ISTITUTO STATALE DI ISTRUZIONE SUPERIORE
GOBETTI VOLTA

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

Abstract

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

Viene, inoltre, previsto 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.

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

Per il monitoraggio si utilizza 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.

E' 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, a 100, valore più alto indicante l'assenza di problemi.

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*.

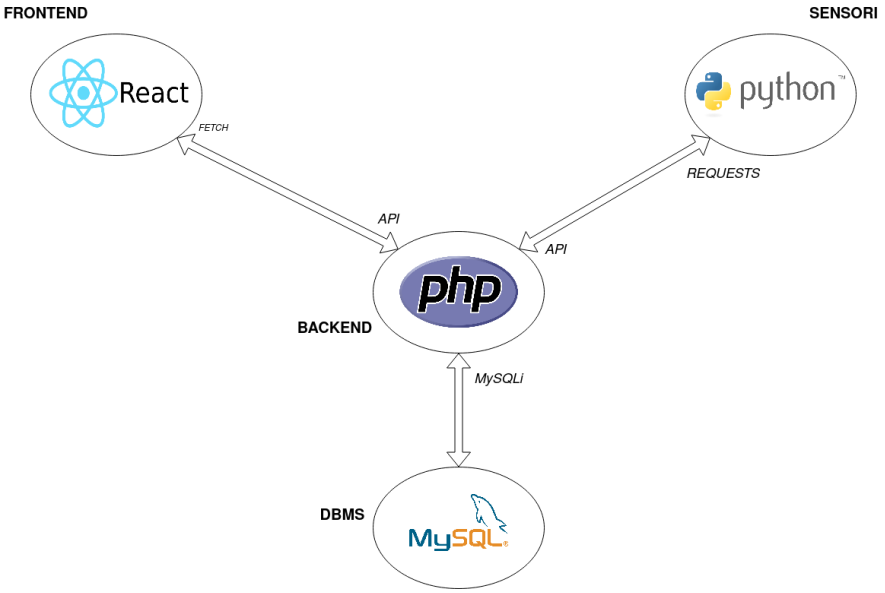

figura 1

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 genere delle SPA, Single Page Application.

Il funzionamento logico di React è descritto nella seguente immagine:

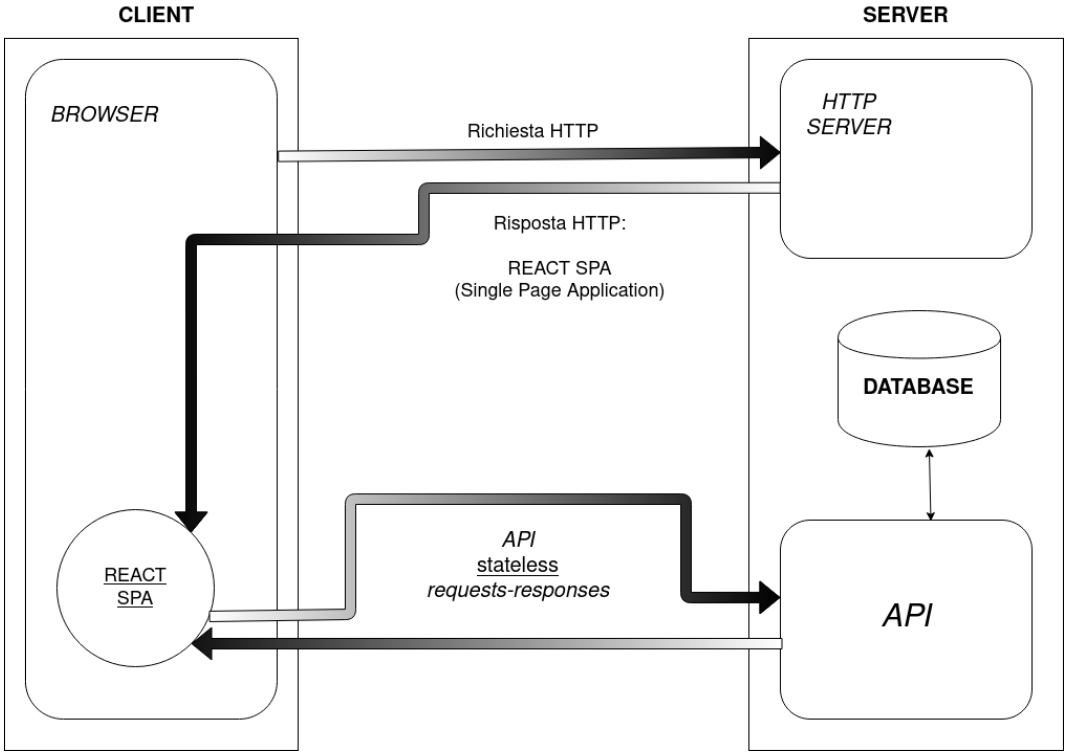

figura 2

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

Per quanto riguarda lo strato di logica, detto Backend, è stato deciso di codificarlo in linguaggio PHP mediante la realizzazioni di API
Un'API, Application Programming Interface, è un'interfaccia software che permetta lo scambio di dati tra un client ed un server.
Solitamente viene sviluppata in un'architettura web mediante protocollo HTTP basandosi sui suoi metodi; principalmente GET e POST.
Questi due metodi permettono la ricezione di dati dal server, GET, e la possibilità di inviare dati, POST.
La principale differenza tecnica è che i parametri inviati tramite GET si trovano all'interno dell'URL; nel POST i parametri verranno inviati all'interno del BODY del pacchetto HTTP.
Le interfacce realizzate hanno lo scopo di rispondere solo ad una determinata richiesta proveniente da particolari tipi di utenti.
Esistono tre diverse categorie di utente all'interno di questo sistema: ministero, società autostradale, società di manutenzione.

Il processo di accesso all'area riservata è il medesimo per ogni utente, sarà il Frontend a preoccuparsi di mostrare a schermo le informazioni dedicate ad ogni utente.

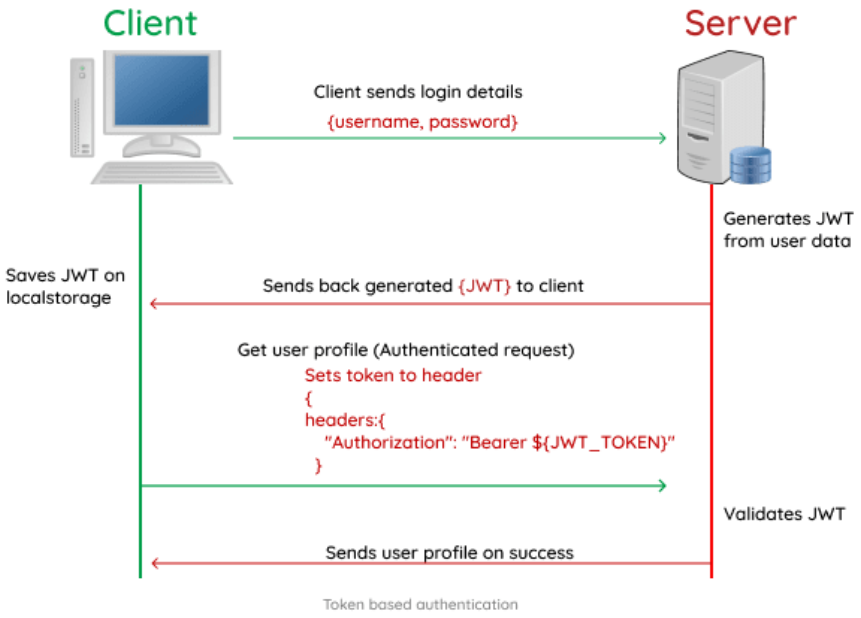

figura 3

Quello descritto in figura è il processo di autenticazione basato sui token, una stringa alfanumerica generata a partire da una stringa relativa all'utente, per esempio l'email, e da una segreta usata successivamente per la verifica del token stesso.

Questi token inoltre hanno una durata per evitare che, una volta generato, non sia più necessario l'accesso tramite credenziali.

Quando il server genera il token, questo viene inviato al client che lo memorizzerà su disco, nel caso abbiano lunga durata permettendo quindi l'accesso al sito senza l'inserimento di credenziali, oppure verrà memorizzato in memoria e durerà per il tempo di navigazione all'interno del sito.

Per proseguire con l'esplorazione del sito, durante le successive richieste all'API, verrà inviato al server il token appena ricevuto.

Il server, per ogni interfaccia, o rotta, che deve essere riservata, andrà a verificare la stringa appena ricevuta tramite quella segreta usata per la sua generazione. Nel caso in cui il token non risulti valido verrà inviata una risposta al client che lo forzerà ad accedere nuovamente tramite credenziali.

Questa scelta progettuale garantisce scalabilità maggiore e la possibilità di accedere da più dispositivi allo stesso utente.

DBMS e Sensori

Un sistema informativo si trova solitamente a gestire dati provenienti da varie fonti. Questi dati devono essere memorizzati e facilmente accessibili e manipolabili.

Queste funzionalità vengono garantite con l’utilizzo di una base di dati o Database.

Una base di dati è un insieme di tabelle correlate tra loro. Una tabella è una struttura dati composta da righe, dette tuple, e colonne, dette campi o attributi.

La gestione della base di dati, dal punto di vista software, è affidato ad un servizio in esecuzione sul server: il DBMS, Database Management System. Per questo progetto è stato scelto MySQL.

Come scritto nel nome, questo DBMS, si basa su SQL che è un linguaggio strutturato per interrogare le basi di dati.

Questo linguaggio ci permette di fare principalmente due operazioni:

- definizione e manipolazione della struttura delle tabelle, DDL (Data Definition Language)
- ricerca e manipolazione dei dati all’interno della base di dati, DML (Data Manipulation Language).

Lo strato di risorse non è composto solamente dal DBMS ma anche dalle altri componenti che forniscono dati al sistema come unica funzione. In questo progetto sono infatti presenti i sensori che vengono, per ovvi motivi, simulati tramite codice Python.

Come per React, anche Python, simulando i sensori, invia i dati al server mediante chiamate API al PHP.

Il funzionamento dettagliato di come vengono simulati i sensori in Python sarà effettuato successivamente.

Progettazione Database

Una base di dati segue un preciso ciclo di vita che ci permette di identificare le procedure da effettuare per ottenere un database a regola d’arte.

La parte del ciclo che sarà analizzata è quella di progettazione (concettuale, logica e fisica) preceduta da un attento studio di fattibilità e analisi dei requisiti.

Ogni fase della progettazione risulterà in uno schema logico che porterà, infine, all’implementazione sulla macchina del database.

Studio di fattibilità e analisi dei requisiti

Questa fase del ciclo di vita presuppone un'analisi della richiesta.

[...]

Progettazione Concettuale

La progettazione concettuale è la prima delle sottofasi della progettazione e consiste nella formalizzazione dei requisiti per un successivo sviluppo dello schema ER, Entità-Relazione, e la descrizione delle Regole Aziendali, RA, necessarie per il completamento logico dello schema ER.

Con Regole Aziendali si definisce un documento che comprende tutti i vincoli non esprimibili nello schema ER, solitamente definiti da locuzioni come SI OTTIENE e SI DEVE.

Il modello ER è un modello che descrive i concetti indipendenti, Entità, e la loro correlazione, Relazione o Associazione. Sia Entità che Relazioni possono necessitare di informazioni aggiuntive dette Attributi.

Questo modello si basa su dei costrutti grafici che permettono di rappresentare visivamente lo schema.

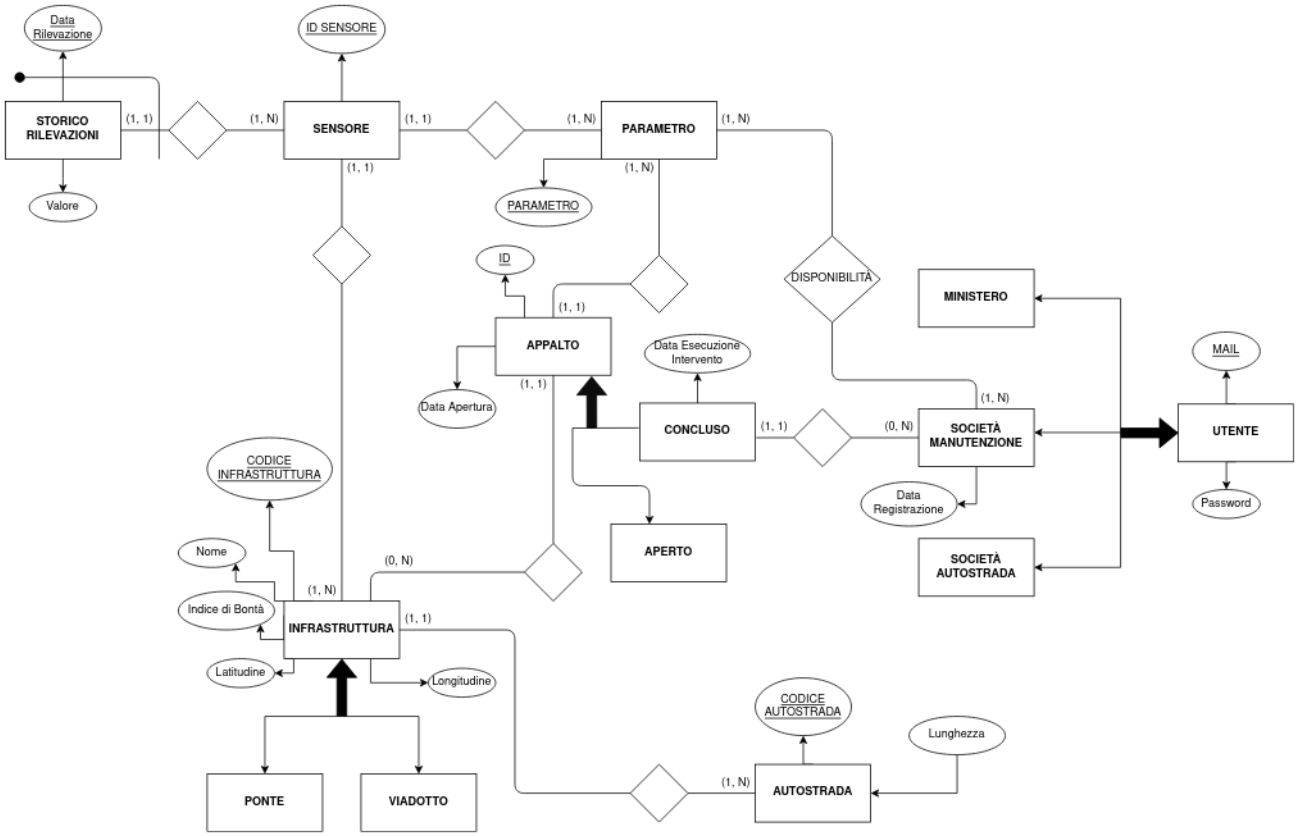

figura 4

Quello rappresentato in figura è il modello ER completo.

Analizzando attentamente lo schema si trovano una coppia di numeri ai lati di ogni relazione; questa coppia definisce la cardinalità di una relazione.

Il vincolo di cardinalità è un particolare vincolo di integrità delle basi di dati che determina la correlazione delle tuple di due entità all'interno di una relazione.

Il primo numero determina la copertura cioè l'obbligatorietà della presenza all'interno della relazione di ogni tupla dell'entità; può avere valori 0, facoltativo, e 1, obbligatorio.

Il secondo numero determina la cardinalità massima cioè il numero massimo di volte che la stessa tupla può essere presente nella relazione; assume valori 1 e N, non specificando un massimo che può quindi variare.
In questa relazione verranno analizzati nel dettaglio le tre entità più importanti:

- Storico Rilevazioni
- Infrastruttura
- Appalto

Storico Rilevazioni è l'entità che racchiude tutti i valori dei sensori.
Per identificare una singola rilevazione si utilizza il codice del sensore stesso, presente come chiave esterna dall'entità Sensore, e la data di rilevazione.
La data è considerata univoca perché i sensori inviano i dati al server una sola volta al giorno.
La valore della rilevazione è memorizzato nel campo Valore.
Infrastruttura è l'entità in cui vengono salvate le informazioni relative alle infrastrutture che sono divise in Ponti e Viadotti.
Ogni infrastruttura è identificata univocamente da un Codice numerico progressivo. Sono di interesse anche il Nome e le Coordinate di dove si trova.
Il parametro IndiceBontà è un valore compreso tra 0 e 100 che rappresenta lo stato generale dell'infrastruttura.
Questo valore viene calcolato come la media dell'ultima rilevazione di ogni sensore presente sul ponte o viadotto. Il server aggiorna questo numero ogni volta che un sensore invia un nuovo valore al database.
Appalto è l'entità che gestisce tutti gli appalti del sistema, si dividono in:

- Aperto, appalto che non è stato ancora assegnato ad una società di manutenzione
- Chiuso, identifica l'appalto che ha portato ad un intervento di manutenzione

Dell'appalto è di interesse la data di apertura ed è identificato tramite un id progressivo.
Nello specifico, l'Appalto Chiuso prevede anche una data di esecuzione dell'intervento e l'identificatore della società che l'ha effettuato.

Progettazione Logica

Alla fase di progettazione concettuale segue quella di progettazione logica.

Questa fase ha come scopo quello di trasformare lo schema ER in modello Relazionale.

Questo modello permette una agevole implementazione su MySQL.

Prima di realizzarlo però è indispensabile ristrutturare lo schema ER modificandolo a dovere per renderlo pronto per diventare modello Relazionale.

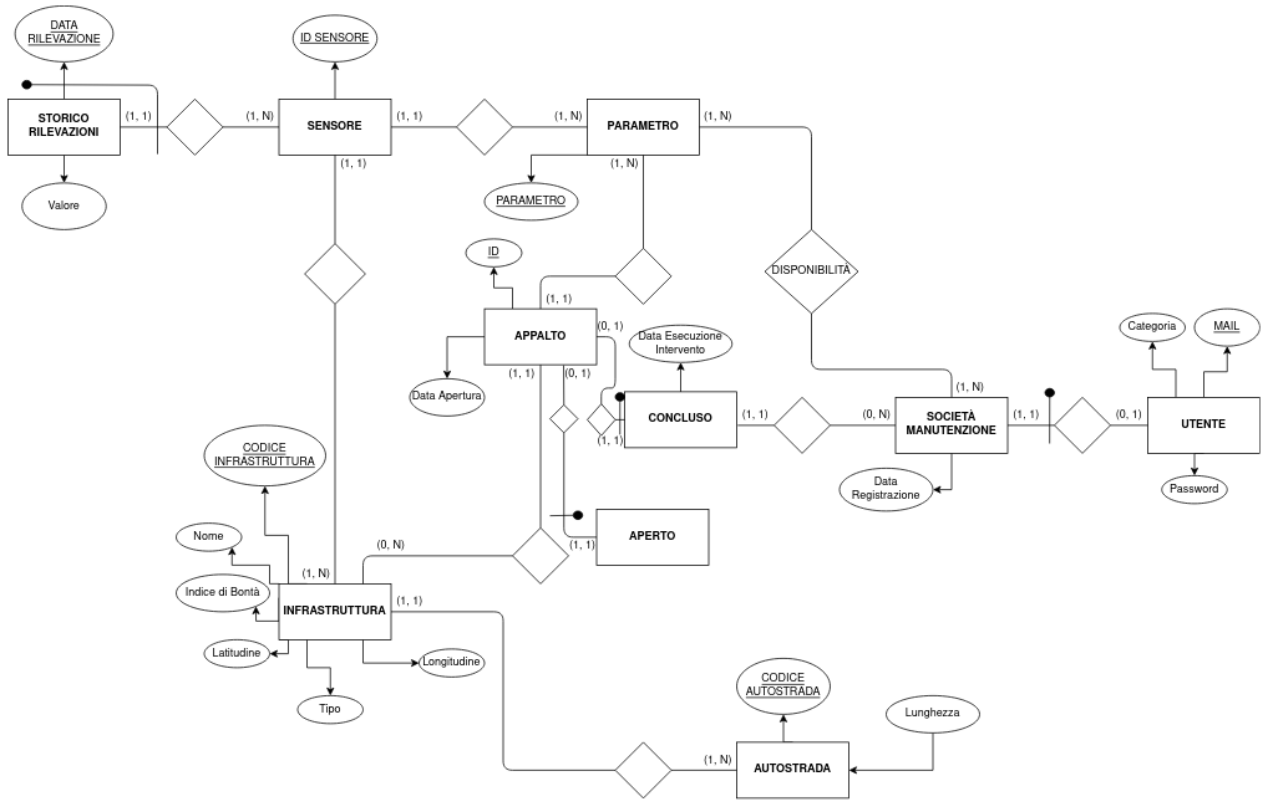

figura 5

Quello in figura è lo schema ristrutturato, le modifiche effettuabili sono quattro:

- Analisi delle ridondanze, porta alla rimozione di attributi ricavabili tramite inferenze statistiche e analizza gli indici di prestazione, volume dei dati e frequenza delle operazioni
- Accorpamento e separazione di concetti, porta alla creazione o rimozione di entità e relazioni
- Scelta degli identificatori e risoluzione degli attributi multivalore, solitamente le chiavi vengono scelte durante la fase di progettazione concettuale
- Eliminazione delle generalizzazioni, rimuove le generalizzazioni in vari modi possibili

Di queste operazioni di ristrutturazione, lo schema, ha avuto necessità della sola eliminazione delle generalizzazioni.

Le tre generalizzazioni presenti sono quelle che riguardano:

- Infrastruttura
- Appalto
- Utente

Quella di *Infrastruttura* è stata risolta eliminando le entità figlie e creando un nuovo campo nel padre che ne identifica il tipo (ponte o viadotto).

E' stata effettuata questa scelta in quanto le entità figlie non erano direttamente coinvolte in nessuna relazione o funzionalità.

La generalizzazione con *Appalto* viene invece risolta lasciando le due entità figlie e correlandole con il padre tramite una relazione.

Questo perché è presente una relazione specifica con l'Appalto Chiuso.

Infine quella con *Utente* viene risolta in modo parziale; collassano nel padre le entità Ministero e Società Autostrada tramite un attributo simile a come avviene su Infrastruttura, rimane l'entità Società Manutenzione tramite una relazione come avviene con Appalto.

Una volta ristrutturato lo schema ER si procede, tramite le regole di mappatura, alla stesura del modello Relazionale.

Seguendo le regole di mappatura è possibile trasformare lo schema ER in modello Relazionale.

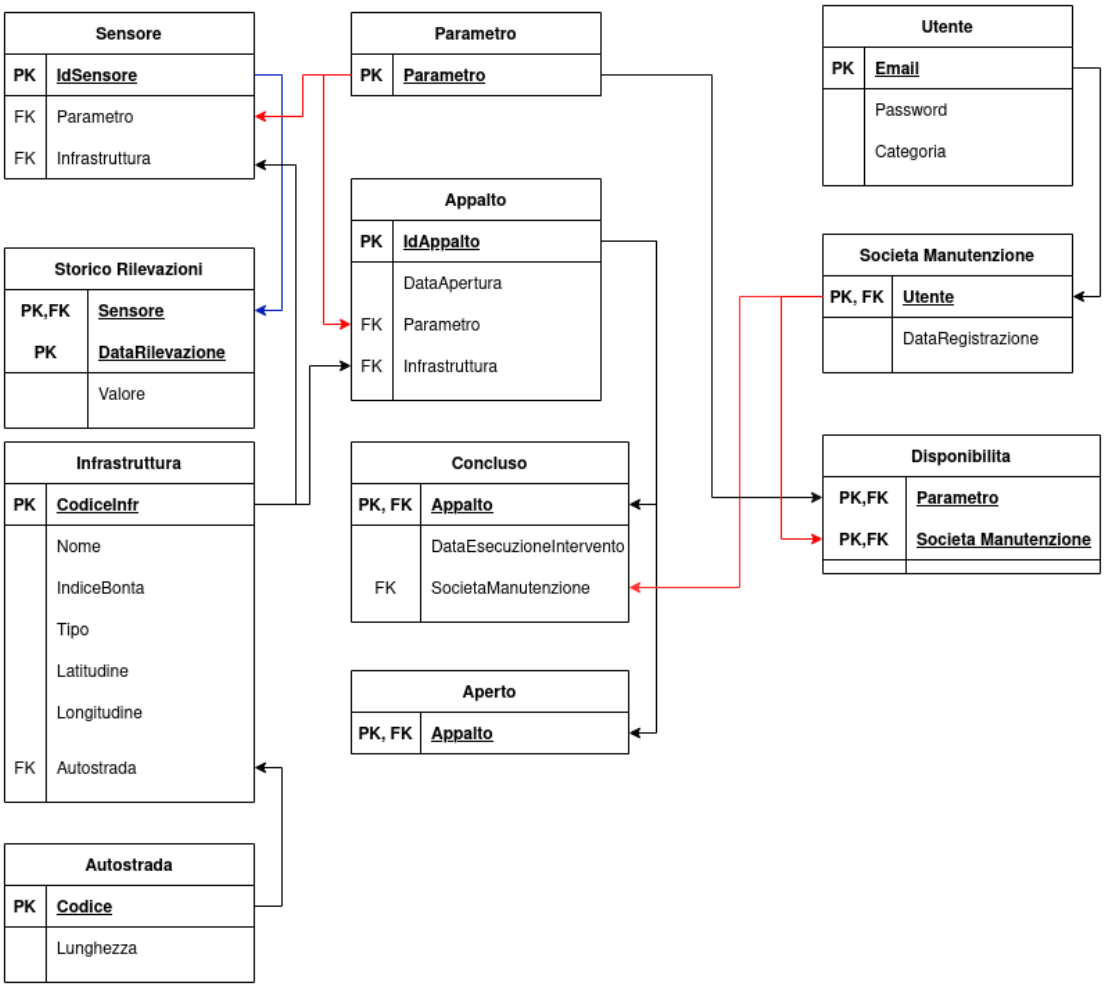

figura 6

In figura è possibile osservare il modello Relazionale. Sono presenti tutte le entità dello schema ER e la relazione Disponibilità; queste vengono chiamate tabelle.