



## Elaborato Esame di Stato

**a.s. 2020/2021**

Materie caratterizzanti:

*Informatica, Sistemi e Reti*

*“Sistema informatizzato di manutenzione  
per infrastrutture autostradali”*



*Viadotto Italia*

Docente Referente:

*prof. Giuseppe Scaranello*

Candidato:

*Lorenzo Bartolini*

# Indice

Indice	1
1. Abstract	2
2. Architettura Software	3
2.1 Frontend con React.js	4
2.2 Backend con API in PHP	5
2.3 DBMS e Sensori	6
3. Database	7
3.1 Studio di fattibilità e analisi dei requisiti	7
3.2 Progettazione Concettuale	8
3.3 Progettazione Logica	10
3.4 Progettazione Fisica	13
4. Sito Web e API php	16
4.1 Home Page	16
4.2 Registrazione	16
4.3 Accesso	18
4.4 Infrastrutture	19
4.5 Informazioni Infrastruttura	21
4.6 Mappa	23
4.7 Appalti	24
5. Schema di Rete	25
5.1 Rete Ministero	26
5.2 Rete Accentuatori e infrastrutture	27
5.2.1 Piano di Indirizzamento	28
5.2.2 Implementazione	29
5.3 Rete Sede Centrale	30
5.3.1 Piano di Indirizzamento	30
5.3.2 Implementazione	31
6. Simulazione Sensori	33
6.1 Analisi dei parametri della simulazione	35
7. Sensori Infrastrutturali	36
7.1 Ponte ad arco	37
7.2 Ponte strallato	37
7.3 Ponte a travi reticolari	37
7.4 Viadotto	37
8. Bibliografia e Sitografia	38

## **1. 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 a 100.

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

## 2. 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.

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

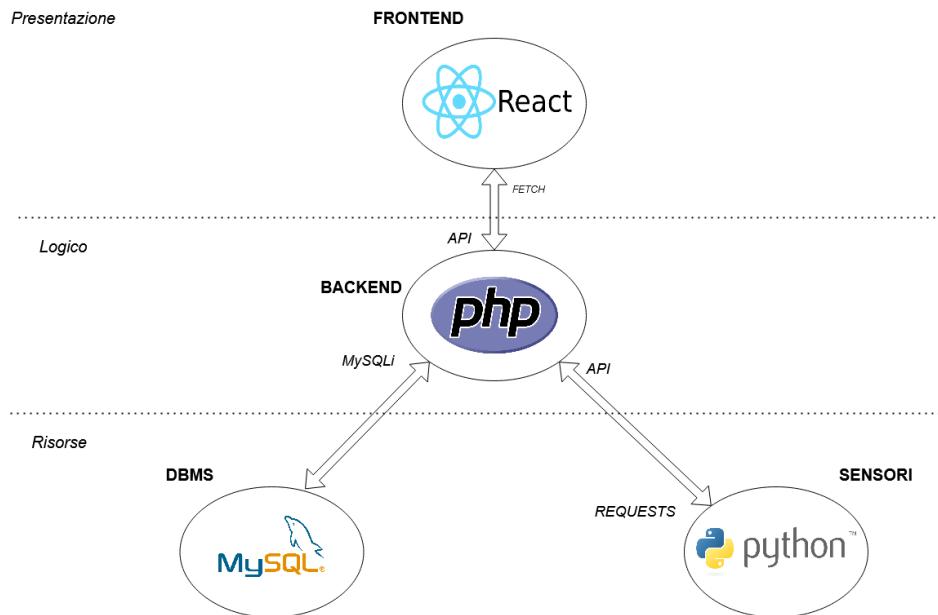


figura 2.1 - Architettura Software

Come è possibile vedere nell'immagine sopra, si identificano 4 diverse sezioni divise su 3 strati:

- Presentazione, o frontend
- Logica, o backend
- Risorse

La parte di Frontend è gestita tramite il framework React.

La parte di Backend è affidata al PHP.

Il DBMS è MySQL.

Infine la simulazione dei sensori è eseguita in Python e fa parte dello strato di Risorse.

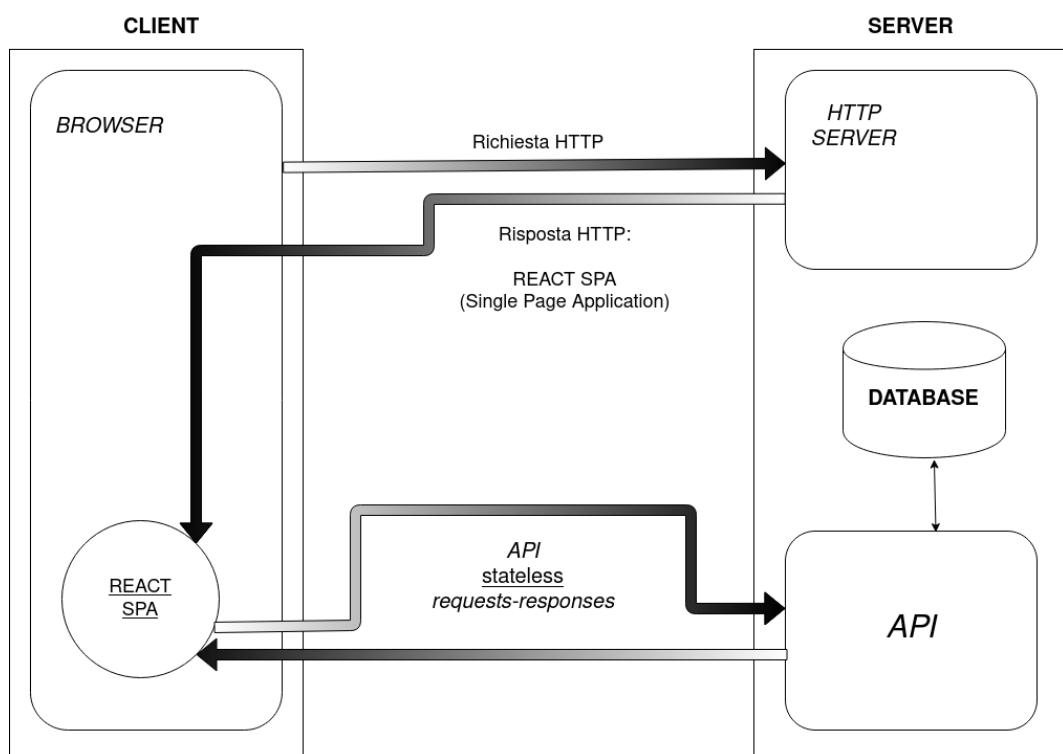
## 2.1 Frontend con React.js

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

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 2.2 - Client Server*

Nel momento in cui il browser invia la richiesta al server, questo restituisce al client una pagina web composta da HTML, CSS ed 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 il client 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 pacchetti HTTP contenenti dati in formato JSON e non HTML.

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

## 2.2 Backend con API in PHP

Per quanto riguarda lo strato di logica è stato deciso di codificarlo in linguaggio PHP mediante la realizzazione di API.

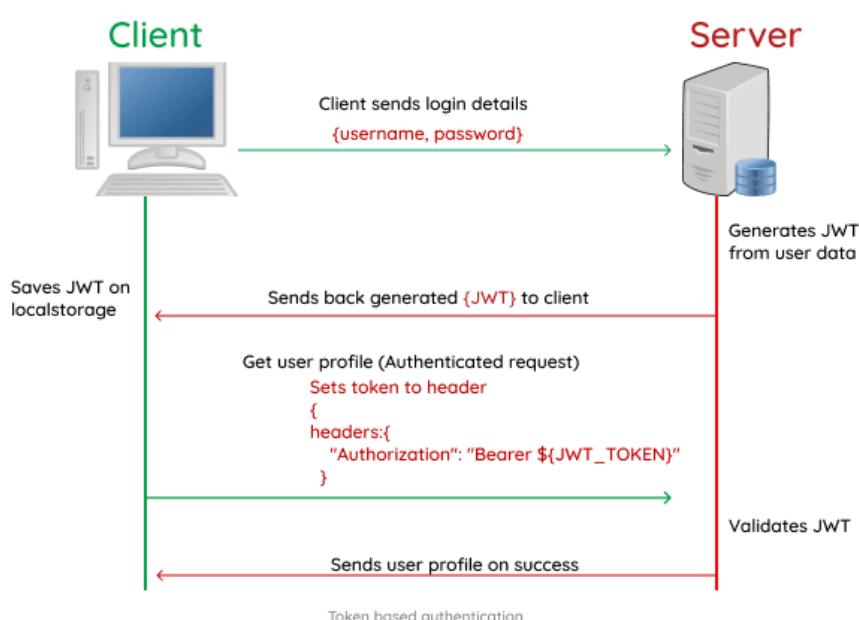
Un’API, Application Programming Interface, è un’interfaccia software che permetta lo scambio di dati tra un client ed un server.

Questi due metodi permettono la ricezione di dati dal server, GET, e la possibilità di inviare dati, POST.

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 2.3 - Autenticazione con Token*

Una volta completato l’accesso sarà predisposto un sistema di autenticazione per le richieste successive.

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

Quando il server genera il token 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.

## **2.3 DBMS e Sensori**

La gestione della base di dati è affidata a MySQL.

Lo strato di risorse è composto anche dalla simulazione dei sensori in 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.

## **3. Database**

### **3.1 Studio di fattibilità e analisi dei requisiti**

Le infrastrutture sono di due tipi:

- Ponti
- Viadotti

Ogni infrastruttura verrà monitorata da almeno un sensore.

Esistono tre tipi di utenti che possono accedere al sito:

- Utente del Ministero dei Trasporti
- Utente della Società Autostradale
- Utente della Società di Manutenzione

Ogni Società di Manutenzione è Disponibile in vari Parametri scelti durante la fase di registrazione.

### 3.2 Progettazione Concettuale

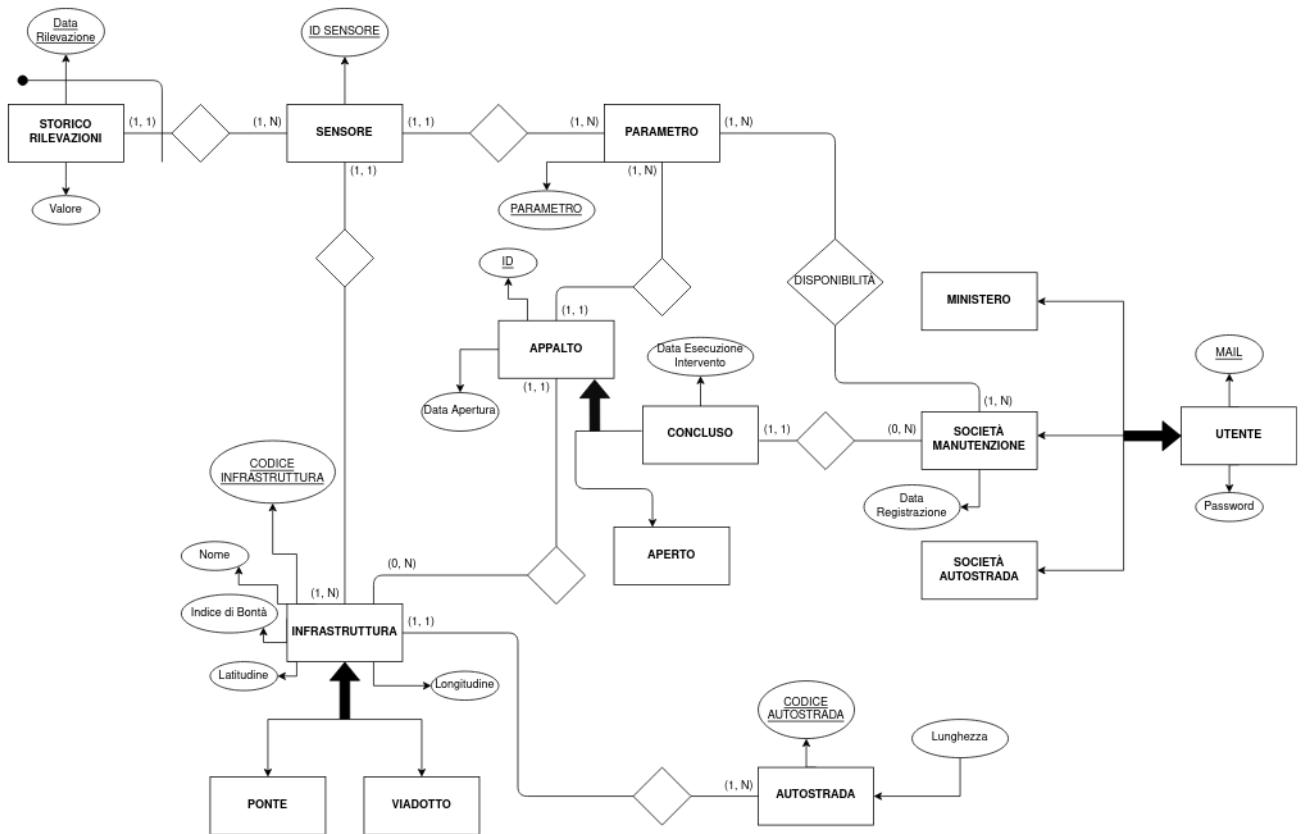


figura 3.1 - Schema Entità-Relazione

Quello rappresentato in figura è il modello E-R completo.

Sono 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 specializzate in:

- Ponti
- Viadotti

Le infrastrutture sono correlate all'autostrada di appartenenza.

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.

Le società di manutenzione sono correlate all’entità Parametro per tenere traccia della specializzazione di ogni società.

Per l’entità Utente sono di interesse la mail e la password.

Per garantire la sicurezza viene memorizzata la fingerprint della password tramite algoritmo BCRYPT.

Questo algoritmo genera un hash di 60 caratteri con prefisso ‘\$2y\$’.

### **Regole aziendali:**

Infrastruttura.Indicebontà SI OTTIENE calcolando la media dell’ultimo valore di ogni sensore

Infrastruttura.Indicebontà DEVE essere compreso tra 0 e 100

StoricoRilevazioni.Valore DEVE essere compreso tra 0 e 100

### 3.3 Progettazione Logica

Si procede ristrutturando lo schema E-R; in figura schema ristrutturato.

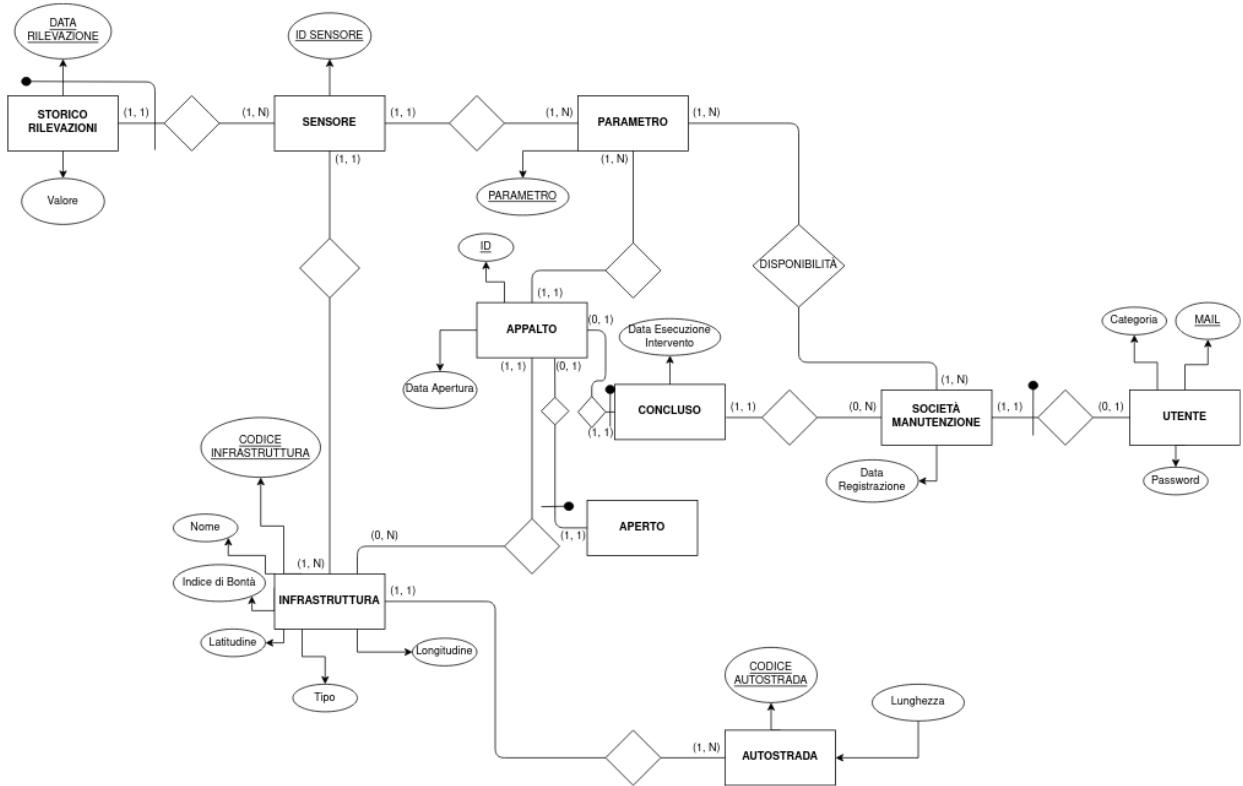


figura 3.2 - Schema ER Ristrutturato

La fase di ristrutturazione prevede quattro operazioni:

- Analisi delle ridondanze
  - In seguito ad un'attenta analisi delle prestazioni è stato deciso di mantenere l'attributo Indice di Bontà nell'entità Infrastruttura perché troppo dispendioso da calcolare ad ogni chiamata
- Accorpamento e separazione di concetti
  - Non necessaria
- Scelta degli identificatori e risoluzione degli attributi multivalore
  - Non necessaria
- 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.

La terza regola di mapping è già stata applicata.

Entità:

**Parametro** (Parametro)

**Autostrada** (Codice, Lunghezza)

**Utente** (Email, Password, Categoria)

**SocietàManutenzione** (Utente, DataRegistrazione)

V.I.R. [Utente con Utente.Email]

**Infrastruttura** (CodiceInfr, Nome, IndiceBonta, Tipo, Latitudine, Longitudine, *Autostrada*)

V.I.R. [Autostrada con Autostrada.Codice]

**Sensore** (IdSensore, *Infrastruttura*, Parametro)

V.I.R. [Infrastruttura con Infrastruttura.CodiceInfr]

V.I.R. [Parametro con Parametro.Parametro]

**StoricoRilevazioni** (Sensore, DataRilevazione, Valore)

V.I.R. [Sensore con Sensore.IdSensore]

**Appalto** (IdAppalto, DataApertura, Parametro, *Infrastruttura*)

V.I.R. [Infrastruttura con Infrastruttura.CodiceInfr]

V.I.R. [Parametro con Parametro.Parametro]

**AppaltoAperto** (IdAppalto)

V.I.R. [IdAppalto con Appalto.IdAppalto]

**AppaltoConcluso** (IdAppalto, DataEsecuzioneIntervento, SocietàManutenzione)

V.I.R. [IdAppalto con Appalto.IdAppalto]

V.I.R. [SocietàManutenzione con SocietàManutenzione.Utente]

Relazioni:

**Disponibilità (*Parametro*, *SocietaManutenzione*)**

V.I.R. [Infrastruttura con Infrastruttura.CodiceInfr]

V.I.R. [Parametro con Parametro.Parametro]

In seguito alla definizione del modello Relazionale verrà implementato tramite SQL.

### 3.4 Progettazione Fisica

Di seguito verranno descritte alcune query per la definizione della struttura di tabelle.

```
33  CREATE TABLE `Appalto` (
34    `IdAppalto` int(11) NOT NULL,
35    `DataApertura` datetime NOT NULL DEFAULT current_timestamp(),
36    `Parametro` varchar(32) NOT NULL,
37    `Infrastruttura` int(11) NOT NULL
38  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

figura 3.3 - Query di creazione tabella Appalto

Questa query permette la creazione di una tabella chiamata ‘Appalto’ con quattro campi:

- IdAppalto, intero chiave primaria
- DataApertura, data di apertura dell’appalto con valore di default alla data odierna
- Parametro, chiave esterna che indica il parametro dell’appalto
- Infrastruttura, chiave esterna che specifica l’infrastruttura interessata

```
466 ALTER TABLE `Appalto`
467   ADD CONSTRAINT `FK_Infrastruttura_Appalto` FOREIGN KEY (`Infrastruttura`) REFERENCES `Infrastruttura`(`CodiceInfr`)
468   ON DELETE CASCADE ON UPDATE CASCADE,
469   ADD CONSTRAINT `FK_Parametro_Appalto` FOREIGN KEY (`Parametro`) REFERENCES `Parametro`(`Parametro`)
470   ON DELETE CASCADE ON UPDATE CASCADE;
```

figura 3.4 - Vincoli di integrità referenziale FK

In figura invece è rappresentato il codice per la creazione dei vincoli di integrità referenziale tra i due campi della tabella Appalto e i loro corrispettivi nelle tabelle Infrastruttura e Parametro.

Sono stati configurati anche i comportamenti in caso di DELETE e UPDATE in Cascata rendendo più facile la modifica dei dati nel database.

```
124  CREATE TABLE `Infrastruttura` (
125    `CodiceInfr` int(11) NOT NULL,
126    `Nome` varchar(32) NOT NULL,
127    `IndiceBonta` float NOT NULL,
128    `Latitudine` double NOT NULL,
129    `Longitudine` double NOT NULL,
130    `Tipo` varchar(32) NOT NULL,
131    `Autostrada` varchar(4) NOT NULL
132  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

figura 3.5 - Query di creazione tabella Infrastruttura

In figura si osserva il codice, simile al precedente, per la creazione della tabella ‘Infrastruttura’.

Tutti i parametri sono impostati come NOT NULL perché obbligatori.

```
401 ALTER TABLE `Infrastruttura`  
402     ADD PRIMARY KEY (`CodiceInfr`),  
403     ADD KEY `FK_Autostrada` (`Autostrada`);
```

figura 3.6 - Chiave primaria per tabella Infrastruttura

Questo invece aggiunge il vincolo di chiave primaria all’attributo CodiceInfr e il vincolo di chiave semplice all’attributo Autostrada che verrà completato da una Foreign Key.

```
1 UPDATE Infrastruttura  
2 SET IndiceBonta=(  
3     SELECT AVG(Valore)  
4     FROM (  
5         SELECT a.Valore  
6         FROM (  
7             SELECT *  
8             FROM StoricoRilevazioni  
9             WHERE Sensore IN (  
10                SELECT IdSensore  
11                FROM Sensore  
12                WHERE Infrastruttura IN (  
13                    SELECT Infrastruttura  
14                    FROM Sensore  
15                    WHERE IdSensore='$IdSensore'  
16                )  
17            )  
18            ORDER BY DataRilevazione DESC) as a  
19            WHERE a.DataRilevazione=  
20                SELECT MAX(b.DataRilevazione)  
21                FROM StoricoRilevazioni AS b  
22                WHERE b.Sensore=a.Sensore  
23            )  
24            GROUP BY a.Sensore) as c  
25        WHERE CodiceInfr=[  
26            SELECT Infrastruttura  
27            FROM Sensore  
28            WHERE IdSensore='$IdSensore'  
29        ]
```

figura 3.7 - Query aggiornamento Indice di Bontà

In figura 10 è rappresentata la query che si occupa di aggiornare il valore ‘Indice di bontà’ di Infrastruttura.

Vengono trovati gli ultimi valori di ogni sensore dell’infrastruttura ed effettuata la media. Verrà poi sovrascritto il valore precedente dell’indice di bontà con la nuova media appena calcolata.

```
SELECT IdSensore
FROM Sensore
WHERE Infrastruttura IN (
    SELECT Infrastruttura
    FROM Sensore
    WHERE IdSensore='$IdSensore'
)
```

*figura 3.8 - Focus annidamento*

Questa è la query più annidata e seleziona gli Id dei sensori dell'infrastruttura analizzata partendo da un IdSensore.

```
SELECT AVG(Valore)
FROM (
    SELECT a.Valore
    FROM (
        SELECT *
        FROM StoricoRilevazioni
        WHERE Sensore IN (
            SELECT IdSensore
            FROM Sensore
            WHERE Infrastruttura IN (
                SELECT Infrastruttura
                FROM Sensore
                WHERE IdSensore='$IdSensore'
            )
        )
    )
    ORDER BY DataRilevazione DESC) as a
WHERE a.DataRilevazione=(
    SELECT MAX(b.DataRilevazione)
    FROM StoricoRilevazioni AS b
    WHERE b.Sensore=a.Sensore
)
GROUP BY a.Sensore) as c)
```

*figura 3.9 - Focus annidamento*

Successivamente vengono ordinati i valori in ordine decrescente e tramite un GROUP BY per il campo Sensore si calcola la media dei valori che rispettano la condizione di WHERE.

In questo caso l'obiettivo è fare la media degli ultimi valori di ogni sensore perciò si imporrà la condizione che la data di rilevazione sia uguale alla data più grande in cui ne è stata effettuata una, di fatto ottenendo l'ultima rilevazione.

Si procede modificando la tabella Infrastruttura aggiornando il valore Indice di Bontà.

## 4. Sito Web e API php

### 4.1 Home Page



*figura 4.1 - Home page*

Nella Home Page si viene accolti da una breve descrizione del portale.

Sulla sinistra è presente la barra di navigazione che permette di spostarsi all'interno del sito.

Inizialmente sono presenti i pulsanti per effettuare l'accesso e la registrazione.

### 4.2 Registrazione

The registration form is titled "Registrazione". It contains two input fields: "Email" and "Password". Below these, under the heading "Parametri di manutenzione:", are three checkboxes for "Elettricità", "Struttura", and "Asfalto". At the bottom left is a blue link "Torna al login" and at the bottom right is a purple "Registrati" button.

*figura 4.2 - Pagina di registrazione*

Questo è il form per effettuare la registrazione, effettuabile solamente da società di manutenzione ponti e viadotti.

Viene data la possibilità di scegliere uno o più parametri di manutenzione in cui è specializzata la società.

Una volta registrati, se la mail non è già stata registrata, viene fornita la possibilità di accedere al sito.

```
public function emailExist($email)
{
    $prep = $this->connessione->prepare("SELECT Email
                                         FROM Utente
                                         WHERE Email=?");

    $prep->bind_param("s", $email);
    $prep->execute();
    $result = $prep->get_result();

    $arr = [];

    while ($row = $result->fetch_assoc()) {
        $arr[] = $row;
    }

    if (sizeof($arr) > 0) {
        // email presente
        return true;
    }

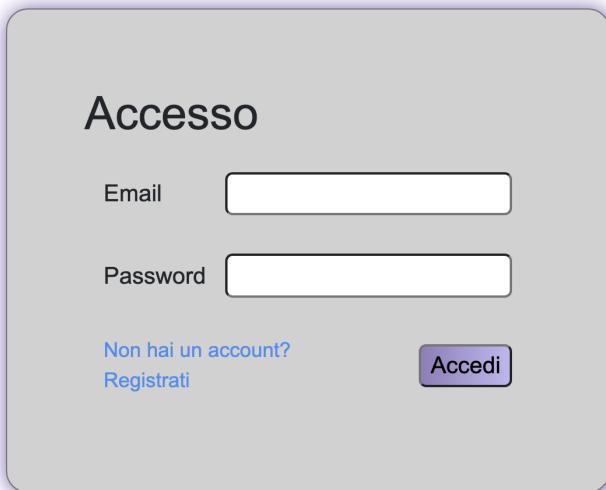
    // email non presente nel DB
    return false;
}
```

figura 4.3 - Controllo email nel database

Per verificare la presenza della mail nel database si cerca un utente con la stessa mail, se esiste allora non è possibile creare l'account.

Per proteggersi da SQL Injection viene utilizzata la funzione *prepare* di mysqli che permette di parametrizzare gli input inseriti dall'utente all'interno della query. Così facendo il dbms riconosce se la stringa ricevuta è conforme al tipo aspettato (string, int, float, ...) e se la stringa ricevuta contenga codice SQL iniettato.

## 4.3 Accesso



The image shows a login form titled "Accesso". It contains two input fields: "Email" and "Password", both represented by white rectangles with black outlines. Below these fields are two blue links: "Non hai un account? Registrati" and a purple button labeled "Accedi".

*figura 4.4 - Pagina di accesso*

Viene visualizzato un form simile per effettuare l'accesso.

In caso di accesso con credenziali corrette il server invierà al client un token con le modalità discusse precedentemente in *figura 3*.

Ci sono tre diverse categorie di utenti, queste portano a due interfacce diverse:

- Ministero e Società Autostradale, interfaccia completa



*figura 4.5 - Barra di navigazione Ministero e Autostrada*

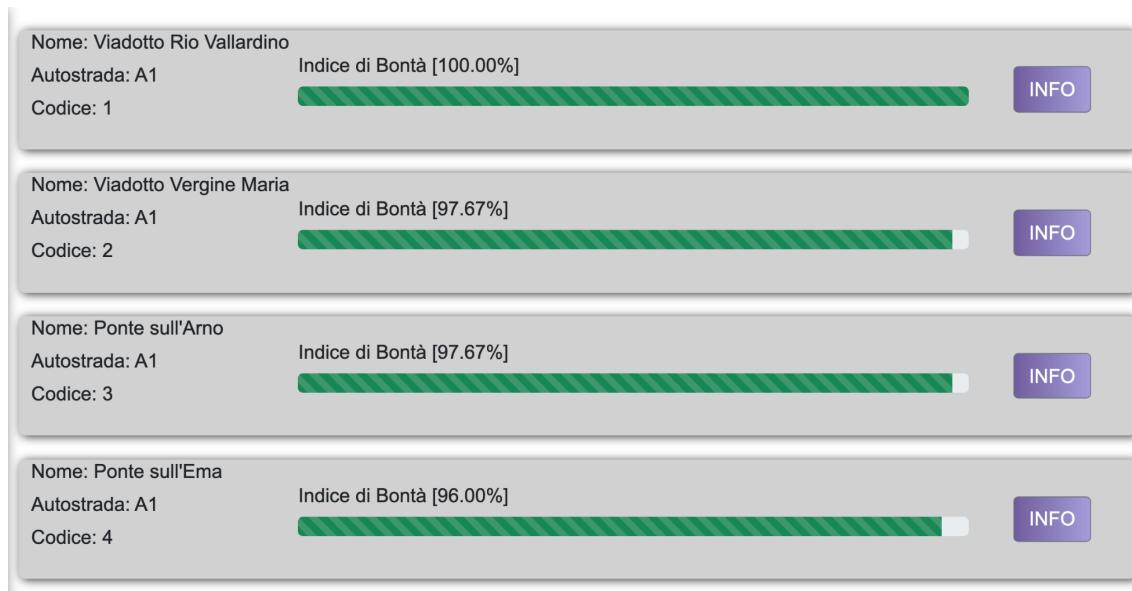
- Società di Manutenzione, interfaccia parziale



*figura 4.6 - Barra di navigazione Società di Manutenzione*

Come sarà visto successivamente verrà differenziata anche l'interfaccia tra utente Ministero e Società Autostradale in quanto il ministero potrà solo vedere le informazioni presenti sul sito mentre la Società Autostradale potrà Indire nuovi Appalti.

#### 4.4 Infrastrutture



*figura 4.7 - Pagina visualizzazione Infrastrutture*

In questa schermata sono visibili tutte le infrastrutture.

Sulla sinistra sono presenti delle informazioni di base mentre al centro si trova una barra dinamica che mostra l'indice di bontà. Quando il valore scende sotto una soglia cambia il colore per indicare pericolo. Sulla destra infine è presente un pulsante per ottenere le informazioni dettagliate passando alla pagina *infr-info*.

Per l’ottenimento delle informazioni da mostrare è necessaria una chiamata *fetch* all’API.

```
fetch(GlobalVar.urlAPI+'/infrastrutture.php', {
    method: 'GET',
    headers: {
        "Authentication": GlobalVar.token
    }
})
.then(response => {
    if(response.status == 200){
        return response.json();
    }else{
        setUser(null);
        GlobalVar.token = "";
        history.push("/");
        throw new Error;
    }
})
.then(data => {
    setInfr(data);
})
.catch(err => console.log(err));
```

figura 4.8 - *fetch infrastrutture*

Il metodo utilizzato è il GET e nell’header è aggiunto un parametro *Authentication* che sarà utilizzato dal server per autenticare l’utente attuale all’accesso ai dati richiesti.

In caso in cui il token non sia valido ci viene chiesto di rieffettuare l’accesso.

Altrimenti vengono salvate le informazioni all’interno della variabile di stato *infr*.

Le variabili di stato in React vengono utilizzate perché, modificandole viene renderizzata nuovamente la pagina con la possibilità di collegarci l’esecuzione di una funzione qualsiasi.

## 4.5 Informazioni Infrastruttura

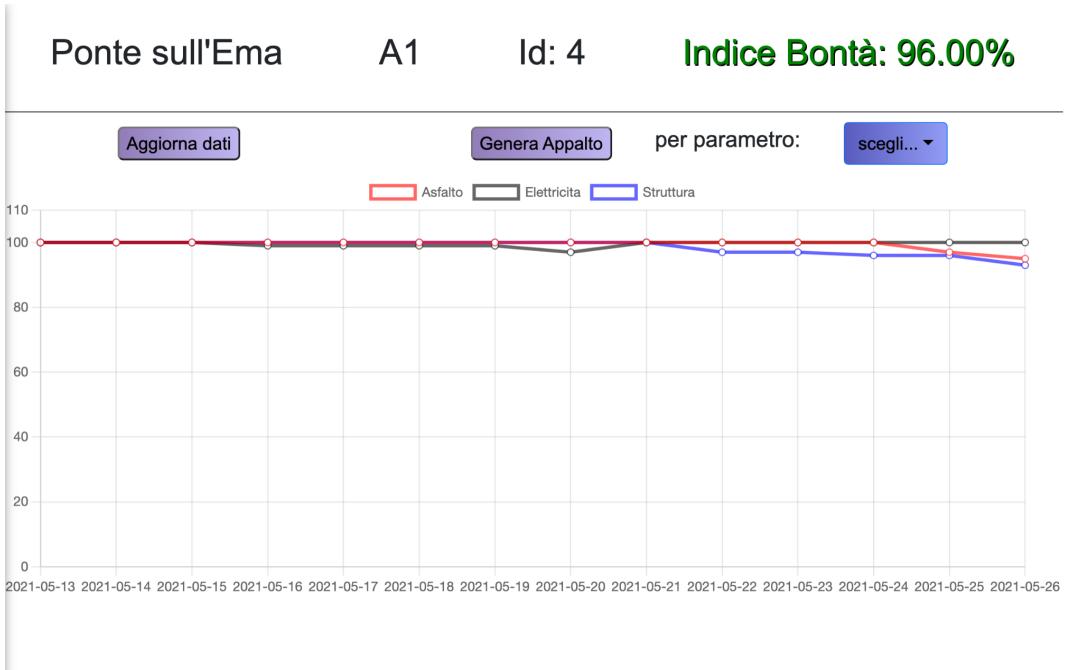


figura 4.9 - Pagina di informazioni dettagliate Infrastruttura

Dalla figura possiamo identificare due zone:

- Informazioni generiche, in alto
- Storico delle rilevazioni dei sensori, in basso

Sono presenti tre pulsanti che permettono all'operatore della società autostradale di indire un nuovo appalto per un preciso parametro e di aggiornare i dati mostrati sul grafico.

Il grafico mostra gli ultimi 15 valori in ordine temporale.

Il codice che prende questi valori è osservabile nella *figura 3.7*.

Ogni linea indica un parametro distinguibile dal colore anche grazie alla legenda sopra.

In caso di utente Ministero visualizzerà solamente il pulsante per aggiornare i dati.

Per la generazione di un nuovo appalto si effettuerà una chiamata all'API.

```
let to_send = {
    id: id,
    parametro: paramAttuale
};
```

fig 4.10 - Creazione oggetto to\_send

Prima però viene generato un oggetto che memorizzi due informazioni: il codice dell'infrastruttura e il parametro di cui effettuare la manutenzione.

```
fetch(GlobalVar.urlAPI+'/new-appalto.php', {
    method: 'POST',
    headers: {
        "Authentication": GlobalVar.token,
        "Content-Type": "application/json"
    },
    body: JSON.stringify(to_send)
})
.then(response => {
    if(response.status == 200){
        return response.json();
    }else{
        setUser(null);
        GlobalVar.token = "";
        history.push("/");
        throw new Error;
    }
})
.then(data => {
    if(data){
        alert("Appalto indetto con successo!");
    }else{
        alert("Impossibile indire appalto!");
    }
})
.catch(err => console.log(err));
```

figura 4.11 - indicuzione nuovo appalto

In caso di metodo POST, parametri vengono inseriti all'interno del body del pacchetto.

Viene mantenuta la gestione del token su ogni fetch effettuata che necessiti l'accesso da parte dell'utente.

## 4.6 Mappa

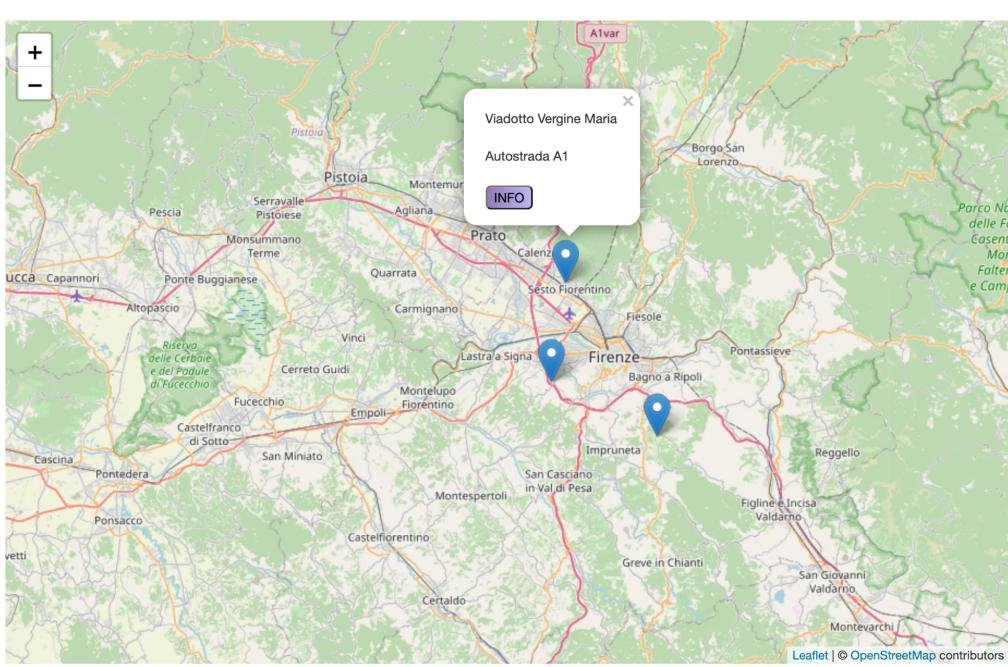


figura 4.12 - Mappa

Le infrastrutture possono essere visualizzate agevolmente sulla mappa grazie al marker posizionato tramite le coordinate esatte.

Cliccandolo si apre un Popup che permette di accedere alla pagina di informazioni dettagliate precedentemente osservata.

La mappa sfrutta un servizio gratuito per la visualizzazione della stessa: OpenStreetMap.

```
[markers.map((marker) => {
  return (
    <Marker position={marker.Coordinate} key={marker.Id}>
      <Popup>
        <p>{marker.Nome}</p>
        <p>Autostrada {marker.Autostrada}</p>
        <button onClick={() => history.push('/infr-info/' + marker.Id)}>INFO</button>
      </Popup>
    </Marker>
  )
})]
```

figura 4.13 - Marker con Popup su mappa

Tramite il metodo map applicato all’array markers è possibile inserire nella pagina, nello specifico sulla mappa, un marker per ogni infrastruttura nel database. Le informazioni sulle infrastrutture vengono ottenute tramite una chiamata API al server. Sfruttando poi i singoli attributi come Nome e Coordinate è possibile posizionare nella mappa i marker e modificare il Popup con dati personalizzati.

Come è possibile vedere, React al posto di HTML use JSX che permette l'utilizzo di tag HTML standard insieme a dei custom tag creabili dai programmatori. Questa flessibilità è utile in quanto permette una facile lettura del codice ed il riutilizzo agevole di parti comuni.

## 4.7 Appalti



*figura 4.14 - Pagina visualizzazione Appalti*

In figura è osservabile la schermata di gestione degli appalti.

In particolare si tratta della schermata secondo l'utente Società di Manutenzione in quanto è presente il pulsante ‘Esegui Intervento’.

In caso venga premuto, il parametro interessato (in questo caso Struttura), verrà impostato al valore 100 significando il fatto che è avvenuta una manutenzione perciò lo stato è ottimo.

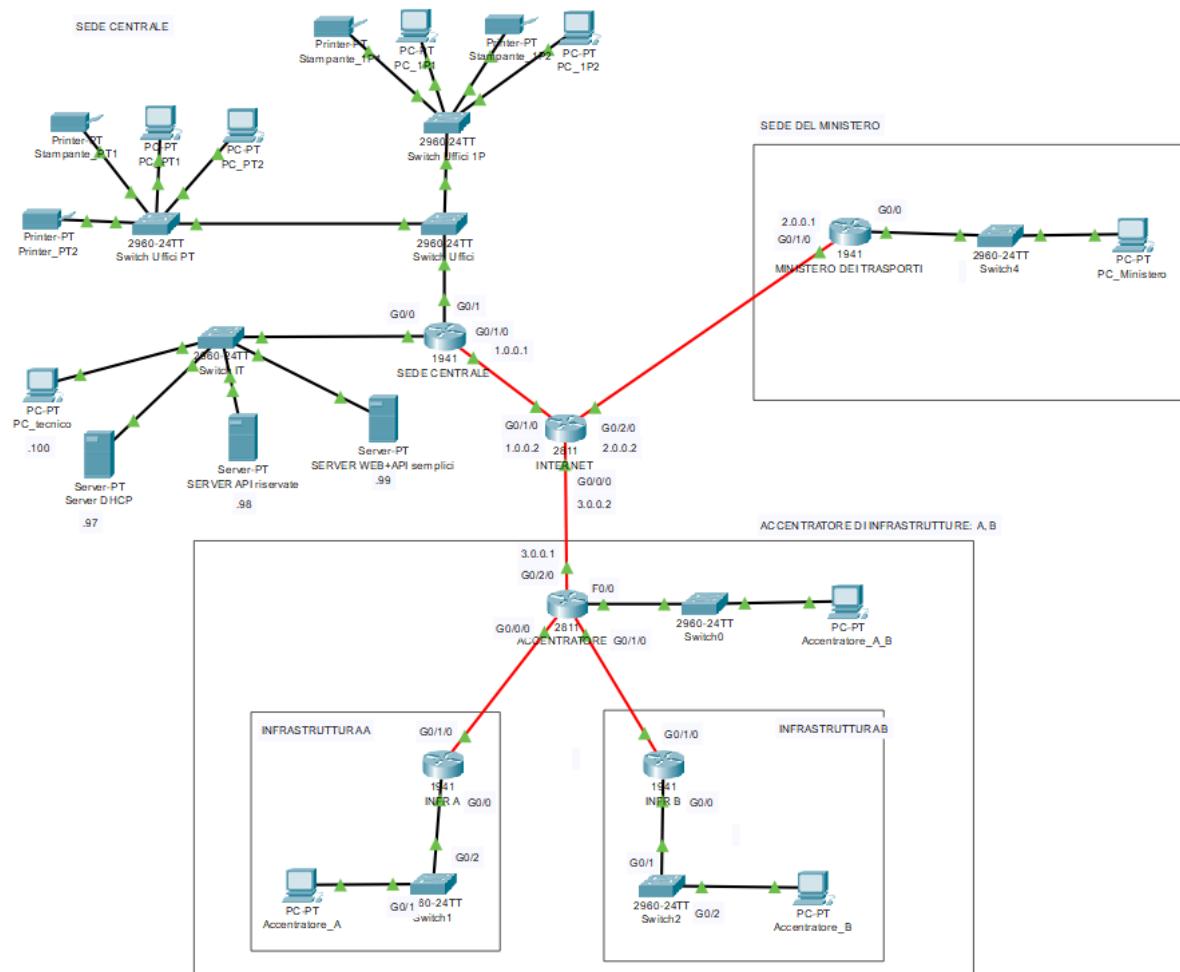
In alto si nota anche un filtro che permette all'utente di mostrare solo alcuni appalti basandosi sul Parametro.

## 5. Schema di Rete

Lo schema di rete prevede la progettazione considerando le richieste iniziali quali:

- Sensori posti su ogni ponte
- Presenza in vari punti della rete autostradale di postazioni di accentrimento dati
- Collegamento diretto e sicuro con il ministero dei trasporti

Ipotizzando che ogni ponte abbia più sensori dello stesso ‘Parametro’ e che questi sensori siano collegati a loro volta ad un accentratore di ponte si identifica il seguente schema di rete.



*figura 5.1 - Schema di rete completo*

Lo schema di rete è diviso in tre reti separate:

- Sede centrale
- Ministero
- Accentratore e infrastrutture

La Sede Centrale rappresenta la rete in cui sono collocati gli uffici della società e i server con il sito web accessibile dai manutentori e dal ministero.

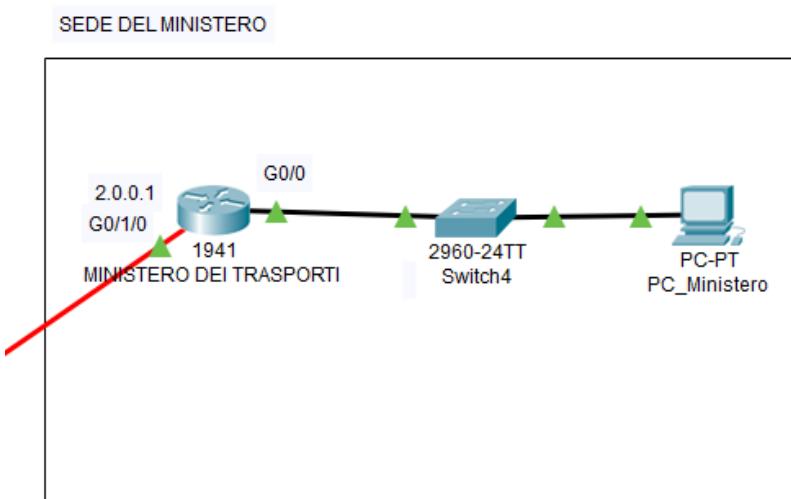
La rete del Ministero viene rappresentata con un singolo host in quanto non è di interesse la sua progettazione.

La rete dell'Accentratore e delle infrastrutture rappresenta la rete in cui sono presenti i dispositivi addetti alla ricezione dei dati dai sensori, alla loro manipolazione e all'invio al server centralizzato.

Viene simulato l'Internet tramite un router collegato in Punto a Punto alle diverse reti.

I collegamenti in rosso rappresentano un collegamento in Fibra Ottica. E' stato scelto questo canale trasmissivo perché la rete autostradale ha già predisposto una backbone in Fibra Ottica per tutta la lunghezza dell'autostrada.

## 5.1 Rete Ministero



*figura 5.2 - Focus rete Ministero*

La rete del Ministero è ipotizzata con indirizzo privato 192.168.0.0 e slash di rete /24 con la possibilità di indirizzare  $2^8$ , 256, host.

Viene predisposto un collegamento in VPN sicuro tra un singolo host della rete Ministero con la Sede Centrale. Questo host otterrà l'indirizzo ip 192.168.0.101 appartente alla sottorete IT della sede centrale.

La VPN è configurata come client-to-gateway, questa configurazione permette all'host di collegarsi a piacimento tramite l'utilizzo di credenziali fornite dall'amministratore di rete.

Una volta autenticato verrà instaurato un tunnel tra i due router coinvolti che procederà criptando i messaggi in uscita dall'host tramite protocollo IPSEC.

Questo protocollo è stato configurato con:

- Algoritmo crittografico - AES
- Algoritmo di Hashing - SHA

Il canale trasmissivo verrà utilizzato dal Ministero per consultare i dati direttamente nel Database e per accedere alle interfacce API riservate del server web.

Deve essere prima configurato il router della Sede Centrale per abilitarne il servizio VPN.

Per la configurazione su Packet Tracer è necessario configurare il protocollo ISAKMP che si occuperà di negoziare la connessione.

```
crypto isakmp policy 10
  encr aes
  authentication pre-share
  group 2
  lifetime 3600
```

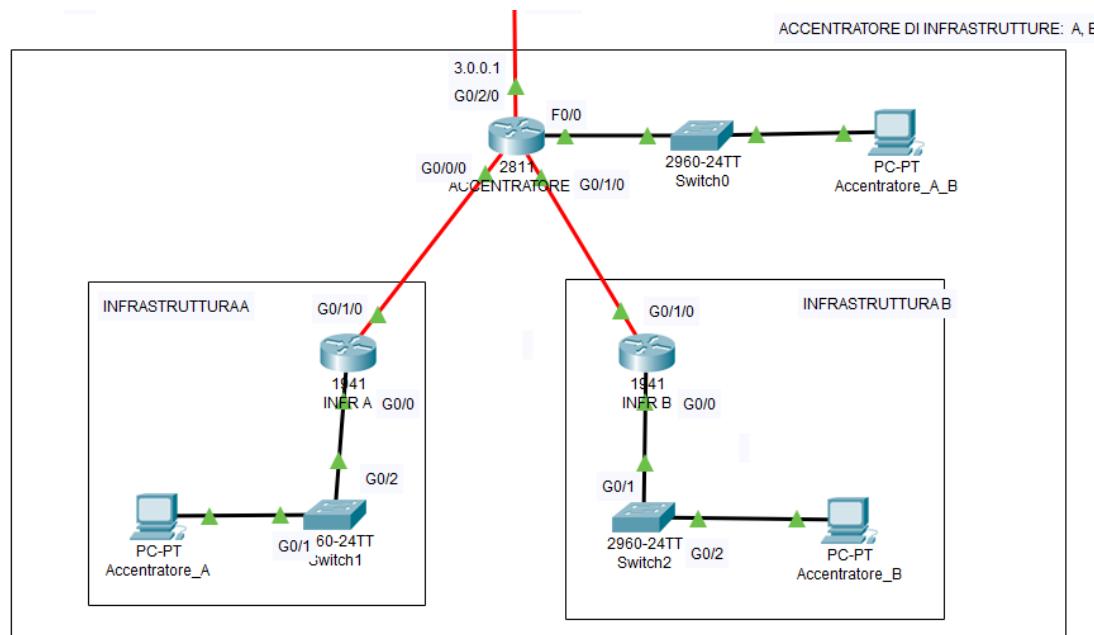
*figura 5.3 - Configurazione protocollo ISAKMP*

Successivamente viene configurato il protocollo IPSEC per la comunicazione sicura tra i peer connessi.

Viene definito un pool di indirizzi privati disponibili da associare agli host in collegamento VPN

Si procede creando uno username e password utilizzati per il collegamento. Infine viene effettuato un binding tra tutte le regole definite assegnandole ad una mappa crittografica applicata successivamente all’interfaccia outside del router della Sede Centrale.

## 5.2 Rete Accentuatori e infrastrutture



*figura 5.4 - Focus rete Accentuatori*

Per questa rete deve essere predisposto un canale trasmissivo sicuro tra l'host facente parte della sottorete Accentratore e la Sede Centrale.

### 5.2.1 Piano di Indirizzamento

Indirizzo di partenza: 192.168.0.0

Sottoreti:

- Accentratore
- Infrastruttura A
- Infrastruttura B
- PaP A-GW
- PaP B-GW

Sottorete	Host	Stampanti	Ampliamenti Futuri	IP Riservati	Fabbisogno	Slash di sottorete
<b>Accentratore</b>	1	0	0	3 (This net, Gateway, Broadcast)	4 ip	/30, h=2
<b>Infrastruttura A</b>	1	0	0	3	4 ip	/30
<b>Infrastruttura B</b>	1	0	0	3	4 ip	/30
<b>PaP A-GW</b>	2	0	0	2 (This net, Broadcast)	4 ip	/30
<b>PaP B-GW</b>	2	0	0	2	4 ip	/30

Complessivo

Fabbisogno: 4 (Accentratore) + 4 (Infr A) + 4 (Infr B) + 4 (PaP A-GW) + 4 (PaP B-GW) = 20 ip

Slash di Rete: /27, h=5

Indirizzo: 192.168.0.0/27

Subnet Mask: 255.255.255.224

Assegnazione IP:

Sottorete	This net	Host Min	Host Max	Gateway	Broadcast
<b>Accentratore</b>	192.168.0.0	192.168.0.1	//	192.168.0.2	192.168.0.3
<b>Infrastruttura A</b>	192.168.0.4	192.168.0.5	//	192.168.0.6	192.168.0.7
<b>PaP A-GW</b>	192.168.0.8	192.168.0.9	192.168.0.10	//	192.168.0.11
<b>Infrastruttura B</b>	192.168.0.12	192.168.0.13	//	192.168.0.14	192.168.0.15
<b>PaP B-GW</b>	102.168.0.16	192.168.0.17	192.168.0.18	//	192.168.0.19

### 5.2.2 Implementazione

Deve essere garantita la comunicazione tra l'host delle due infrastrutture con l'accentratore. Perciò sono configurate, sui router intermedi, delle rotte statiche per permettere al pacchetto di raggiungere la destinazione.

Deve essere negata la possibilità degli host delle infrastrutture di comunicare con l'esterno e tra di loro. Questa problematica viene risolta tramite l'implementazione di ACL che filtrano il traffico in ingresso alle interfacce del Router di ogni infrastruttura eliminando i pacchetti che non hanno come ip di destinazione un ip della sottorete Accentratore.

Viene configurato, come per la rete del Ministero, un collegamento client-to-gateway VPN tra l'accentratore e la Sede Centrale. L'IP assegnato all'host farà parte della sottorete specificatamente progettata, VPN Accentratori.

```
# ip nat inside source list 1 interface GigabitEthernet0/2/0 overload
# access-list 1 permit 192.168.0.0 0.0.0.3
```

Per garantire la comunicazione verso l'esterno è implementato l'NAPT per la traslazione dell'IP e porta sorgente dei pacchetti fuori dalla rete. In questo modo è consentito l'utilizzo di questo servizio da parte della sola sottorete Accentratore.

```
# access-list 110 permit ip host 192.168.0.5 host 192.168.0.1
# <access-list 110 deny ip any any>
# ip access-group 110 in
```

Tramite queste ACL è possibile vietare la comunicazione verso l'esterno da parte degli host di Infrastruttura A. La seconda riga è impostata di default.

Con ip access-group si indica la direzionalità dei pacchetti in transito su una interfaccia che devono essere verificati dalla ACL 110.

Viene implementata specularmente per l'host di Infrastruttura B modificando l'ip dell'host sorgente dell'ACL.

### 5.3 Rete Sede Centrale

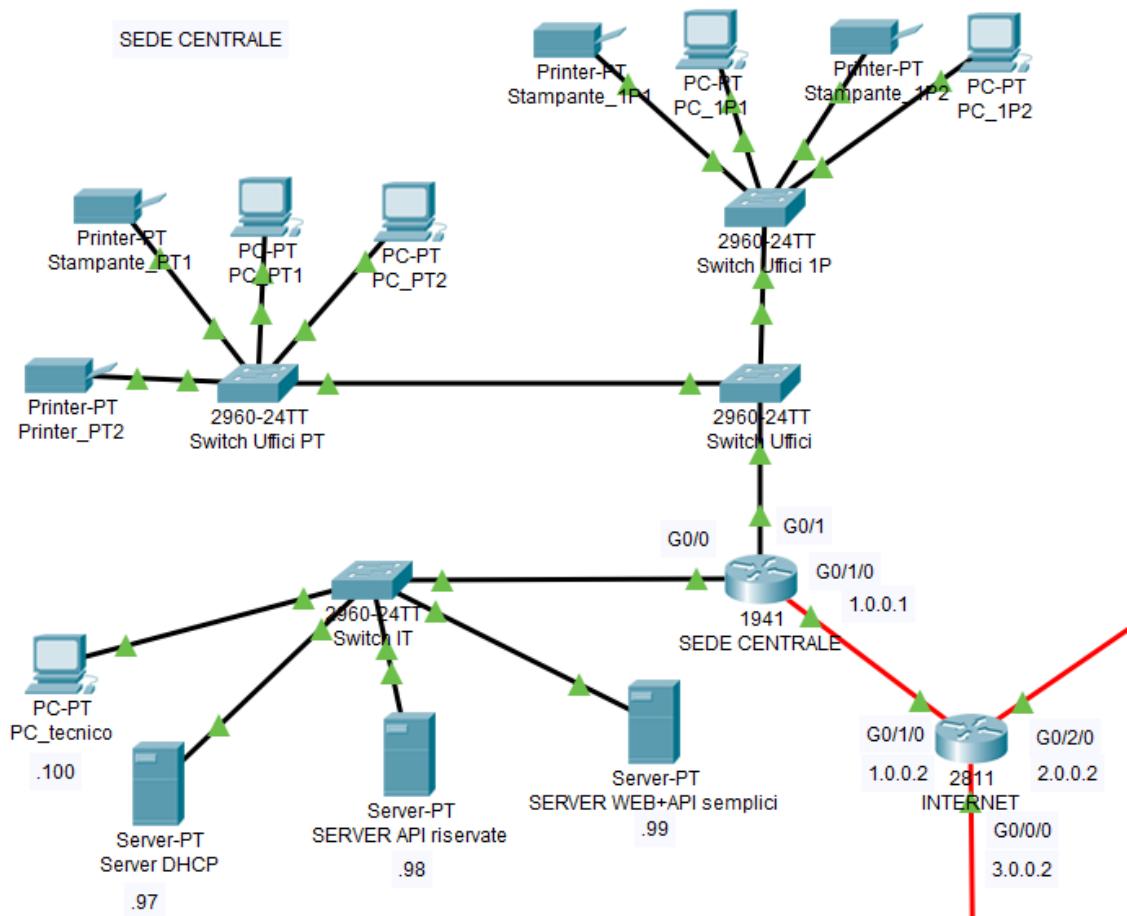


figura 5.5 - Focus Sede Centrale

#### 5.3.1 Piano di Indirizzamento

Indirizzo di partenza: 192.168.0.0

Sottoreti:

- Reparto IT
- Uffici PT (Piano terra)
- Uffici 1P (primo piano)
- VPN Accentuatori

Sottorete	Host	Stampanti	Ampliamenti Futuri	IP Riservati	Fabbisogno	Slash di sottorete
<b>Reparto IT</b>	5	0	0	3 (This net, Gateway, Broadcast)	8 ip	/29, h=3
<b>Uffici PT</b>	20	2	5	3	30 ip	/27, h=5
<b>Uffici 1P</b>	20	2	5	3	30 ip	/27
<b>VPN Accentuatori</b>	10	0	15	3	28 ip	/27

### Complessivo

Fabbisogno: 32 (Uffici PT) + 32 (Uffici 1P) + 32 (VPN Accentuatori) + 8 (Reparto IT) = 104 ip

Slash di Rete: /25, h=7

Indirizzo: 192.168.0.0/25

Subnet Mask: 255.255.255.128

### Assegnazione IP:

Sottorete	This net	Host Min	Host Max	Gateway	Broadcast
<b>Uffici PT</b>	192.168.0.0	192.168.0.1	192.168.0.29	192.168.0.30	192.168.0.31
<b>Uffici 1P</b>	192.168.0.32	192.168.0.33	192.168.0.61	192.168.0.62	192.168.0.63
<b>VPN Accentuatori</b>	192.168.0.64	192.168.0.65	192.168.0.93	192.168.0.94	192.168.0.95
<b>Reparto IT</b>	192.168.0.96	192.168.0.97	192.168.0.101	192.168.0.102	192.168.0.103

### 5.3.2 Implementazione

La Sede Centrale deve essere raggiungibile dall'esterno per quanto concerne la fruizione del servizio WEB.

Come si può vedere, nella sottorete IT sono presenti 3 server:

- Server DHCP
- Server API Riservate
- Server WEB+API Semplici

Il servizio API è diviso su due server (nella realtà basta che venga separato il servizio in ascolto su due porte diverse ma sulla stessa macchina), il server con API Riservate deve essere protetto e accessibile solo dall'amministratore della società autostradale, dal Ministero e dai singoli accentuatori sparsi per la regione.

Al contrario il server Web con API Semplici è accessibile semplicemente tramite IP pubblico configurando il port forwarding.

Il port forwarding permette infatti di inoltrare determinati pacchetti in arrivo sul router verso il server web.

Per negare l'accesso al server protetto vengono configurate delle ACL.

I due uffici, primo piano e piano terra, sono il ramo più sicuro della rete e non possono essere contattati direttamente e non possono comunicare con la sottorete IT.

Identifichiamo nella sottorete IT la DMZ della rete. La DMZ è il ramo più esposto della rete in cui le regole di filtraggio pacchetti sono meno stringenti, solitamente ci si trovano i server.

```
# ip nat inside source static tcp 192.168.0.99 80 1.0.0.1 80
```

Con questo comando viene configurato il port forwarding sul router permettendo di raggiungere il server web attraverso un inoltro dei pacchetti da parte del router arrivati sull'interfaccia outside con porta di destinazione 80 (standard protocollo http).

```
# access-list 111 deny ip 192.168.0.32 0.0.0.31 192.168.0.96 0.0.0.7  
# access-list 111 permit ip 192.168.0.32 0.0.0.31 any  
# access-list 110 deny ip 192.168.0.0 0.0.0.31 192.168.0.96 0.0.0.7  
# access-list 110 permit ip 192.168.0.0 0.0.0.31 any
```

Viene così negato l'accesso alla sottorete Reparto IT da parte degli uffici ad entrambi i piani.

Vengono applicate all'interfaccia che fa da Gateway per le sottoreti degli uffici per ottimizzare le prestazioni della rete. Si preferisce infatti applicare le regole di *deny* nel punto più vicino possibile alla sorgente mentre di *permit* più vicino alla destinazione.

## 6. Simulazione Sensori

La simulazione dei sensori in Python ha come obiettivo quello di inviare una volta al giorno un nuovo valore ad server simulando il campionamento di un vero sensore.

Come prima cosa viene contattato l'API per richiedere la lista di tutti i sensori presenti nel sistema. Per ognuno vengono ottenute successivamente gli ultimi N valori in ordine temporale.

E' stato sviluppato un algoritmo di 7 punti da seguire per la generazione del valore successivo applicato ad ogni sensore presente.

### Fase 1:

Vengono presi gli ultimi N valori presenti sul database, punti\_momentum; se non ne sono presenti abbastanza il nuovo valore sarà di default 100, cioè il massimo possibile.

Altrimenti si procede con le fasi successive.

### Fase 2:

Viene calcolata la derivata dei punti acquisiti a coppie di due punti. Si procede calcolandone la media per ottenere un andamento medio negli ultimi N giorni.

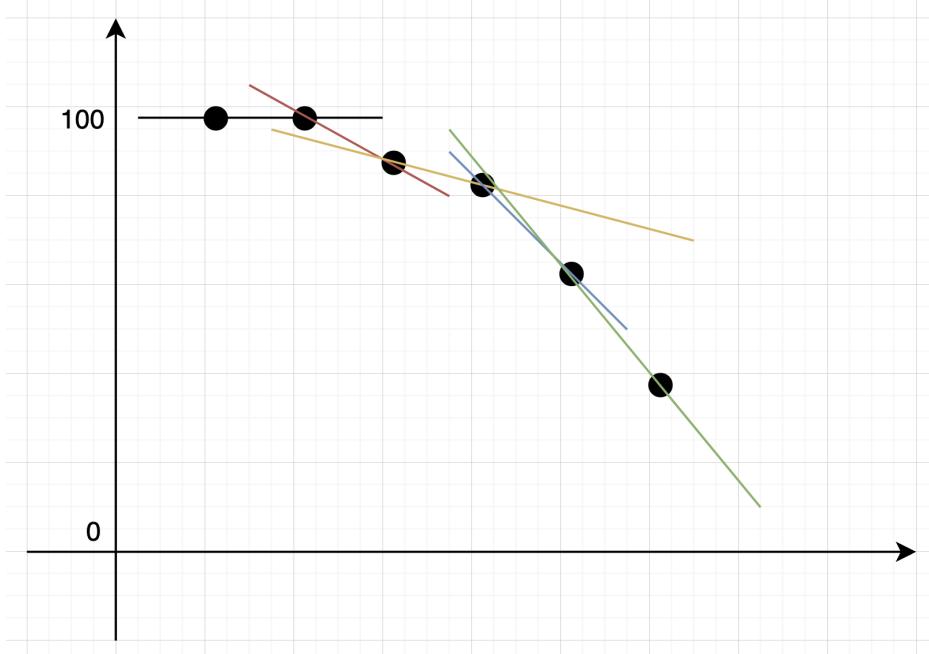


figura 6.1 - Derivata valori passati

### Fase 3:

Se questo valore, definito come ‘Momentum’, risulta positivo significa che c'è stato un intervento di manutenzione recentemente perciò viene ritornato il valore 100; l'idea alla base di questa scelta è che se è appena stato eseguito un intervento è molto improbabile quindi assumiamo come impossibile che avvenga un ulteriore guasto o rottura.

Altrimenti, se il momentum è minore o uguale a zero, c'è un trend negativo e si procede con le successive fasi.

**Fase 4:**

Viene calcolata la probabilità di stazionare, `prob_base_stazionare`, cioè che il nuovo valore sia uguale al precedente.

La probabilità minima è del 20% partendo da un 80% come valore massimo.

Il valore massimo può variare in funzione del momentum appena calcolato per un massimo del 40%.

**Fase 5:**

Basandosi sulla probabilità calcolata si determina se il nuovo valore deve diminuire o stazionare.

Se deve stazionare viene ritornato l'ultimo valore altrimenti si prosegue con le prossime fasi.

**Fase 6:**

Si procede calcolando un offset dal quale il nuovo valore potrà partire.

Il momentum determina l'offset con la seguente formula:

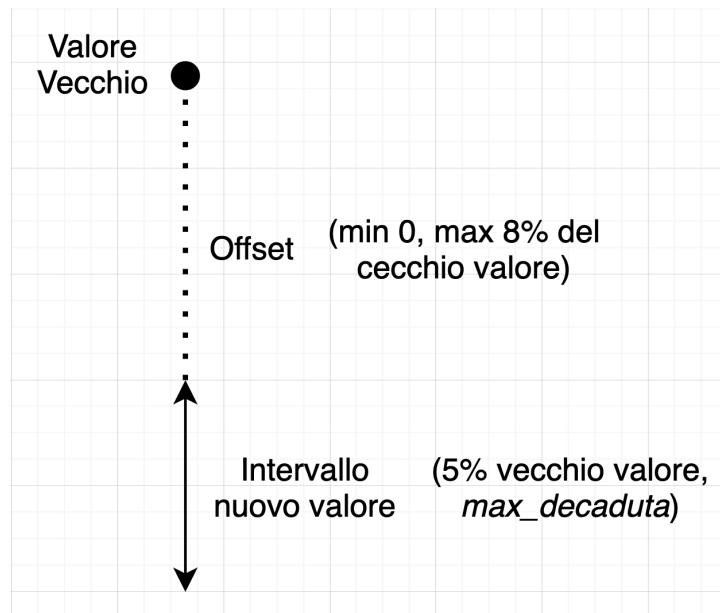
$$\text{offset} = (\min(\text{momentum}, 4) * 2) / 100$$

*figura 6.2 - Formula per calcolare l'offset*

Nella formula il minimo viene calcolato per impostare un massimo a 4 per il momentum evitando che diventi troppo grande. Infatti nel caso peggiore l'offset sarà l'8% dell'ultimo valore.

**Fase 7:**

Viene infine scelto casualmente il nuovo valore appartenente al seguente intervallo la cui dimensione è prefissata, `max_decaduta`.



*figura 6.3 - Visualizzazione intervallo nuovo valore*

## **6.1 Analisi dei parametri della simulazione**

*prob\_base\_stazionare*

- modificando questo parametro si rende l'infrastruttura più facilmente soggetta a decadimento

*max\_decaduta*

- indica la gravità della ricaduta delle infrastrutture; un valore troppo alto rende facile scendere sotto la soglia limite di pericolo

*punti\_momentum (N)*

- un valore basso rende estremamente instabile la simulazione allontanandosi dalla realtà osservabile
- un valore alto fa sì che ogni valore passato influenzi quello successivo con il risultato che la simulazione genererà una curva più docile avvicinandosi alla realtà

## 7. Sensori Infrastrutturali

Nel progetto sono stati simulati i sensori semplificandoli ad un valore tra 0 e 100.

Durante la fase di costruzione di un ponte vengono installati i seguenti sensori:

- Celle di carico 

  - Sensori che controllano il bilanciamento del carico della trave sui piloni

- Barrette estensimetriche 

  - sensori che misurano lo stato tensionale della struttura
  - vengono inseriti in vari punti del ponte in base alla tipologia di infrastruttura

- Piezometri elettrici 

  - sensori per il monitoraggio di dati ambientali come vento e precipitazioni atmosferiche

- Inclinometro fisso 

  - sensore in grado di misurare l'inclinatura e la rotazione

- Catena inclinometrica 

  - posizionata sulla spalla del ponte è in grado di misurarne il movimento

- Misuratore di giunti 

  - posizionato tra le giunture misura la grandezza della fessura presente

- Sensore di temperatura 
- Livellometro 

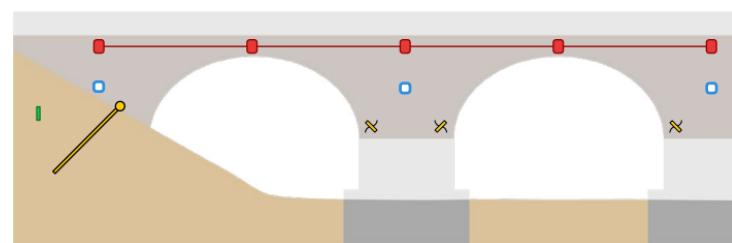
  - sensore capace di identificare l'inclinamento dell'intero ponte

Esistono quattro tipologie di ponti:

- ad arco
- strallati
- a travi reticolari
- viadotti

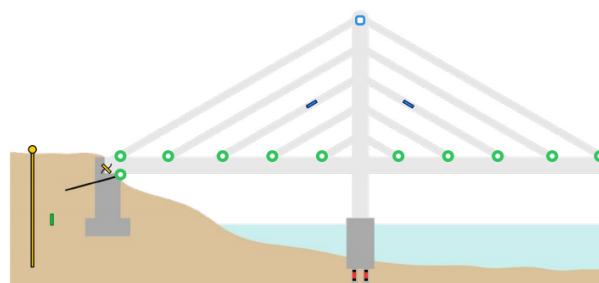
Per ognuno di questi tipi vengono scelti sensori diversi e posizionati diversamente per l'analisi di punti specifici.

## 7.1 Ponte ad arco



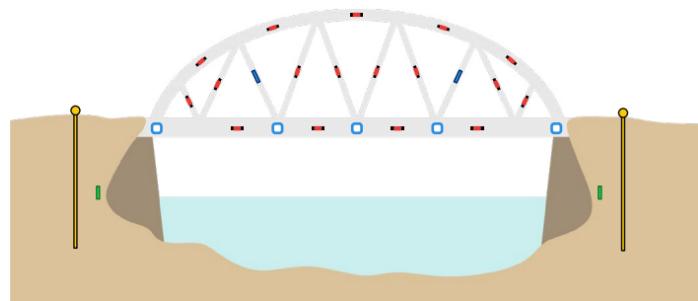
*figura 7.1 - Ponte ad arco*

## 7.2 Ponte strallato



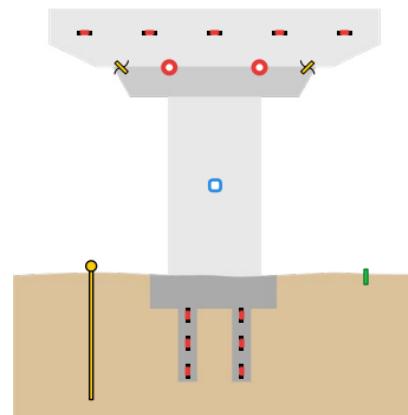
*figura 7.2 - Ponte strallato*

## 7.3 Ponte a travi reticolari



*figura 7.3 - Ponte a travi reticolari*

## 7.4 Viadotto



*figura 7.4 - Viadotto*

## **8. Bibliografia e Sitografia**

- Corso di Informatica VOL 3
  - autori: FORMICHI LORENZO e MEINI GIORGIO
  - editore: ZANICHELLI
- Sistemi e Reti VOL 3
  - autori: LUIGI LO RUSSO e ELENA BIANCHI
  - editore: HOEPLI
- Monitoraggio Ponti, SIM STRUMENTI
  - [https://simstrumenti.com/app\\_notes/Monitoraggio\\_ponti\\_SIM\\_STRUMENTI.pdf](https://simstrumenti.com/app_notes/Monitoraggio_ponti_SIM_STRUMENTI.pdf)
- Lista Ponti e Viadotti italiani
  - [http://www.autostrade.it/documents/10279/32570951/20190915\\_Dettaglio\\_Opere\\_Darte\\_Maggiori\\_Ponti\\_e\\_Viadotti\\_v1.1.pdf](http://www.autostrade.it/documents/10279/32570951/20190915_Dettaglio_Opere_Darte_Maggiori_Ponti_e_Viadotti_v1.1.pdf)