



Istituto Statale di
Istruzione Superiore
GobettiVolta

Elaborato Esame di Stato

a.s. 2020/2021

Materie caratterizzanti:

Informatica, Sistemi e Reti

“Autostrade Toscana Manutenzione Infrastrutture”



Viadotto Italia

Docente Referente:

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
4.2 Accesso	17
4.3 Infrastrutture	19
4.4 Informazioni Infrastruttura	19
4.5 Mappa	20
4.6 Appalti	21
5. Schema di Rete	22
6. Simulazione Sensori	23
6.1 Analisi dei parametri della simulazione	25

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*.

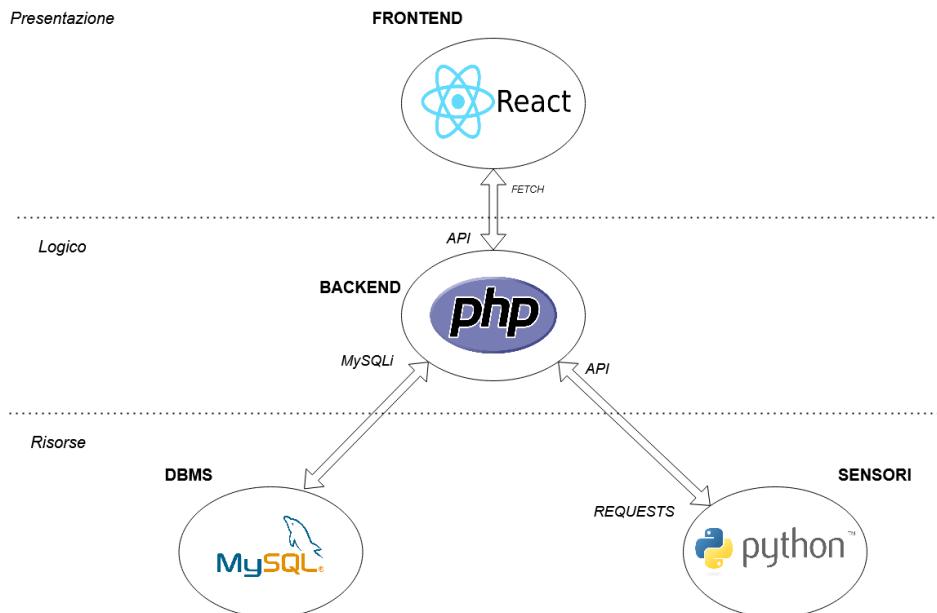


figura 1

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:

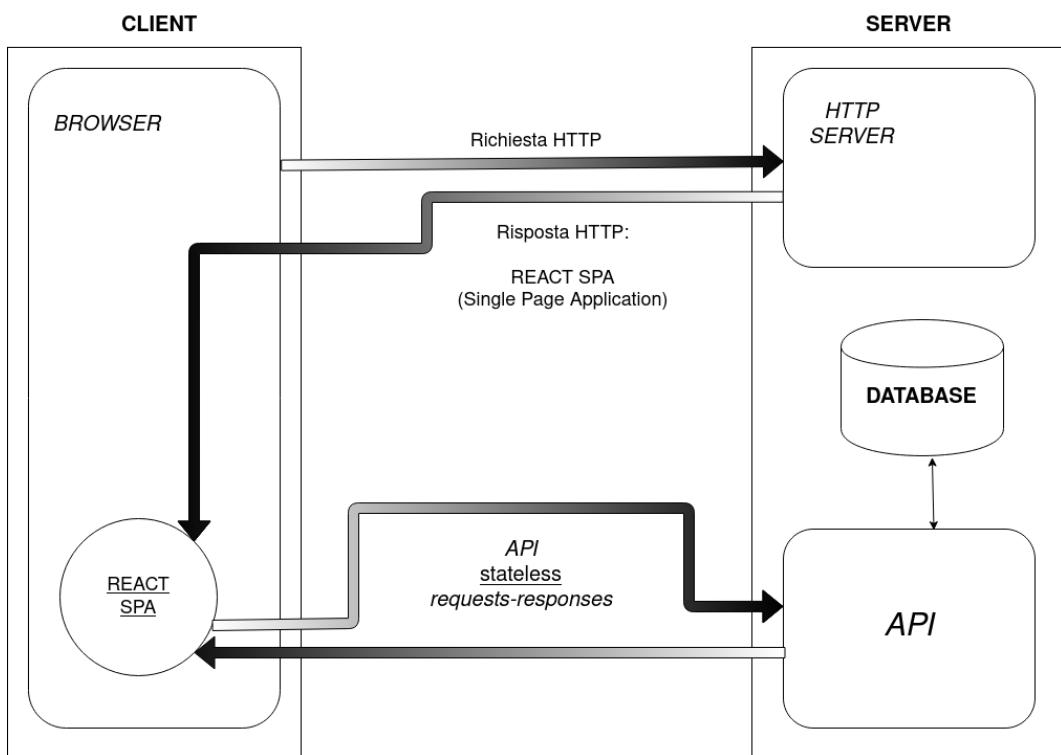


figura 2

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.

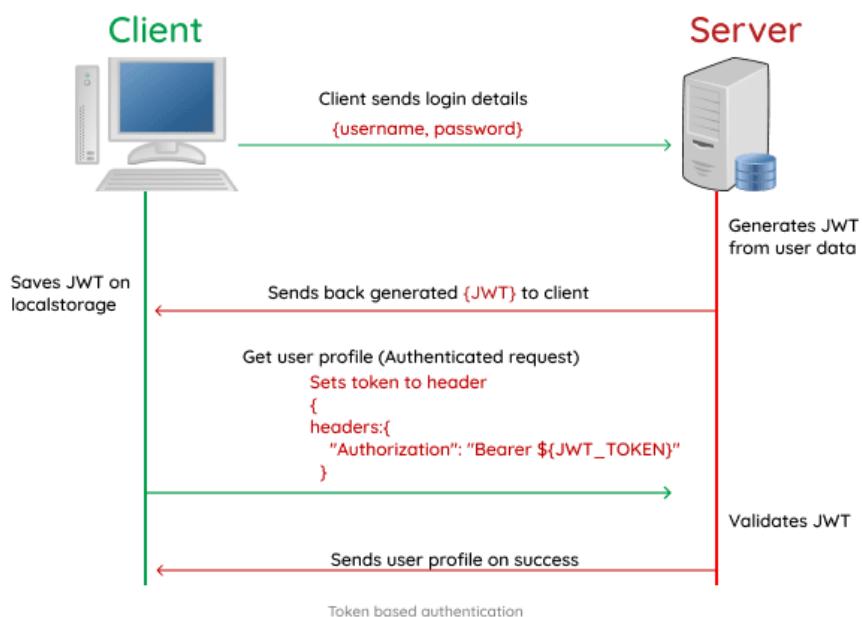


figura 3

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

3.2 Progettazione Concettuale

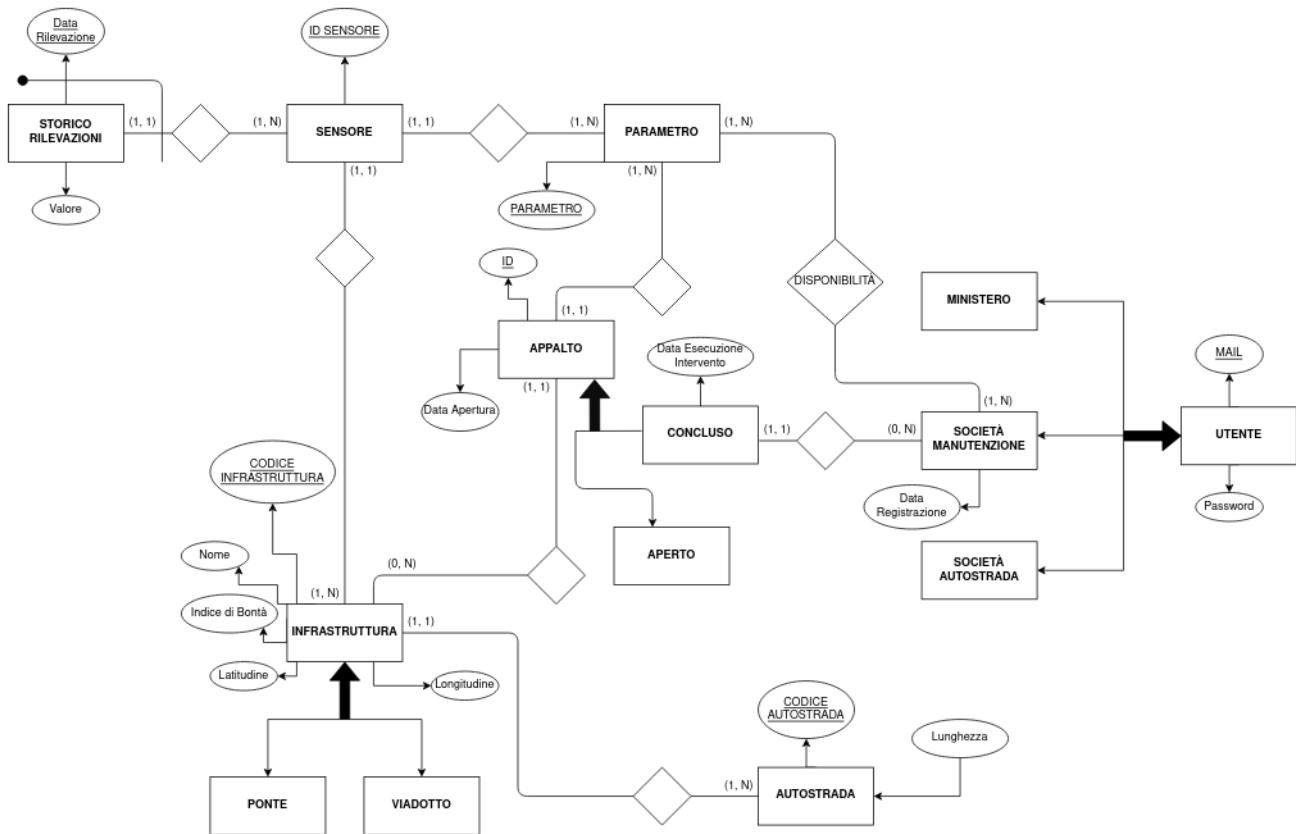


figura 4

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.

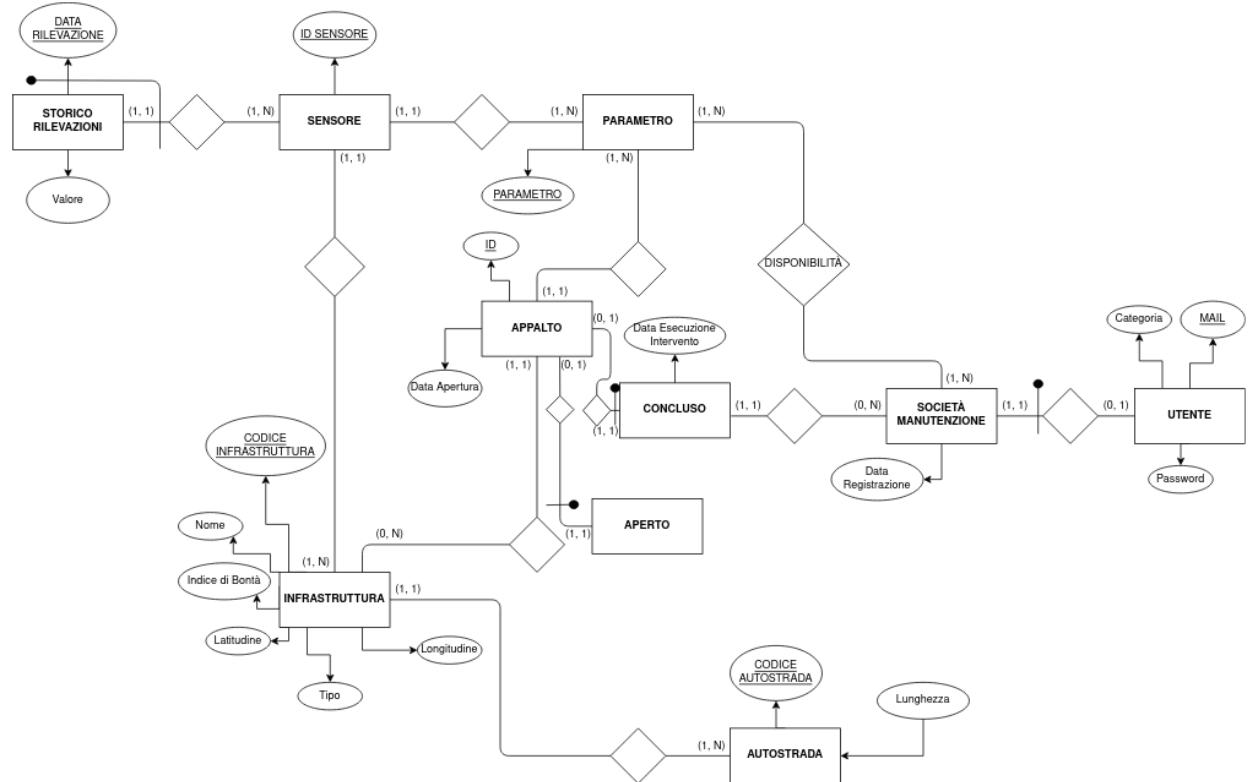


figura 5

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 6

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 7

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 8

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

Tutti i parametri sono impostati come NOT NULL perchè obbligatori.

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

figura 9

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 10

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 11

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 12

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 13

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", both with placeholder text. Below these is a section titled "Parametri di manutenzione:" with three checkboxes: "Elettricità", "Struttura", and "Asfalto". At the bottom left is a blue "Torna al login" link, and at the bottom right is a purple "Registrati" button.

figura 14

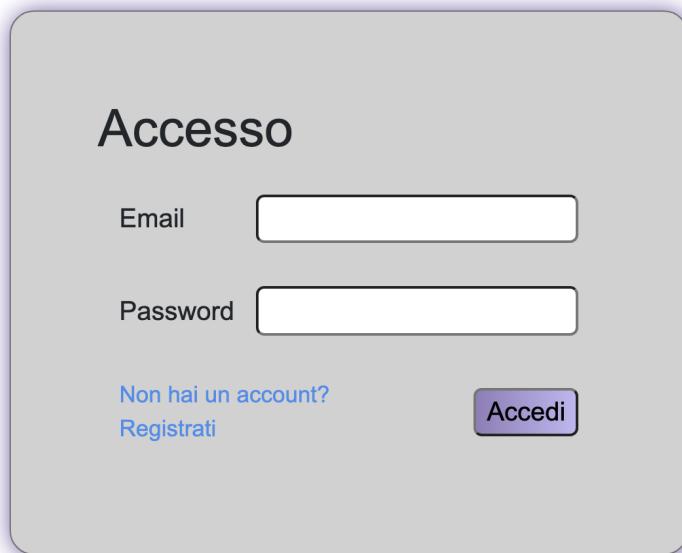
Questo è il form per effettuare la registrazione, effettuabile solamente da società di manutenzione ponti e viadotti.

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.

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.

4.2 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 hyperlinks: "Non hai un account?" and "Registrati". To the right of the "Registrati" link is a purple rectangular button labeled "Accedi". The entire form is set against a light gray background with a rounded rectangular border.

figura 15

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 16

- Società di Manutenzione, interfaccia parziale



figura 17

Come vedremo 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.

4.3 Infrastrutture

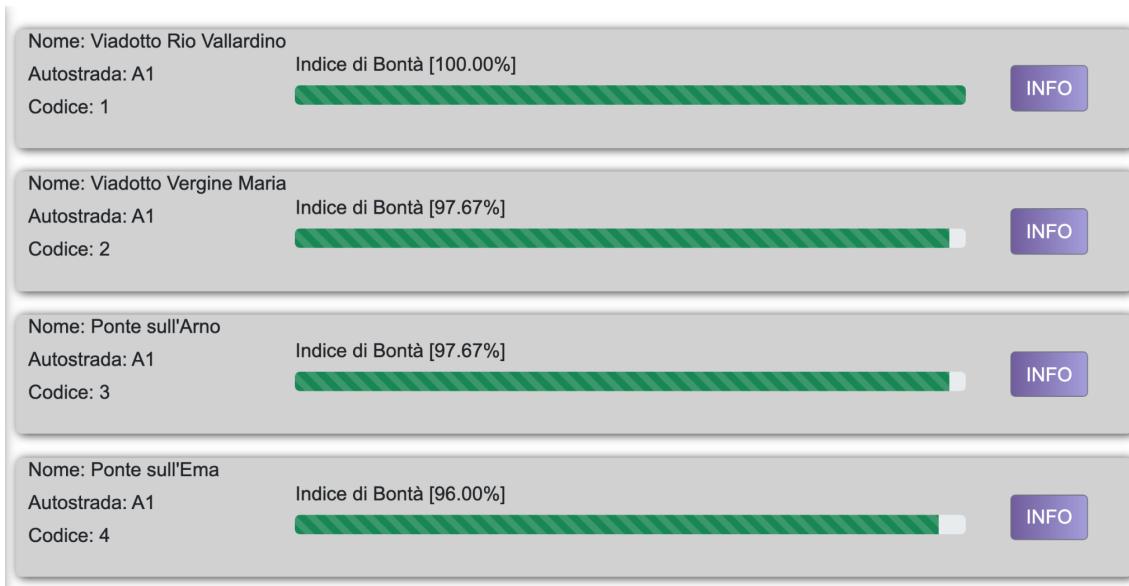


figura 18

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](#).

4.4 Informazioni Infrastruttura

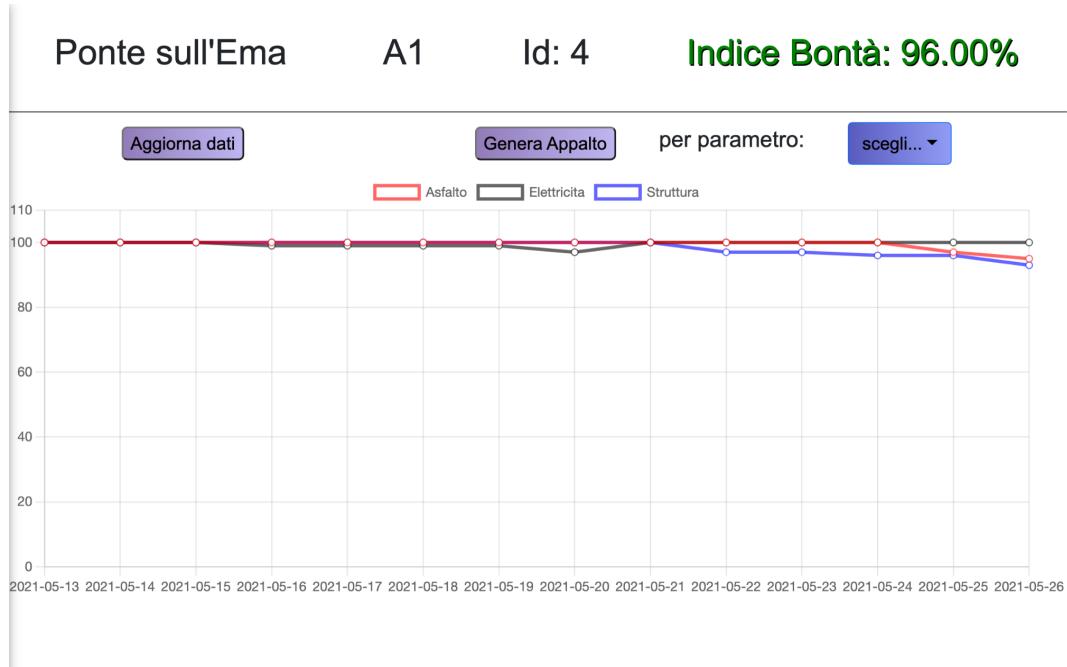


figura 19

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.

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.

4.5 Mappa

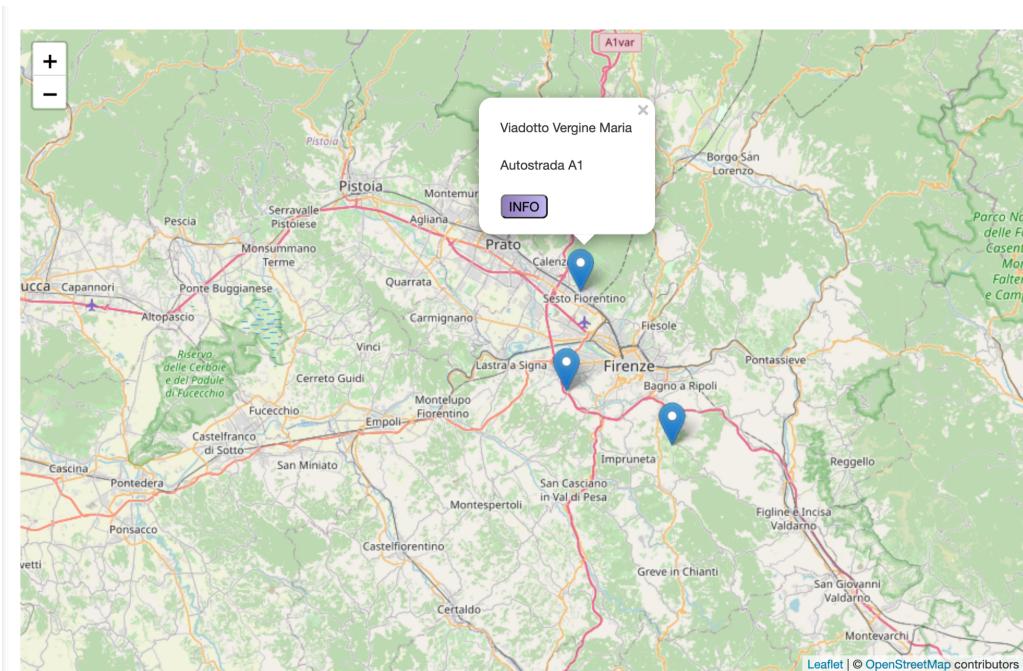


figura 20

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.

4.6 Appalti



figura 21

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.

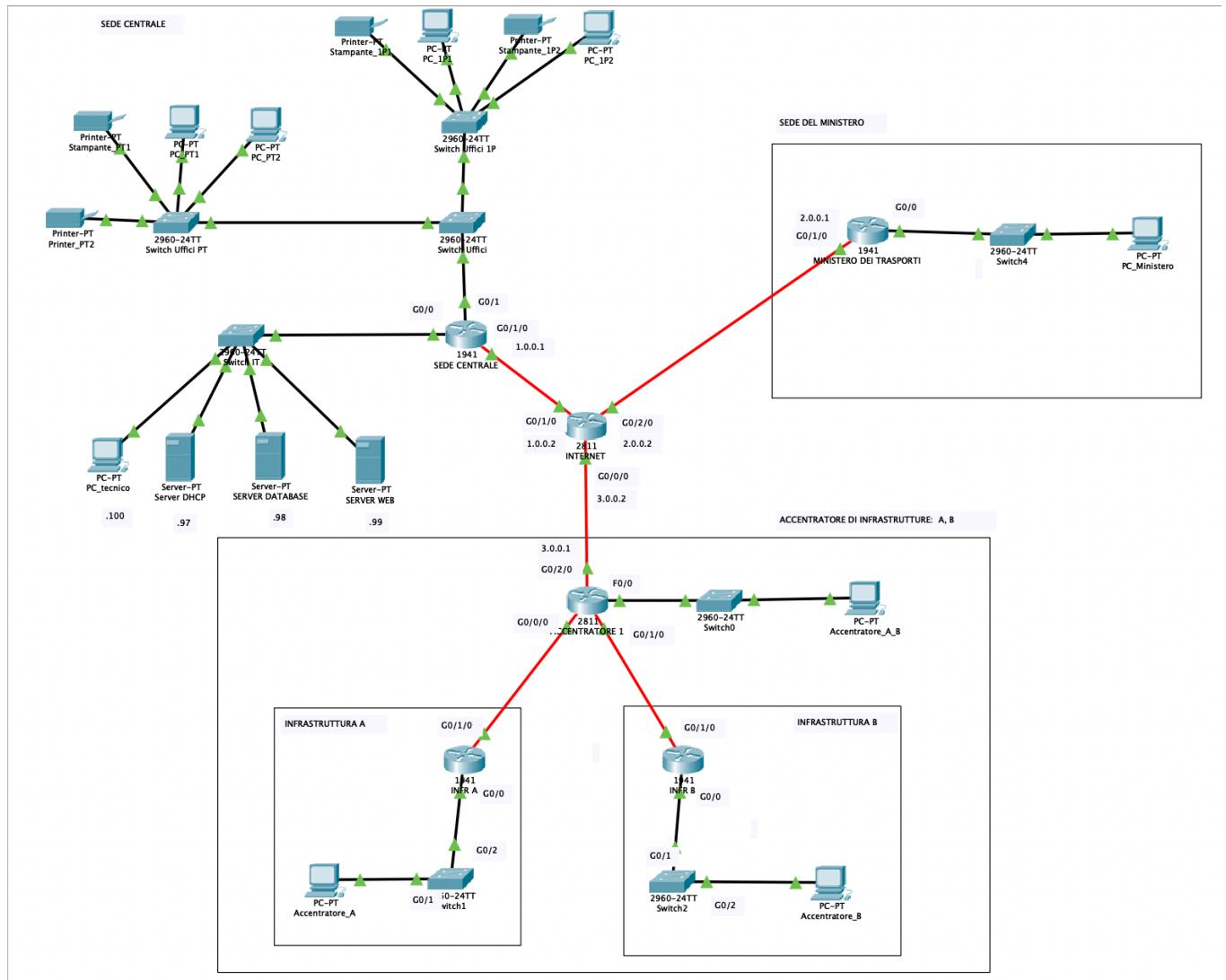


figura 22

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.

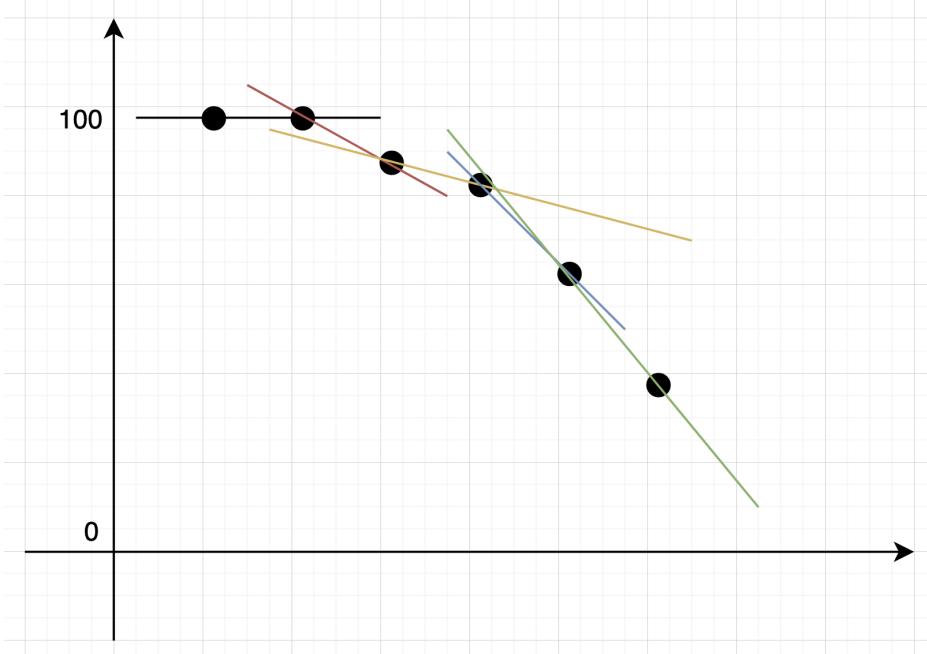


figura 23

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 24

Nella formula il minimo viene calcolato per imporre 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`.

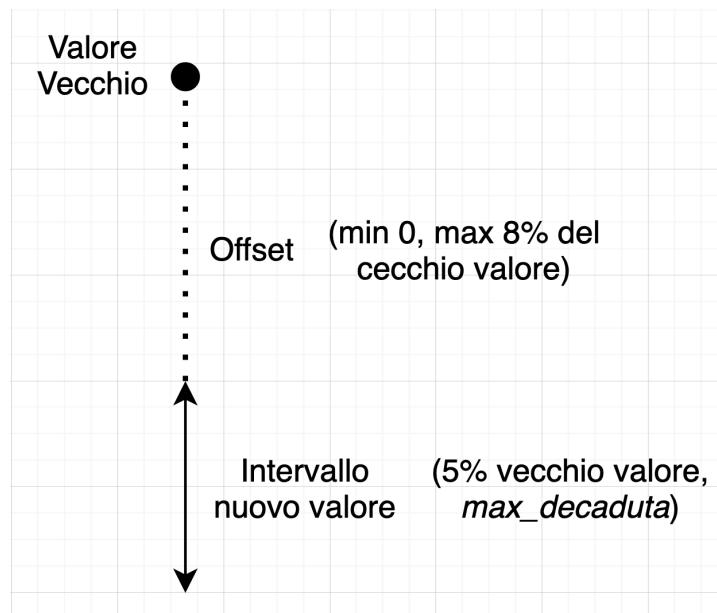


figura 25

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