utile durante lo sviluppo del server in quanto permette di mantenere aggiornato il server senza doverlo riavviare, funziona in maniera simile al rendering di react, ma per applicazioni node.
* **Mongoose:** Dato l’utilizzo di MongoDB come DBMS è stato necessario installare su node il modulo che ne permettesse l’interazione.

## Database

Per il database è stato utilizzato MongoDB che è uno dei principali software per basi di dati NoSQL, MongoDB è basato su documenti JSON organizzati in collezioni. Sulla base di dati vengono salvate le informazioni dell’utente username, password (sha-256), nome, cognome e inoltre vengono salvate la cronologia delle ricerche e le preferenze di configurazione per quanto riguarda i DNS.

La scelta di utilizzare MongoDB è che si integra perfettamente con Javascript e quindi con NodeJS e inoltre è estremamente prestazionale.

## Steps di sviluppo del progetto

# Guida utente

# Conclusioni