



Elaborato Esame di Stato

a.s. 2020/2021

Materie caratterizzanti:

Informatica, Sistemi e Reti

“Autostrade Toscana Manutenzione Infrastrutture”



(Viadotto Italia, CS)

Docente Referente:

Giuseppe Scaranello

Candidato:

Lorenzo Bartolini

Indice

Indice	1
Abstract	2
Architettura Software	3
Frontend con React.js	4
Backend con API in PHP	5

Abstract

Il progetto tratta la realizzazione di un sistema informativo finalizzato alla gestione di sensori per la manutenzione di ponti e viadotti.

Viene, inoltre, prevista lo sviluppo di un portale web che permetta alle società di manutenzione di accedere agli appalti aperti nella regione con la successiva possibilità di eseguirne la manutenzione.

Vengono monitorati tre diversi parametri di interesse: l'elettricità, la struttura e l'asfalto.

Per il monitoraggio vengono utilizzati varie tipologie di sensori che comunicheranno il loro stato ad un dispositivo.

Il suddetto analizzerà i valori ricevuti e invierà alla base di dati centralizzata, tramite un canale trasmissivo sicuro, i valori aggregati dei diversi sensori presenti sul posto.

Viene predisposto per il Ministero dei Trasporti un accesso sicuro e diretto ai dati prodotti dai sensori.

I valori dei sensori memorizzati all'interno della base di dati sono espressi con un numero che ne indica la bontà; questo numero varia da 0, valore più basso che indica lo stato peggiore in cui può essere il particolare parametro di analisi, a 100, valore più alto indicante l'assenza di problemi al dato parametro d'interesse.

I sensori campionano e comunicano i dati una volta al giorno regolarmente.

Architettura Software

Durante la fase di analisi di un progetto viene definita l’architettura software utilizzata che, in questo caso, si basa sul concetto di sistema distribuito.

Con sistema distribuito viene identificata quell’applicazione i cui componenti fisici e logici risiedono lontani e separati tra loro.

Per questo progetto è stata sviluppata un’architettura *n-tier* o *multistrato*.

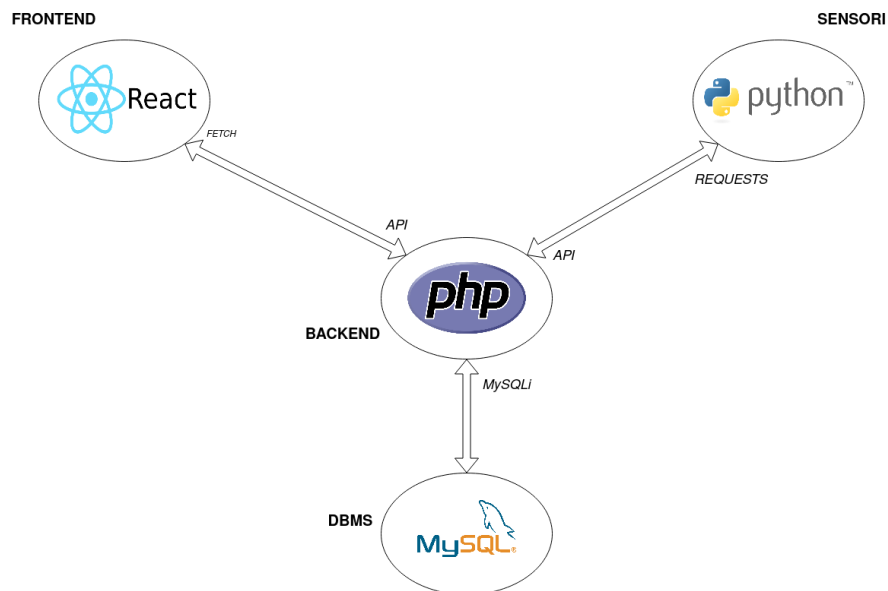


figura 1

Come è possibile vedere nell’immagine sopra, si identificano 4 diverse sezioni divise su 3 strati: presentazione, logica e risorse.

Lo strato di presentazione identifica le parti software che si occupano di prendere i dati e di visualizzarli all’utente.

Questo viene solitamente effettuato tramite un’interfaccia grafica in un web browser.

Un web browser è un client che dialoga, mediante protocollo HTTP, con un server che fornisce i dati da presentare.

Lo strato di logica si pone come intermediario tra la presentazione e le risorse.

Si occupa perciò di soddisfare le richieste dei client, presentazione, fornendogli i dati necessari, risorse.

Questi dati devono essere formattati in modo agevole al client per essere utilizzati.

Lo strato di risorse identifica le componenti software che non prevedono un’interazione diretta con l’utente; al contrario, solitamente forniscono dati che sono stati precedentemente memorizzati al loro interno, basi di dati, oppure forniscono dati che generano loro stessi, sensori.

Si identifica nello strato logico una funzione cardine e di vitale importanza per l’intero progetto in quanto deve essere in grado di gestire i dati provenienti da ogni strato superiore e inferiore.

Frontend con React.js

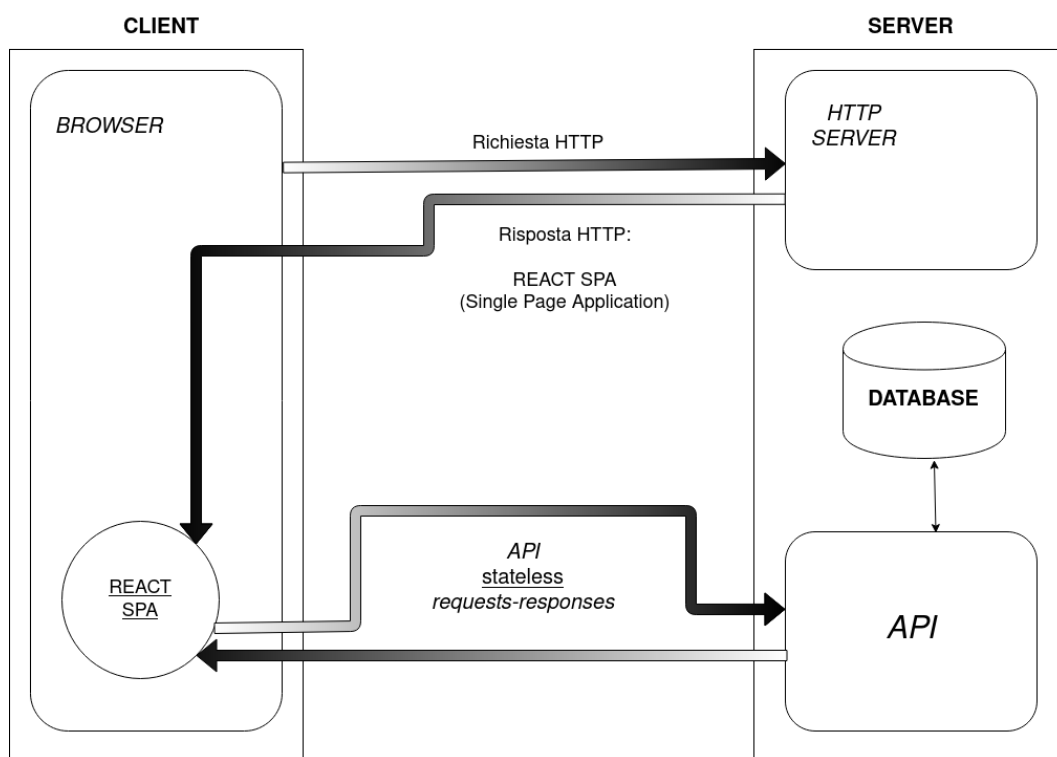
Nello specifico, analizzando lo strato di presentazione, solitamente chiamato Frontend, è stato scelto di utilizzare un framework JavaScript: React.js.

JavaScript è un linguaggio che viene integrato con HTML e CSS per la visualizzazione di pagine web interattive, proprio grazie a JavaScript, mentre un framework non è altro che un insieme di codice che permette ai programmatori astrarre alcuni processi complessi.

Ciò che ha permesso la separazione concettuale tra lo strato di presentazione e quello di logica è proprio l'utilizzo di React.

La differenza tra l'uso di React e lo sviluppo classico di pagine web è che React genera delle SPA, Single Page Application.

Il funzionamento logico di React è descritto nella seguente immagine:



In questa immagine si identificano due entità: il client e il server.

Il client identifica colui che richiede un servizio mentre il server colui che lo fornisce; in questo caso il servizio è la pagina web e il client non è altro che il web browser.

Nel momento in cui il browser invia la richiesta ad un determinato URL, questa viene gestita dal servizio HTTP Server presente sulla macchina ricevente.

Questo restituisce al client una pagina web composta da HTML, linguaggio di presentazione delle informazioni, CSS, linguaggio di grafica che permette di modificarne l'aspetto, infine il JavaScript, che rende la pagina dinamica tramite l'utilizzo di React.

Una volta che il client riceve la pagina web non avrà più bisogno di comunicare con il Server HTTP; questa è la maggiore differenza con lo sviluppo classico di pagine web.

Nel momento in cui la pagina web ha necessità di alcune specifiche informazioni, queste verranno richieste ad un altro servizio presente sul server: l'API, Application Programming Interface.

La comunicazione con l'API avverrà scambiandosi un particolare tipo di pacchetti HTTP contenenti dati in formato JSON e non HTML.

Questa scelta progettuale garantisce velocità e fluidità all'intero sito perchè vengono scambiati solo pochi dati ogni volta che è necessario cambiare schermata o visualizzare informazioni diverse.

Backend con API in PHP