



UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA

Corso di Laurea Triennale in Ingegneria Elettronica

**PROGETTAZIONE, SVILUPPO E TEST DI
FUNZIONALITÀ RELATIVE
ALL'INFRASTRUTTURA INFORMATICA DI
INTEGRITYKEY CON FOCUS SPECIFICO
SULLE ATTIVITÀ DI FRONT-END**

**DESIGN, DEVELOPMENT AND TESTING OF
FUNCTIONALITIES RELATED TO
INTEGRITYKEY'S IT INFRASTRUCTURE WITH A
SPECIFIC FOCUS ON FRONT-END ACTIVITIES**

Relatore: Chiar.mo
Prof. SIMONE FIORI

Tesi di Laurea di:
MARJO BARDHI

A.A. 2021/2022

INDICE

PREFAZIONE	5
1 INTRODUZIONE	6
1.1 APPLICAZIONI WEB.....	6
1.2 WEB 1.0.....	6
1.3 WEB 2.0.....	7
2 STUDIO DELLE TECNOLOGIE E DEGLI STRUMENTI	8
2.1 HTML.....	8
2.2 CSS.....	9
2.2.1 BOOTSTRAP 5.....	10
2.3 JavaScript.....	11
2.4 JSON	12
2.5 Librerie JavaScript	13
2.5.1 Chart.js.....	14
2.5.2 Leaflet.js.....	15
2.6 GIT	16
2.6.1 GITHUB	16
2.7 VISUAL STUDIO CODE.....	17
3 PROGETTO	18
3.1 OBIETTIVO.....	18
3.2 DATI	18
3.2.1 AZIENDE	18
3.2.2 MAPPA.....	19
3.3 REALIZZAZIONE CODICE.....	20
3.4 CONCLUSIONI.....	26
3.5 RISULTATO FINALE:	27
4 BIBLIOGRAFIA E SITOGRAFIA	27

PREFAZIONE

Lo scopo di questo progetto è quello di creare un sito web che combina l'utilizzo di diversi linguaggi di programmazione e librerie. In particolare, nella prima parte di questo lavoro vengono spiegate tutte le tecnologie impiegate nella creazione della pagina web sia da un punto di vista storico che descrittivo. Nella seconda parte viene descritto il codice utilizzato con tanto di spiegazione delle varie funzioni e immagini dei risultati ottenuti. Questi strumenti sono estremamente importanti, in quanto ai giorni nostri è impensabile immaginare di dover vivere senza internet e i vantaggi che derivano da esso. Infatti, il web è diventato una delle risorse più importanti in molti aspetti della vita, sia in ambito sanitario che istruttivo, oltre che semplicemente ricreativo. La conoscenza e la padronanza delle tecnologie descritte in questa tesi consentono, in poche parole, di creare tutta la parte di un sito web che gli utenti vedono e con la quale possono interagire.

1 INTRODUZIONE

1.1 APPLICAZIONI WEB

Una applicazione web indica, in informatica, tutte le applicazioni accessibili via web per mezzo di un network. Un servizio web è un software system progettato per supportare l'interazione tra computer su Internet. Nel mondo di oggi, lo scambio di informazioni e una comunicazione efficiente sono forti necessità. Il web ormai è una risorsa sempre più importante in molti aspetti della vita, quali istruzione, assistenza sanitaria, commercio, attività ricreative e molto altro. Il web è un sistema di documenti ipertestuali interconnessi a cui si accede tramite Internet. Con un browser web, l'utente ha la possibilità di visualizzare pagine web che possono contenere testo, immagini, video e altri contenuti multimediali e navigare tra di esse utilizzando hyperlinks.

Il web è stato creato nel 1989 da Sir Tim Barners-Lee, che lavorava al CERN di Ginevra in Svizzera. Da allora Barners-Lee ha svolto un ruolo attivo nel guidare lo sviluppo di standard web, e negli ultimi anni ha sostenuto la sua visione di un web semantico.

1.2 WEB 1.0

Nel web 1.0 un piccolo numero di scrittori creava pagine web per un gran numero di lettori. Di conseguenza le persone potevano ottenere informazioni andando direttamente alla fonte. Il WWW o Web 1.0 è un sistema di documenti ipertestuali interconnessi a cui si accede tramite internet.

La prima implementazione del web rappresenta il web 1.0 che, secondo Barners-Lee potrebbe essere considerato il “web di sola lettura”. In altre parole, il primo web ci permetteva di cercare informazioni e di leggerle. L'interazione con l'utente e il contributo ai contenuti erano molto limitati. Tuttavia, questo è esattamente ciò che la maggior parte dei proprietari di siti web voleva. L'obiettivo di un sito web era stabilire una presenza online e rendere le proprie informazioni disponibili a chiunque in qualsiasi momento.

Tra i vantaggi di un sito web statico (web 1.0), oltre al non coinvolgere nessuna attività di programmazione lato server, possiamo includere che lo sviluppo di un sito di questo tipo è tipicamente meno costoso, in quanto richiede minimi carichi di CPU per il relativo server e non necessita la presenza di database per la separata memorizzazione dei contenuti. Per questi motivi più la sicurezza per quanto riguarda un attacco hacker rispetto a un sito dinamico che ne deriva, e per la velocità superiore rispetto a un sito dinamico, poiché hanno molte meno righe di codice, i siti statici sono ancora largamente utilizzati per pubblicazioni web di piccola entità o come semplici siti informativi. Il punto debole di un sito di questo tipo, che permette all'utente di accedere alle pagine, navigare tra i vari contenuti, visualizzare immagini e altri media, è che non permette nessuna interazione dinamica, come ad esempio l'inserimento di commenti in un forum o la scrittura di un post in un blog.

1.3 WEB 2.0

Il web 2.0, o come lo descrive Barners-Lee, il web "read-write", consente di contribuire ai contenuti web e di interagire con altri utenti.

Il web 2.0 utilizza le tecnologie e i concetti più recenti per rendere l'esperienza dell'utente più interattiva, utile e interconnessa. Ha portato un altro modo di interconnettere il mondo, raccogliendo informazioni e consentendo di condividerle in modo efficace. Questo tipo di web prende il nome di web dinamico e viene utilizzato quindi per indicare tutte le applicazioni web che interagiscono attivamente con l'utente, infatti il web 2.0 modifica le informazioni mostrate in base alle informazioni generate dall'utente e ricevute dall'application server. Grazie a questa tecnologia è ora possibile creare Ecommerce, blog, forum e siti web dove l'utente può commentare, registrarsi, interagire con gli altri utenti del sito, chattare e addirittura modificare il sito stesso tramite le wiki. Il Content Management System consente anche ad un utente che non conosce linguaggi come HTML, CSS, JavaScript, PHP, SQL... di creare un sito statico senza contenuti con estrema facilità. D'altro canto, in un sito web di questo genere si

rischiano attacchi informatici, furto di identità e furto dei dati di pagamento degli utenti.

2 STUDIO DELLE TECNOLOGIE E DEGLI STRUMENTI

2.1 HTML

HTML è una delle tecnologie fondamentali del web, ed è spesso utilizzato in congiunzione con CSS e JavaScript nella creazione delle pagine web.

Nasce nei primissimi anni 90 da Tim Berners-Lee al CERN di Ginevra (Svizzera) insieme al protocollo http dedicato al trasferimento di documenti in tale formato.

HTML (Hypertext Markup Language) è un linguaggio usato per strutturare e organizzare il contenuto di una pagina web, che può essere formato da testo, immagini o altri elementi multimediali.

HTML è costituito da una serie di elementi, ognuno dei quali è rappresentato da un tag che definisce la struttura e il layout della pagina.

I cosiddetti tag sono formati da un tag di apertura “<tag>” il quale indica l’inizio della parte di codice che verrà espressa usando il linguaggio HTML per poi essere chiuso usando la voce “</tag>”.

La struttura base di un documento HTML è formata dall’header della pagina, indicato con “<head>”, e dal corpo della pagina indicato con “<body>”.

L’“<head>” racchiude delle informazioni importanti per il funzionamento della pagina, che noi però non possiamo vedere dal dispositivo, e racchiude inoltre un “<title>”, ovvero il titolo che rappresenta il contenuto della pagina.

Il “<body>” invece è costituito dal contenuto effettivo della pagina.

Nel 2014 è stata rilasciata la versione 5 di HTML, versione che rappresenta una vera e propria innovazione nella progettazione delle pagine web dando la possibilità di integrare all’interno di esse contenuti multimediali quali audio, video e animazioni grafiche tramite i comandi “<video>”, “<audio>”, “<canvas>”.

Grazie all'introduzione di nuovi tag, quali “<header>”, “<nav>”, “<article>”, “<section>”, “<aside>” e “<footer>”, con HTML5 viene data la possibilità di migliorare la struttura semantica della pagina.

Esempio di codice HTML:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title> HTML Hello World </title>
5      </head>
6      <body>
7          <h1> Hello World! </h1>
8      </body>
9  </html>
```

2.2 CSS

CSS, o Cascading Style Sheets, è un linguaggio usato per programmare la resa grafica di documenti scritti in un linguaggio di markup, come HTML e XML. Le classi CSS sono usate per specificare attributi grafici come font, dimensione, colore, spaziatura, bordo e posizione degli elementi all'interno di una rappresentazione web. Insieme ad HTML e JavaScript, CSS è uno dei tre pilastri della programmazione front end: regolano rispettivamente il contenuto (HTML), la dinamicità (JavaScript) e l'aspetto visivo (CSS) di una pagina web. CSS permette agli sviluppatori di separare la presentazione grafica di una pagina web dalla sua struttura e dal suo contenuto, rendendo quindi più facili aggiornamenti che riguardano il design di una pagina web. CSS offre un ampio range di opzioni di stilizzazione del testo (font-size, font-family...), adattamenti cromatici (color, background-color...) e formattazione del

testo (text-align, text-indent, text-transform...), che lo rendono, accanto all'HTML, lo standard web più importante nel campo del web design.

Esempio di codice CSS:

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <title>Document</title>
6      <style>
7          p {
8              color: ■ green;
9          }
10
11         .color-blue {
12             color: ■ blue;
13         }
14
15         #color-red {
16             color: ■ red;
17         }
18     </style>
19 </head>
20
21 <body>
22     <p>Green text !</p>
23     <p class="color-blue">Blue text !</p>
24     <p id="color-red" class="color-blue">Red text !</p>
25 </body>
26
27 </html>
```

2.2.1 BOOTSTRAP 5

Bootstrap 5 è l'ultima versione del popolare framework open-source per front-end, Bootstrap.

Si tratta di un framework “responsive” e “mobile-first”, progettato per facilitare le creazioni di pagine web e applicazioni web con le suddette caratteristiche.

Bootstrap 5 si basa su HTML, CSS e JavaScript e fornisce una ampia serie di componenti precostituiti, come “form”, “buttons” e “navigation bars”, che possono essere facilmente personalizzati e integrati in una pagina web.

Una delle caratteristiche principali di Bootstrap 5 è il suo sistema di “responsive grids”, che consente di creare facilmente layout che si adattano alle diverse dimensioni dello schermo. Inoltre, sono inclusi anche una serie di classi CSS precostituite per la tipografia, i colori, la spaziatura e altro ancora, che possono essere usate per creare in modo rapido lo styling di una pagina web.

Bootstrap viene molto utilizzato dagli sviluppatori web e vanta una comunità numerosa e attiva, che ha creato una serie di plugin e componenti aggiuntivi che estendono le funzionalità del framework, rendendolo ancora più potente.

2.3 JavaScript

JavaScript è un linguaggio di scripting lato client utilizzato per rendere interattive e dinamiche le pagine web. Insieme ad HTML e CSS, JavaScript è una delle tecnologie principali della programmazione web front end. Il suo ruolo è gestire il comportamento degli elementi dinamici di un sito web. JavaScript è un linguaggio di scripting lato client, il che significa che viene eseguito direttamente nel browser web, anziché su un server. Il vantaggio di questo approccio è che, anche con la presenza di script particolarmente complessi, il web server non rischia sovraccarichi. JavaScript consente agli sviluppatori di creare un'ampia gamma di elementi interattivi sulle pagine web, come responsive forms, dropdown menus, e image sliders. Trova impiego infatti anche in giochi (2d o 3d) basati su browser web e creazione di diapositive animate anch'esse basate sul web. JavaScript è un linguaggio di alto livello, il che significa che è facile da leggere e scrivere e non ha bisogno di essere compilato prima di essere eseguito, inoltre, è anche un linguaggio multi-paradigma, il che significa che supporta diversi stili di programmazione, come la programmazione orientata agli oggetti e quella funzionale. JavaScript è diventato uno dei linguaggi di programmazione più usati al mondo, diventando, da un semplice linguaggio di scripting per creare pagine web interattive, un potente strumento per costruire applicazioni web complesse.

2.4 JSON

JSON (JavaScript Object Notation) è un formato leggero per l'interscambio di dati, facile da leggere e scrivere per gli umani e da analizzare e generare per le macchine. JSON è un formato di testo completamente indipendente dal linguaggio, ma utilizza convenzioni familiari ai programmatori della famiglia dei linguaggi C.

Queste proprietà rendono JSON un linguaggio ideale per lo scambio di dati.

I dati JSON sono rappresentati come coppie key-value, dove le "key" sono stringhe e i "value" possono essere stringhe, numeri, array, oggetti, valori booleani (vero/falso) o null. Gli oggetti JSON sono racchiusi in parentesi graffe "{}" e contengono un insieme di coppie key-value separate da virgole. Le "key" e i "value" sono invece separati da due punti ":".

Ecco un esempio di oggetto JSON;

```
{
  "nome": "Mario",
  "cognome": "Rossi",
  "eta": 35,
  "matricola": "12345",
  "laureato": false,
  "indirizzo": {
    "via": "Piazza Roma",
    "numero_civico": "22",
    "cap": "60121",
    "citta": "Ancona",
    "nazione": "Italia"
  },
  "corsi": [
    {
      "materia": "Analisi Matematica 1",
      "crediti": 9
    },
    {
      "materia": "Telecomunicazioni",
      "crediti": 9
    }
  ]
}
```

In questo esempio, l'oggetto JSON ha sette coppie key-value: "nome", "cognome", "età", "matricola", "laureato", "indirizzo", e "corsi". I valori di "nome", "cognome" e "matricola" sono stringhe, il valore di "età" è un numero, il valore di "laureato" è un booleano e il valore di "corsi" è un array di oggetti JSON.

Gli array JSON sono rappresentati da parentesi quadre "[]" e contengono un insieme di valori separati da virgole. I valori di un array JSON possono essere di qualsiasi tipo, compresi altri oggetti JSON e array.

Il formato JSON è frequentemente utilizzato per la comunicazione asincrona tra browser e server, per l'archiviazione di dati e per lo scambio di dati in una varietà di linguaggi di programmazione. Molte API web molto conosciute, ad esempio Twitter API, restituiscono i dati in formato JSON.

Inoltre, JSON viene utilizzato per la serializzazione e la trasmissione di dati strutturati attraverso connessioni di rete, generalmente tra un server e un client.

In conclusione, JSON è un formato di interscambio dati leggero e flessibile, facile da leggere, scrivere e analizzare, il che lo rende una scelta ideale per lo scambio di dati tra sistemi e piattaforme diverse.

2.5 Librerie JavaScript

Per facilitare il lavoro di programmazione vengono ormai usati degli strumenti detti librerie JavaScript e framework, nei quali sono stati raccolti molti comandi utili.

Le librerie JavaScript contengono varie funzioni, metodi o oggetti per eseguire compiti pratici su una pagina web o un'applicazione basata su JS, infatti esse contengono codici o funzioni che gli sviluppatori possono riutilizzare senza doverli scrivere da capo. Questo permette un grosso risparmio in termini di tempo e fatica, ed è quindi il motivo per cui esistono molteplici librerie JavaScript per diverse necessità di uso.

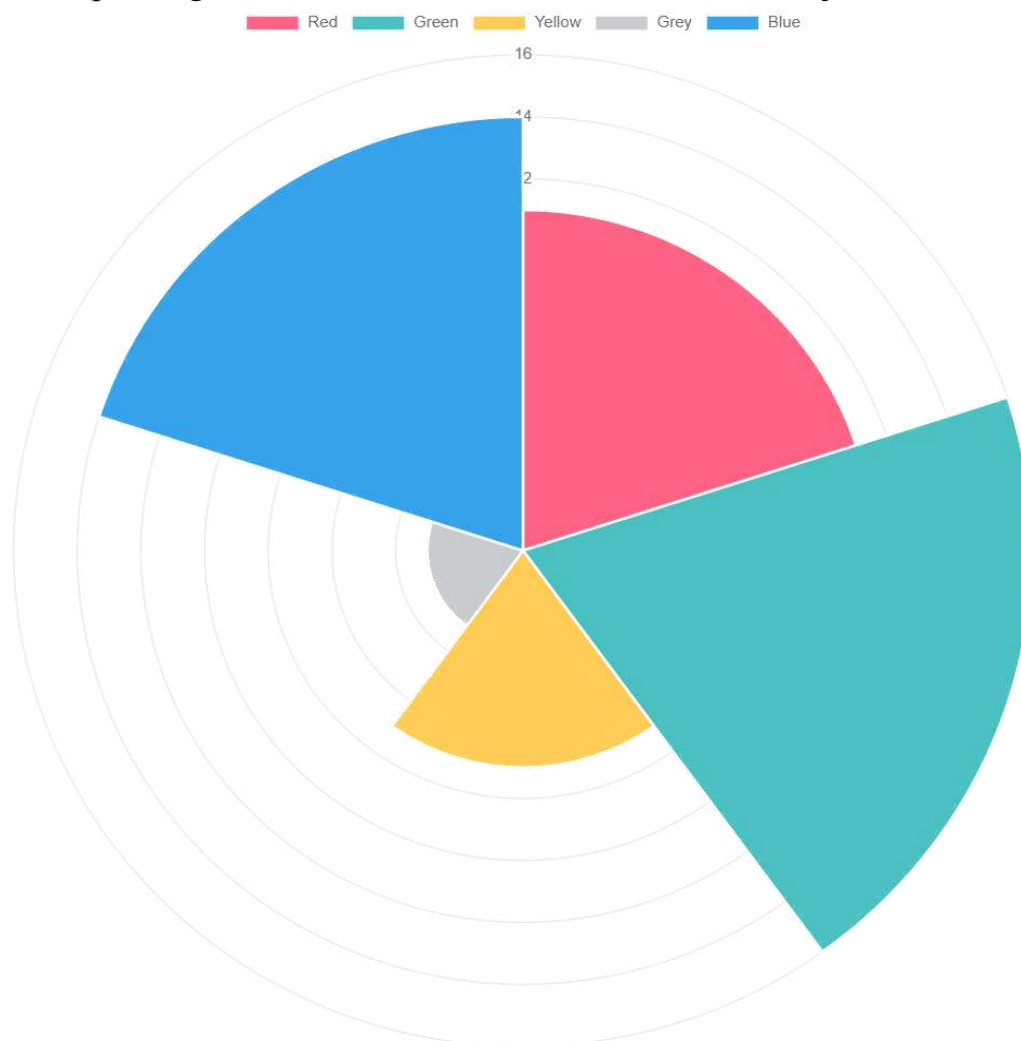
Queste librerie permettono, ad esempio, di visualizzare i dati in mappe e grafici (Chart.js, Apexcharts..), di trattare enormi quantità di dati (D3.js), di gestire in modo efficace un database (Taffy.db...), di semplificare le funzioni dei moduli (wForms, LiveValidation...) e altro ancora.

2.5.1 Chart.js

Chart.js è una libreria JavaScript open-source, quindi chiunque può eseguire, studiare, modificare e ridistribuire il codice sorgente.

Chart.js fornisce un insieme di tipi di grafici, plugin e opzioni di personalizzazione che vengono usati più frequentemente. Inoltre, è possibile combinare diversi tipi di grafico in un grafico misto (essenzialmente fondendo più tipi di grafico in uno sullo stesso canvas). Chart.js è altamente personalizzabile con plugin personalizzati per, ad esempio, creare annotazioni, zoom o funzionalità di trascinamento. Gli sviluppatori, grazie a questa libreria, hanno la possibilità di creare in modo facile una varietà di grafici (bar charts, line charts, pie charts...) semplicemente usando alcune linee di codice. La semplicità di utilizzo di questa libreria la rende una delle più usate nella creazione di grafici.

Esempio di grafico creato utilizzando la libreria Chart.js:



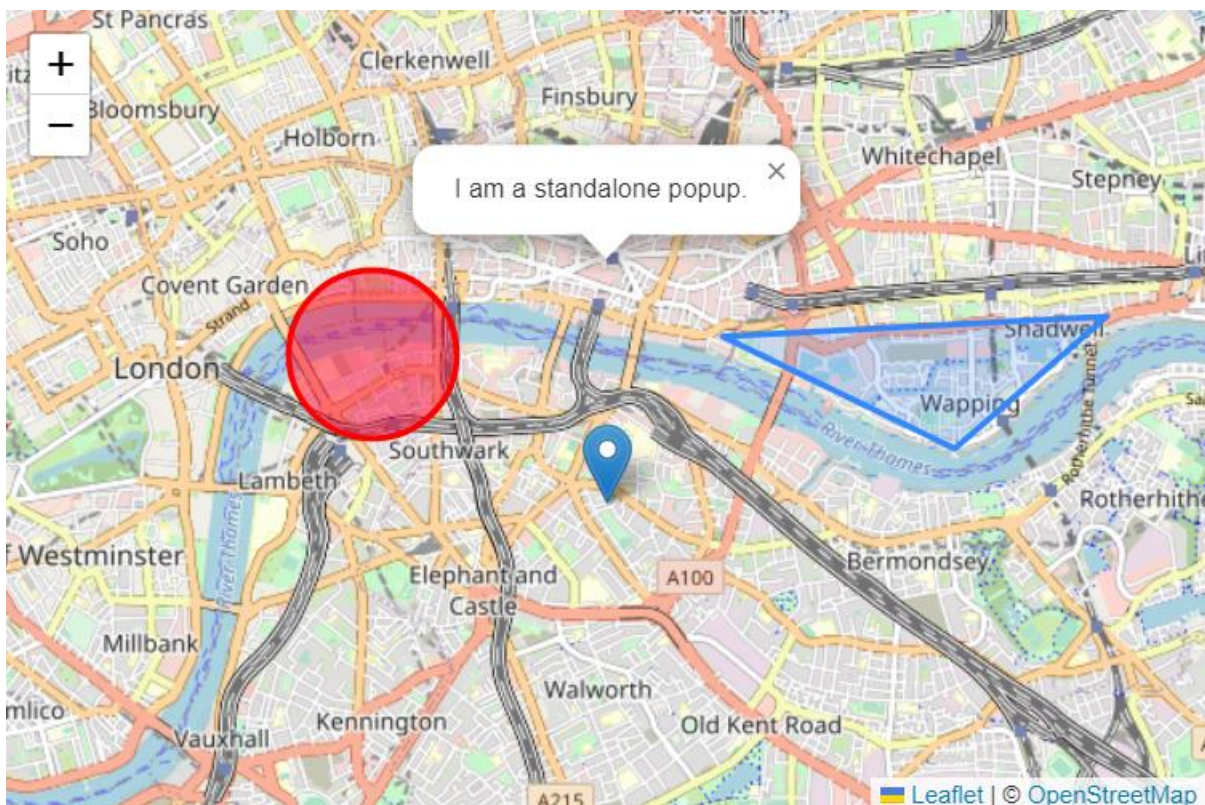
2.5.2 Leaflet.js

Leaflet è una libreria JavaScript open-source che serve a creare mappe interattive per siti web. Leaflet.js è stato progettato per essere leggero e facile da usare e si basa su OpenLayers, una libreria per creare mappe open-source.

Consente inoltre agli sviluppatori di creare facilmente mappe con diverse funzionalità, tra cui markers, popups e layers. Supporta inoltre un'ampia gamma di database di mappe, come OpenStreetMap, MapBox e Bing Maps, e consente l'uso di mappe personalizzate.

Leaflet.js fornisce una serie di controlli integrati per interagire con le mappe, come lo zoom, oltre a una serie di eventi per rispondere alle interazioni dell'utente con la mappa. Dispone inoltre di una comunità ampia e attiva che ha creato una serie di plugin per aggiungere ulteriori funzionalità alle mappe. Leaflet.js è compatibile con tutti i principali browser web e può essere utilizzato con una varietà di framework e librerie, come angular, react e vue.js, il che lo rende una delle librerie più diffuse per quanto riguarda l'ambito della creazione di mappe interattive nelle applicazioni web.

Esempio di una mappa Leaflet:



2.6 GIT

Git è un sistema che viene utilizzato per tracciare le modifiche ai file e collaborare ai progetti inerenti allo sviluppo del software. Permette agli sviluppatori di salvare le versioni del loro codice e di collaborare con altri su un progetto unendo le modifiche. Git memorizza le informazioni sul progetto in una repository, che altro non è che un insieme di file e cartelle. La repository contiene l'intera storia del progetto, compresi tutti i singoli commit di ogni file.

Una volta che uno sviluppatore apporta modifiche a un file in una repository Git, può effettuare un commit di tali modifiche nella repository. Un commit è un comando che viene utilizzato per creare una istantanea che include le modifiche effettuate ai file, una descrizione delle modifiche e informazioni sull'autore del commit. Git consente inoltre di creare rami (Branch) e di unire (Merge) il codice. Un branch è una linea di sviluppo separata che permette agli sviluppatori di lavorare sulla correzione di bug o su nuove funzionalità senza influenzare il codice principale. Una volta completato lo sviluppo su un branch, tramite il comando "merge" le modifiche possono essere unite nel branch principale. Git è molto utilizzato nello sviluppo dei software, in particolare perchè permette agli sviluppatori di collaborare facilmente sul codice, tenere traccia delle modifiche ed eventualmente ripristinare precedenti versioni. Git dispone di una potente Command Line Interface (CLI), ma sono disponibili anche numerose interfacce grafiche (GUI) per coloro che preferiscono un approccio di più facile utilizzo.

2.6.1 GITHUB

GitHub è una piattaforma web di hosting e gestione dei repository Git. Si basa su Git e fornisce una serie di funzioni aggiuntive per la collaborazione, quali "code review" e "project management".

La piattaforma fornisce quindi la possibilità di commentare linee di codice specifiche ed include anche un "issue tracker", che consente agli sviluppatori di tenere traccia dei bug e di altre attività relative al progetto. GitHub include anche una serie di funzioni di social coding, come ad esempio la possibilità di seguire altri utenti o visualizzare

statistiche e scoprire nuovi progetti. Offre inoltre una interfaccia web pubblica che consente a chiunque di visualizzare il codice sorgente e di scaricare una copia del repository. Il motivo principale per il quale GitHub è utilizzato da sviluppatori e aziende è quindi per la possibilità di “hostare” e collaborare su progetti open-source e privati. Dispone di una comunità molto numerosa e attiva e spesso viene utilizzato come portfolio per esporre i propri progetti.

2.7 VISUAL STUDIO CODE

Visual Studio Code è un editor di codice sorgente gratuito sviluppato da Microsoft, che può essere usato con vari linguaggi di programmazione senza dover cambiare editor. Tra i vari linguaggi supportati troviamo Python, Java, C, C++, C#, HTML, JavaScript e molti altri. Include inoltre il supporto per il Debugging, un controllo per Git integrato, syntax highlighting, IntelliSense, snippet e refactoring del codice. Consente anche la personalizzazione del tema dell’editor, le scorciatoie da tastiera e le preferenze.

Visual Studio Code è basato su Electron, un framework con cui è possibile sviluppare applicazioni Node.js.

3 PROGETTO

3.1 OBIETTIVO

L'obiettivo del progetto al quale ho partecipato, era quello di creare una pagina web che combina le tecnologie di cui ho parlato sopra, in particolare l'interazione tra Chart.js e Leaflet.js.

In sostanza, la finalità era quella di creare una pagina web che contenesse una mappa creata tramite Leaflet.js e un grafico creato tramite Chart.js contenente un set di informazioni presenti sulla mappa.

3.2 DATI

3.2.1 AZIENDE

In particolare, le informazioni riguardano fattorie didattiche, ovvero Aziende Agricole e Agrituristiche che accolgono scuole, famiglie, gruppi di adulti per una comunicazione diretta fra agricoltore e cittadino, appartenenti alla regione Umbria.

Queste informazioni sono state racchiuse all'interno del file "dati.json". Ogni azienda presente nel json è identificata da un "ID" e il relativo oggetto contiene informazioni riguardanti l'azienda stessa. (Le informazioni utilizzate nel progetto sono "NOME COMMERCIALE" e "COMUNE").

Estratto primo oggetto json:

```
[
  {
    "ID": "1",
    "NOME_COMMERCIALE": "COOPERATIVA DI SPERIMENTAZIONE TERRITORIALE PER LO SVILUPPO SOSTENIBILE",
    "RAGIONE_SOCIALE": "COOPERATIVA DI SPERIMENTAZIONE TERRITORIALE PER LO SVILUPPO SOSTENIBILE SOC.COOP.AGR.",
    "COGNOME_RAPP_LEGALE": "Mengucci",
    "NOME_RAPP_LEGALE": "Dino",
    "INDIRIZZO": "Loc. Le Guardie le Pierle",
    "COMUNE": "Passignano sul Trasimeno",
    "PROVINCIA": "PG",
```

```
"EMAIL": "buonaterra@libero.it",
"PEC": "pec@coopsperimentazione.telecompost.it",
"SITO": "http://www.pantarei-cea.it/",
"ATTO_DI_ISCRIZIONE": "4313",
"DATA_ISCRIZIONE": "07/05/2009",
"STATO_ATTIVITA": "CANCELLATA",
"PRESENZA_DI_STRUTTURA_AGRITURISTICA": "SI",
"PERNOTTAMENTO": "SI",
"ACCESSO_DISABILI": "SI"
},
```

...

3.2.2 MAPPA

Per evidenziare i comuni in una mappa c'è bisogno di conoscere i confini territoriali di tutti i comuni in una determinata provincia. Nel file poligoni.json ho inserito i dati provenienti dal GitHub repository [openpolis/geojson-italy](#).

Questo repository contiene quindi i confini territoriali georeferenziati di tutti i comuni italiani e la loro suddivisione per regioni e province. In particolare, i dati da noi utilizzati sono dati geojson (un formato per la codifica di strutture di dati geografici).

Questo file, in sostanza, contiene le coordinate geografiche dei vari comuni presenti

```
"features": [
  {
    "type": "Feature",
    "geometry": {
      "type": "Polygon",
      "coordinates": [ ...
    ]
  },
  "properties": {
    "name": "Assisi",
    "op_id": "4811",
    "minint_elettorale": "2100580010",
    "minint_finloc": "3100580010",
    "prov_name": "Perugia",
    "prov_istat_code": "054",
    "prov_istat_code_num": 54,
    "prov_acr": "PG",
    "reg_name": "Umbria",
    "reg_istat_code": "10",
    "reg_istat_code_num": 10,
    "opdm_id": "4705",
    "com_catasto_code": "A475",
    "com_istat_code": "054001",
    "com_istat_code_num": 54001
  }
},
],
```

nella regione raggruppate sotto “coordinates”.

Questo file è stato utilizzato interamente in congiunzione con Leaflet.js.

Le informazioni sono racchiuse nell’array “features”. L’array è composto da oggetti che racchiudo tutti i dati.

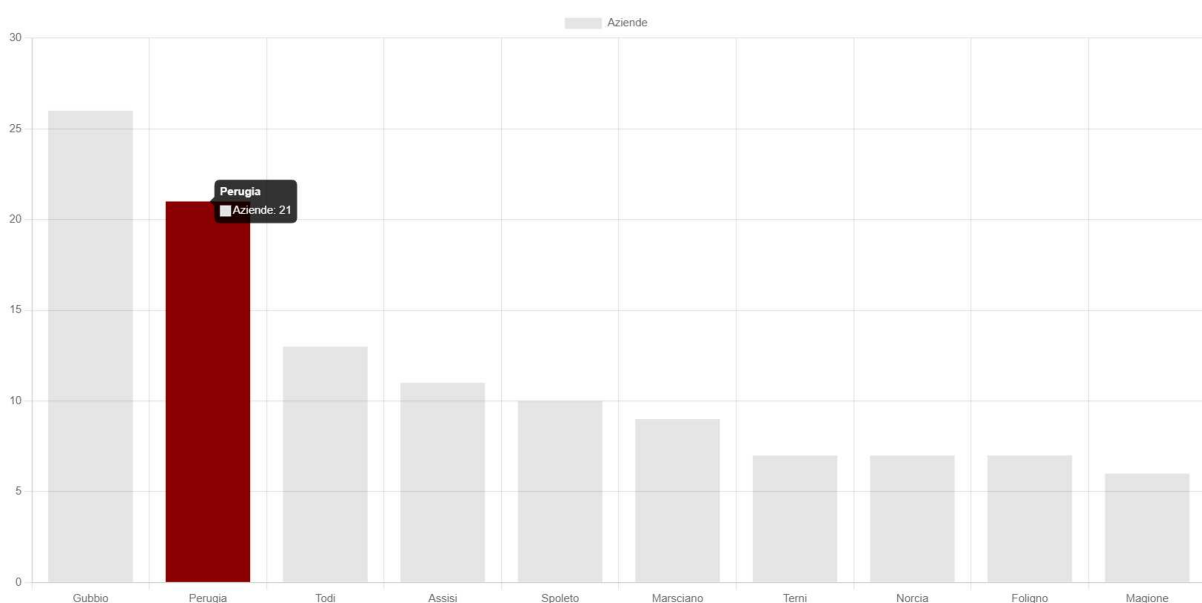
L'oggetto "geometry" contiene invece le coordinate racchiuse nell'array chiamato "coordinates":

```
"coordinates": [  
  [  
    12.695827612211195,  
    43.15368805794795  
  ],  
  [  
    12.695241977788978,  
    43.15195582196511  
  ],  
  [  
    12.694929493556732,  
    43.15113309914046  
  ]  
]
```

3.3 REALIZZAZIONE CODICE

Il grafico, di tipo "bar", deve contenere i dieci comuni con più aziende. Nell'asse x devono essere mostrati i nomi dei comuni mentre nell'asse y il numero delle aziende. Inoltre, passando col puntatore del mouse su una barra deve essere mostrato un popup contenente il nome del comune e il numero delle aziende presenti, e bisogna fare in modo che la barra si illumini di rosso finché il puntatore resta su di essa.

Grafico:



Nella realizzazione del codice, come prima cosa bisogna installare Chart.js.

In questo progetto è stato installato attraverso CDN, aggiungendo lo script tag che si può trovare sul sito ufficiale di Chart.js.

Per visualizzare il grafico dobbiamo aggiungere l'elemento canvas al div principale con l'id="Canvas" in questo modo:

```
<div class="container">
  <canvas id="Canvas"></canvas>
  <div id="map" class="mb-5"></div>
</div>
```

Ora bisogna utilizzare JavaScript per creare il grafico con le caratteristiche desiderate. A questo punto bisogna aprire un tag script in fondo al file HTML e creare una funzione async chiamata “creaTabella()”.

La funzione async dichiara una funzione asincrona e consente l’utilizzo della key word “await” all’interno del corpo della funzione.

Le key words “async” e “await” consentono di scrivere codici con comportamenti asincroni basati su promises in uno stile più pulito, evitando le promise chains.

La funzione “creaTabella()” deve essere asincrona perché chiamiamo una funzione ausiliaria chiamata “creaJSON()” che ci fornisce i dati necessari per creare la tabella.

Anche la funzione “creaJSON()” è asincrona perché chiama un’altra funzione asincrona chiamata “leggiJSON()”.

In pratica, la funzione “leggiJSON()” recupera i dati dal file fornito, li trasforma in JSON e li restituisce.

A questo punto “creaJSON()” invoca “leggiJSON()” che fornisce ad essa l’oggetto JSON. Poi “creaJSON”, attraverso un ciclo for, analizza l’intero file JSON, salva e restituisce tutti i comuni con le rispettive aziende. Con l’aiuto di un if è possibile evitare che lo stesso comune compaia due volte.

```
async function leggiJSON(nomeFile) {
  const response = await fetch(nomeFile);
  const json = await response.json();
```

```

        return json;
    }

    async function creaJSON(){
        const json = await leggiJSON("dati.json");
        var comuni = {};
        for(var i=0; i<json.length; i++){
            var oggetto = json[i];
            var comune = oggetto["COMUNE"];
            if(!(comune in comuni)){
                comuni[comune]=[];
            }

            comuni[comune].push(oggetto["NOME_COMMERCIALE"]);
        }
        var myjson = JSON.stringify(comuni);

        return comuni;
    }

```

Infine, la funzione “creaTabella()” inizia invocando “creaJSON()”, che restituisce tutti i comuni con le rispettive aziende.

```

async function creaTabella(){
    var comuni = await creaJSON();

```

Subito dopo aver ottenuto i dati, questi vengono ordinati in base al numero di aziende presenti in ogni comune.

```

var comuniOrdinati = Object.keys(comuni)
    .map((k)=>{return {key:k, value:comuni[k]}})
    .sort((a, b)=>{return b.value.length - a.value.length})

```

Dall’elenco ordinato vengono tagliati i primi 10 comuni con il maggior numero di aziende. Successivamente, vengono creati altri tre array per salvare solo comuni, aziende e dati.

```

const primiNComuni = comuniOrdinati.slice(0, 10);

let mieiComuni = []
let mieiDati = []
let mieAziende = []
for(var comune in primiNComuni){

```

```
mieiComuni.push(primiComuni[comune].key)
mieAziende.push(primiComuni[comune].value)
mieiDati.push(primiComuni[comune].value.length)
}
```

Ora ci troviamo in uno stato in cui abbiamo tre array (“mieiComuni”, “mieiDati”, “mieAziende”) e dobbiamo creare il grafico.

Nella creazione di un grafico, per prima cosa dobbiamo creare un oggetto di configurazione per il nostro grafico, poi, in secondo luogo dobbiamo ottenere l’elemento in cui vogliamo inserire il grafico, nel nostro caso l’elemento canvas con l’id = “Canvas”. Infine, utilizziamo la keyword “new” per creare il grafico, che prende due parametri, l’elemento canvas e l’oggetto di configurazione.

L’oggetto di configurazione ha questo formato:

```
const config = {
  type: 'bar',
  data: {
    labels: mieiComuni,
    datasets: [{
      label: "Aziende",
      data: mieiDati,
      hoverBackgroundColor: "darkred"
    }]
  }
}
```

Dobbiamo ora definire il tipo di grafico (nel nostro caso “bar”) e i dati.

L’oggetto dati contiene i label (nel nostro caso l’array “mieiComuni”) e l’array datasets che contiene un label e i dati (nel nostro caso “mieiDati”).

Per creare un effetto hover, aggiungiamo anche la proprietà “hoverBackgroundColor”. Infine, la funzione termina prendendo l’elemento canvas nel codice html e creando un nuovo oggetto Chart tramite la keyword “new”, fornendo al constructor il canvas e l’oggetto di configurazione.

```
const canvas = document.getElementById("Canvas");
const myChart = new Chart(canvas, config);
}
```

Ora possiamo chiamare la funzione “creaTabella()” per ottenere il nostro grafico.

```
creaTabella()
```

Per quanto riguarda invece la creazione della mappa (vuota inizialmente) utilizziamo il seguente codice:

```
var map = L.map('map').setView([42.938004, 12.621621], 9);
```

con, tra i parametri, le coordinate del punto di interesse della mappa, in modo che quando apriamo il sito web la mappa sia già ingrandita nella posizione desiderata.

Tramite il seguente codice aggiungiamo alla mappa (vuota) una immagine della mappa geografica mondiale:

```
L.tileLayer('https://tile.openstreetmap.org/{z}/{x}/{y}.png', {  
  attribution: '&copy; <a  
href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'  
}).addTo(map);
```

In questo modo quando il sito web viene aperto, la mappa viene mostrata con lo zoom impostato sulla regione Umbria.

Dobbiamo ora inserire i confini dei vari comuni della regione selezionata, e lo facciamo creando una funzione che chiamiamo “creaPoligoni()”, che prende come input la mappa appena creata.

Questa funzione come prima cosa usa la funzione "leggiJSON" per leggere il file poligoni.json che contiene tutte le coordinate geografiche dei vari comuni.

La funzione usa poi la funzione "creaJSON" per ricevere tutti i comuni con le rispettive aziende.

In seguito, crea due stili (uno di base e uno quando ci si passa col mouse) per i poligoni che verranno creati più avanti. In aggiunta creiamo un popup che verrà anch'esso usato in seguito.

```
async function creaPoligoni(map){  
  var geoJson = await leggiJSON("poligoni.json");  
  var comuni = await creaJSON("dati.json");  
  
  var baseStyle = {  
    "color": "blue",  
    "weight": 1,  
    "opacity": 0.3
```



```

};
var hoverStyle = {
    "color": "red",
    "weight": 1,
    "opacity": 1
};
var popup = L.popup();

```

Dopo aver acquisito il file "poligoni.json" lo utilizza nella funzione "L.geoJson", funzione che prende come parametri il json con tutte le coordinate geografiche e la configurazione dei poligoni nella mappa. Ogni poligono riceve lo style "baseStyle" che abbiamo creato sopra e la funzionalità di cambiare stile quando ci si passa sopra col mouse. La funzione è in grado di creare un popup quando si passa su una area della mappa col mouse. Se l'area di interesse è presente nel geoJson viene mostrato un popup con il numero di aziende presenti in quell'area.

```

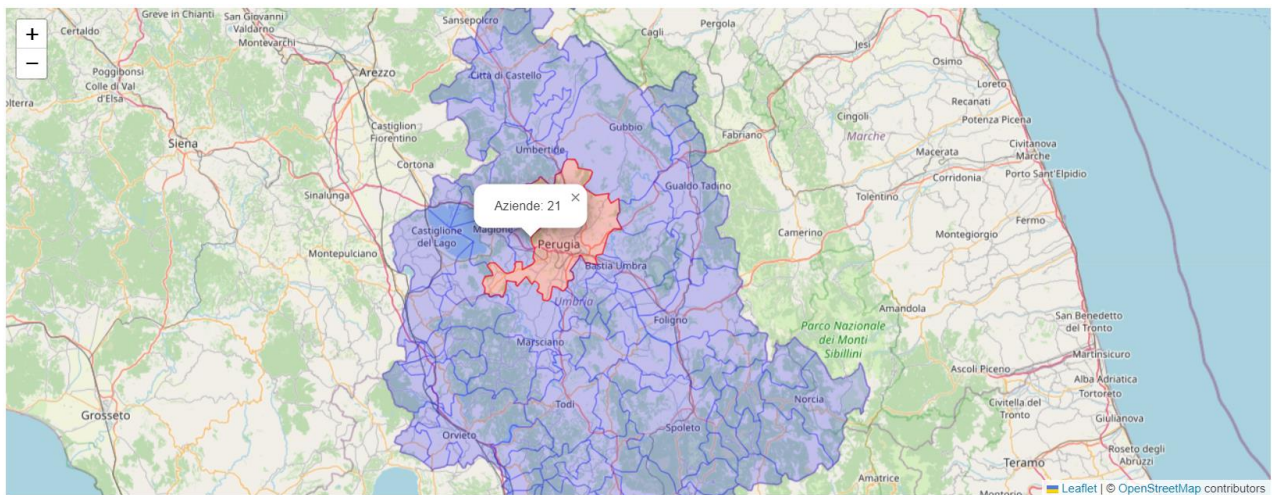
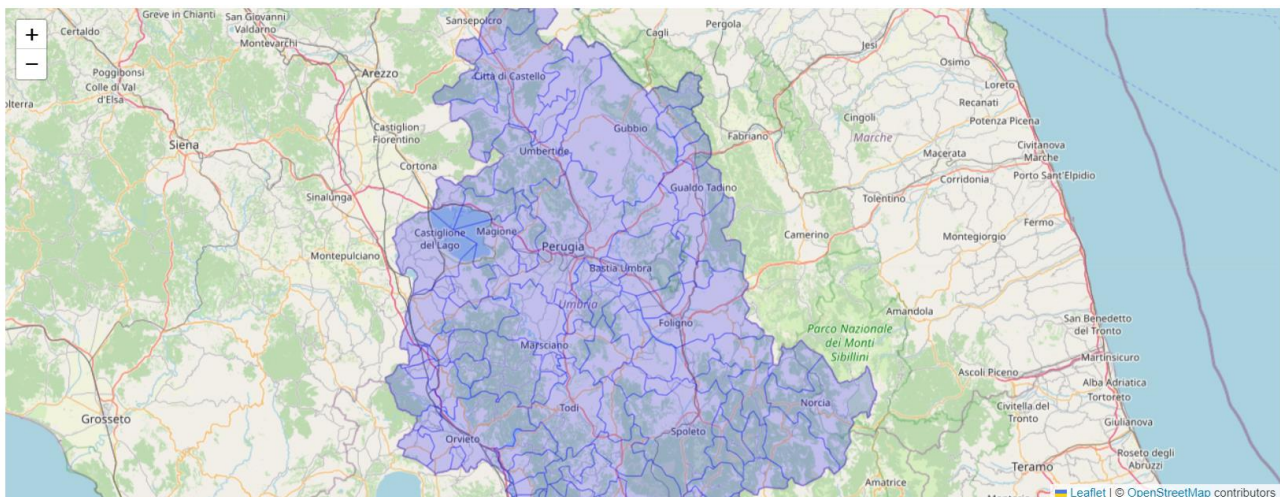
var geoMap = L.geoJson(geoJson, {
    style: baseStyle,
    onEachFeature: function(feature, layer) {
        layer.on('mouseover', function(e) {
            layer.setStyle(hoverStyle);

            if(comuni[layer.feature.properties.name]!==undefined){
                popup
                    .setLatLng(e.latlng)
                    .setContent("Aziende: " +
comuni[layer.feature.properties.name].length)
                    .openOn(map);
            }

        });
        layer.on('mouseout', function(e) {
            layer.setStyle(baseStyle);
            map.closePopup()
        });
    }
}).addTo(map);

```

Rappresentazione della mappa nelle sue varie sfaccettature:



3.4 CONCLUSIONI

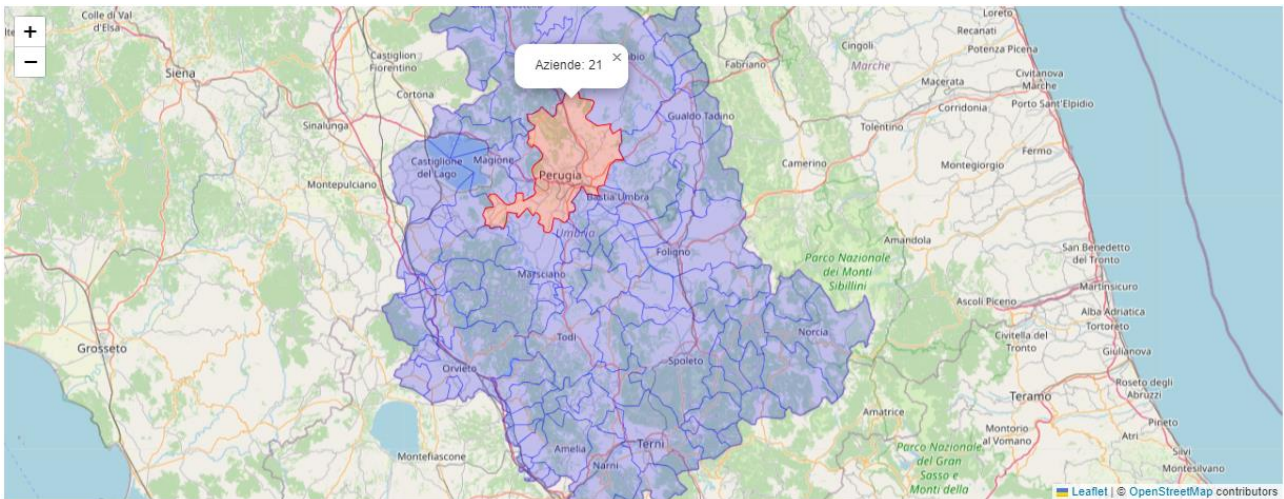
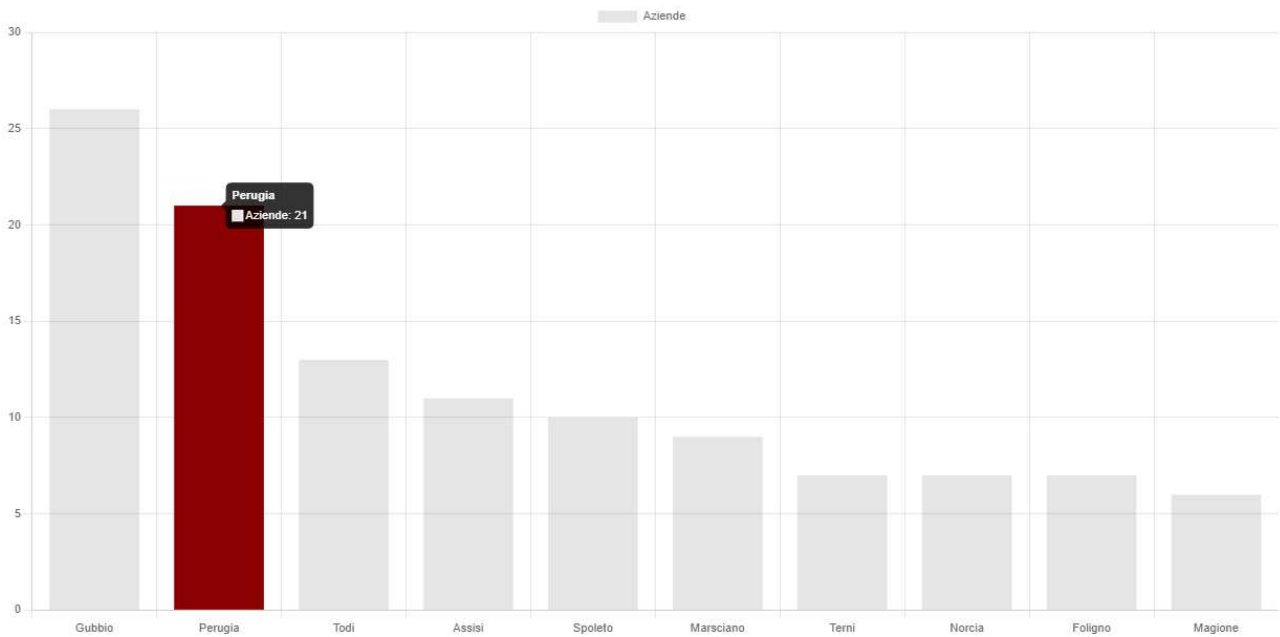
In questa tesi di laurea si è sviluppata una pagina web tramite l'utilizzo di tecnologie front-end moderne in grado di mostrare un grafico a barre interattivo contenente informazioni che possono essere visualizzate anche in una mappa geografica anch'essa interattiva.

Per la realizzazione della pagina web sono stati usati i seguenti linguaggi di programmazione:

- HTML
- CSS
- JavaScript

Per la creazione del grafico e della mappa JavaScript è stato usato in congiunzione con le librerie Chart.js e Leaflet.js.

3.5 RISULTATO FINALE:



4 BIBLIOGRAFIA E SITOGRAFIA

- Umesha Naik, Dr. D. Shivalingaiah, “Comparative Study of Web 1.0, Web 2.0 and Web 3.0”, Conferance Paper, March 2009.
- JavaScript, URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- Silvio Peroni, Francesco Osborne, Angelo di Iorio, Andrea Giovanni Nuzzolese, Francesco Poggi, Fabio Vitali e Enrico Motta, “Research Articles in Simplified

HTML: a Web-first format for HTML-based scholarly articles”, PeerJ Computer Science, October 2017.

- Teng Lv, Ping Yan, Weimin He, “Survey on JSON Data Modelling”, Journal of Physics Conference Series 1069, August 2018.
- Visual Studio Code, URL: <https://code.visualstudio.com/docs>
- Chart.js, URL: <https://chartjs.org>
- Leaflet.js, URL: <https://leafletjs.com>
- Repository file geojson, URL: <https://github.com/openpolis/geojson-italy>
- Elenco fattorie didattiche Regione Umbria,

URL:

https://dati.regione.umbria.it/dataset/elenco_regionale_fattorie_didattiche