



**UNIVERSITÀ  
DI TORINO**

**DIPARTIMENTO DI CULTURA, POLITICA E SOCIETÀ**

***Corso di laurea magistrale in  
Comunicazione, ICT e Media***

***Search Engine Optimization: dallo studio della  
pratica alla realizzazione di un software di analisi***

Relatore:

*Prof. Marino Segnan*

Candidato:

*Davide Morri*

Matricola: 955150

Anno accademico 2021 - 2022

## **INDICE:**

<b>INTRODUZIONE</b>	<b>2</b>
<b>CAPITOLO 1</b>	<b>4</b>
1.1 Breve definizione di Marketing	4
1.2 Panoramica sui motori di ricerca	5
1.3 Le tre fasi della ricerca di Google	8
1.4 L'algoritmo PageRank di Google	9
1.5 Aggirare l'algoritmo	14
1.6 Gli aggiornamenti di Google	16
1.7 I 200 fattori di posizionamento di Google	20
1.8 Le penalizzazioni	22
1.9 Cos'è la SEO?	24
1.10 La SEO Tecnica	26
1.11 La SEO Semantica	34
<b>CAPITOLO 2</b>	<b>38</b>
2.1 Domanda di ricerca	38
2.2 Analisi del mercato	38
2.3 Realizzazione di SEOgull	39
2.3.1 Front-end ed interazione con la GUI	39
2.3.2 Back-end	44
2.3.3 Il Web Crawler	46
2.4 Risposta alla domanda di ricerca	53
<b>CAPITOLO 3</b>	<b>54</b>
3.1 Limiti del progetto	54
<b>CONCLUSIONI</b>	<b>57</b>
<b>RINGRAZIAMENTI</b>	<b>58</b>
<b>BIBLIOGRAFIA</b>	<b>59</b>

# INTRODUZIONE

Questa tesi di laurea nasce come proseguimento di un progetto realizzato grazie al finanziamento della borsa di studio per approfondimento di tematiche di interesse per il Progetto di Eccellenza CPS a.a. 2021/2022. L'analisi dell'argomento è stata realizzata attraverso articoli e manuali, ma anche grazie a *blog*, documentazione ufficiale *online*, *workshop* e video-corsi tenuti da professionisti del settore, data la novità del tema e la sua rapida evoluzione.

Il progetto non si è limitato a porre il *focus* solo sullo studio della *Search Engine Optimization* (SEO), ma anche sulla realizzazione di un *software* che permettesse di applicare le conoscenze acquisite attraverso l'analisi di siti web.

L'obiettivo della ricerca è stato quindi quello di scoprire un nuovo argomento, studiarne i dettagli e gli ambiti applicativi ed infine porre in essere le conoscenze acquisite mediante la realizzazione di un progetto.

Alla base di questo studio vi è la SEO, ovvero quell'insieme di pratiche di ottimizzazione di un sito web (e non solo), volte a migliorare la visibilità dei siti ottimizzati sui motori di ricerca.

La tematica scelta risulta essere un argomento di attualità nella società digitale e digitalizzata nella quale ci troviamo, società in cui più informazioni vengono veicolate più *noise*<sup>1</sup> sembra essere prodotto. Si instaura così un circolo vizioso dove più conoscenza disponibile sembra determinare meno sapere collettivo.

Ed è in quest'era di *information overload* che emergono nuovi intermediari volti al filtraggio delle (troppe) informazioni, offrendo alle persone contenuti *ad hoc* i quali indichino loro dove andare in questo *mare magnum* di dati. L'analogia con l'oceano non è casuale, infatti i motori di ricerca navigano le acque del *Web*, tracciando delle rotte sulle quali indirizzare gli utenti.

Bisogna quindi imparare a creare dei contenuti utili ed interessanti non solo per gli umani, ma anche per le macchine, le quali forse si possono definire come le vere protagoniste di quest'era moderna.

Il primo capitolo della tesi è incentrato sullo studio dei motori di ricerca e di Google nello specifico, analizzando il suo algoritmo di indicizzazione dei contenuti; a tal fine si è anche utilizzato un ambiente di simulazione. Inoltre, in questo capitolo si giunge alla determinazione di quali siano le pratiche della SEO.

Il secondo capitolo è invece focalizzato sulla realizzazione di una *web-application* di analisi di siti web in ottica SEO. Se ne analizza l'aspetto grafico, di interazione con l'utente e di funzionamento, esplicando quale sia stata la metodologia di progettazione applicata.

---

<sup>1</sup> Shannon 1948: 381

Il terzo e ultimo capitolo affronta brevemente quali siano i limiti del progetto e le sue potenzialità di miglioramento. Per poter determinare al meglio questi aspetti è stata effettuata l'analisi di un sito web, considerando anche il relativo *feedback* ricevuto dalle *webmaster* del sito preso in esame.

# CAPITOLO 1

## 1.1 Breve definizione di Marketing

*“Markets are conversations.”*<sup>2</sup>

Questa è la prima tesi del Cluetrain Manifesto, un testo di 95 enunciati pubblicato nel 1999 da Rick Levine, Christopher Locke, Doc Searls e David Weinberger, rivolto alle imprese operanti nel nuovo mercato interconnesso grazie ad Internet.

Il manifesto esordisce proclamando che i mercati sono conversazioni tra le persone, le quali, attraverso Internet, possono parlare, dialogare e discutere dei prezzi e della reputazione delle imprese.

La settima tesi del Manifesto sostiene *“Hyperlinks subvert hierarchy.”*<sup>3</sup>, ovvero i link sovvertono la gerarchia della comunicazione impresa-cliente.

Il mercato si rende ancora più orizzontale in confronto all’epoca dei mass-media pre-Internet, quando non era impossibile instaurare questo tipo di connessioni.

In questo paradigma si posiziona il concetto di *marketing*, elaborato per la prima volta dall’economista italiano Giancarlo Pallavicini nel 1959, definendolo come *“quel processo sociale e manageriale diretto a soddisfare bisogni ed esigenze attraverso processi di creazione e scambio di prodotto e valori.”*<sup>4</sup>

Tale concetto è stato poi esteso dalla *American Marketing Association*, la quale definisce il marketing come il *“processo di organizzazione e di esecuzione del concepimento, della politica dei prezzi, delle attività promozionali e della distribuzione di idee, beni e servizi per creare scambi commerciali e soddisfare gli obiettivi degli individui e delle organizzazioni.”*<sup>5</sup>

Per quest’ultima definizione diede il suo contributo anche Philip Kotler, il quale identificò quattro orientamenti al mercato da parte delle imprese:

- **Orientamento alla produzione:** fase che dura dalla Rivoluzione Industriale fino alla metà del Novecento, caratterizzata da una predominanza della domanda sull’offerta, con la conseguenza per le imprese di focalizzarsi sulla riduzione dei costi di produzione, per massimizzare i profitti;

---

<sup>2</sup> Rick 2000: 6

<sup>3</sup> Ibidem

<sup>4</sup> <http://www.giancarlopallavicini.it/economia/marketing/principi-di-marketing>

<sup>5</sup> Ibidem

- **Orientamento al prodotto:** dagli anni 30 del '900 le imprese hanno iniziato a focalizzarsi sulla tecnologia di produzione del prodotto, continuando a mancare di lungimiranza sugli effettivi bisogni del cliente;
- **Orientamento alle vendite:** dagli anni '50 e '60 le imprese cercarono di ottimizzare le proprie vendite evitando sprechi sui beni e servizi prodotti, secondo un'ottica inside-out;
- **Orientamento al marketing:** dagli anni '90 si è sviluppato un approccio outside-in, volto alla comprensione dei reali bisogni del mercato e alla successiva produzione pro-attiva di beni e servizi.

Quest'ultimo orientamento è ancora in corso e si co-evolve insieme alla tecnologia, giungendo quindi alla definizione di Web Marketing, ovvero *“il marketing che sfrutta il canale online per studiare il mercato e sviluppare i rapporti commerciali (promozione/pubblicità, distribuzione, vendita, assistenza alla clientela, etc.) tramite il Web”*.<sup>6</sup>

È in tale *framework* che si posiziona la *Search Engine Optimization* (d'ora in avanti abbreviata in SEO).

## 1.2 Panoramica sui motori di ricerca

Prima ancora di parlare di cosa si occupi la SEO, è bene contestualizzare l'ambiente nella quale è nata e si è sviluppata.

La sua crescita è strettamente legata a quella del *World Wide Web* ed alla crescita esponenziale dei siti disponibili online; infatti, se già nel 1994, tre anni dopo la nascita del *WWW*, si toccò quota 130.000 *websites*, nel 1996 il numero aumentò a 258.000 ed a 2,4 milioni solamente nel 1998.

Ben presto quindi la mole dei siti web disponibili online divenne troppo consistente per poter permettere agli utenti di trovare autonomamente i contenuti desiderati e quindi nacquero i primi motori di ricerca, dei programmi che permettessero di offrire delle risposte alle interrogazioni degli utenti sotto forma di una collezione filtrata di informazioni.

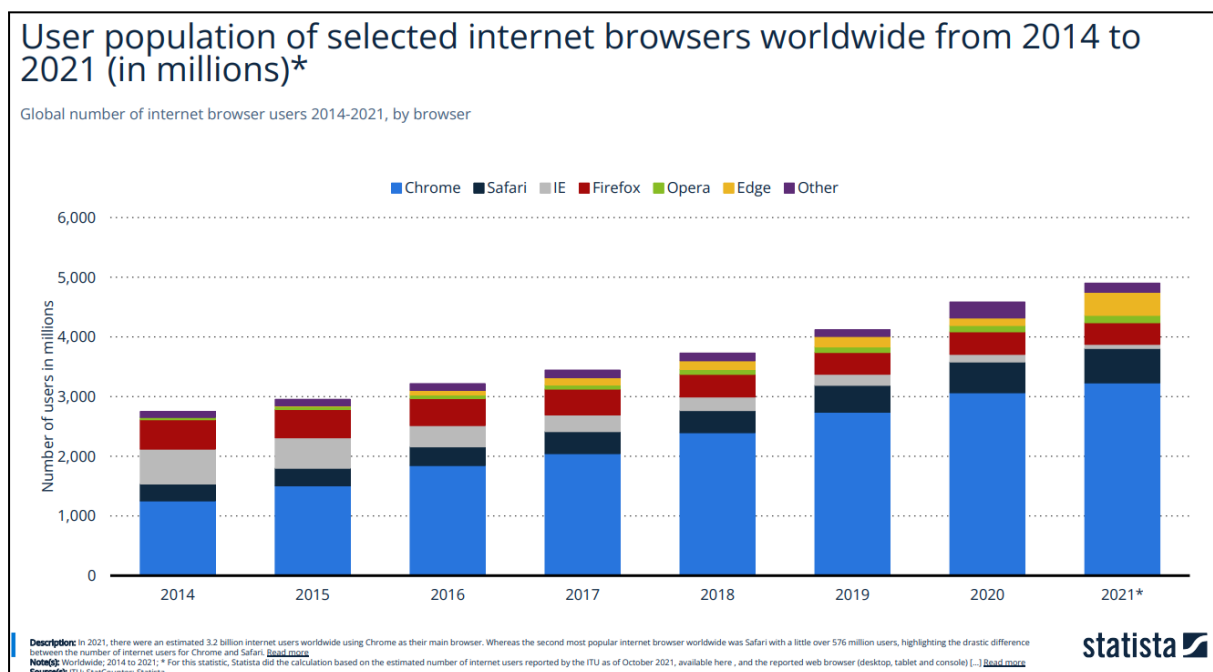
Il successo degli stessi veniva determinato dagli utenti, i quali sceglievano il software che riuscisse maggiormente a restituire dei risultati pertinenti all'argomento di ricerca; ed in tal merito uno spartiacque nella storia dei motori di ricerca venne introdotto da Larry Page e Sergey Brin, i fondatori di Google, con il sistema algoritmico PageRank, il quale classifica la pertinenza di un sito web secondo la

---

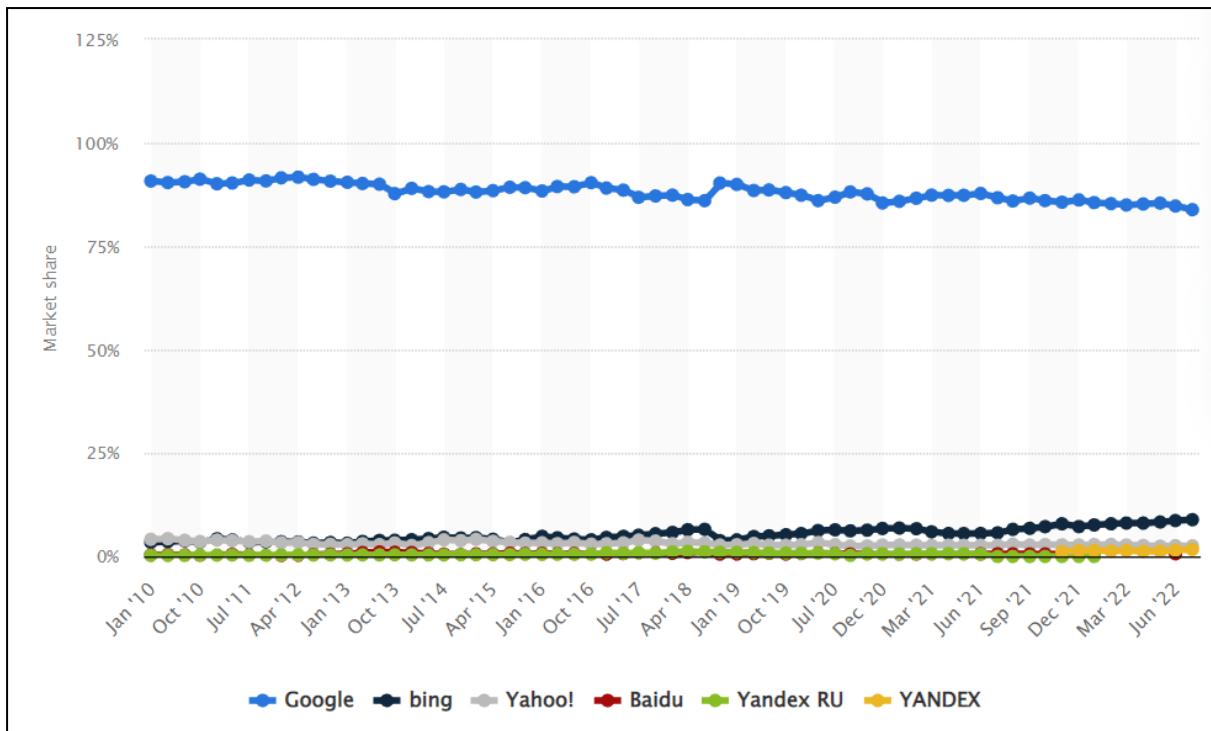
<sup>6</sup> Ibidem

“rilevanza” dello stesso, ovvero un approccio al posizionamento (*ranking*) basato sui suoi contenuti e sui link che puntano ad essa.

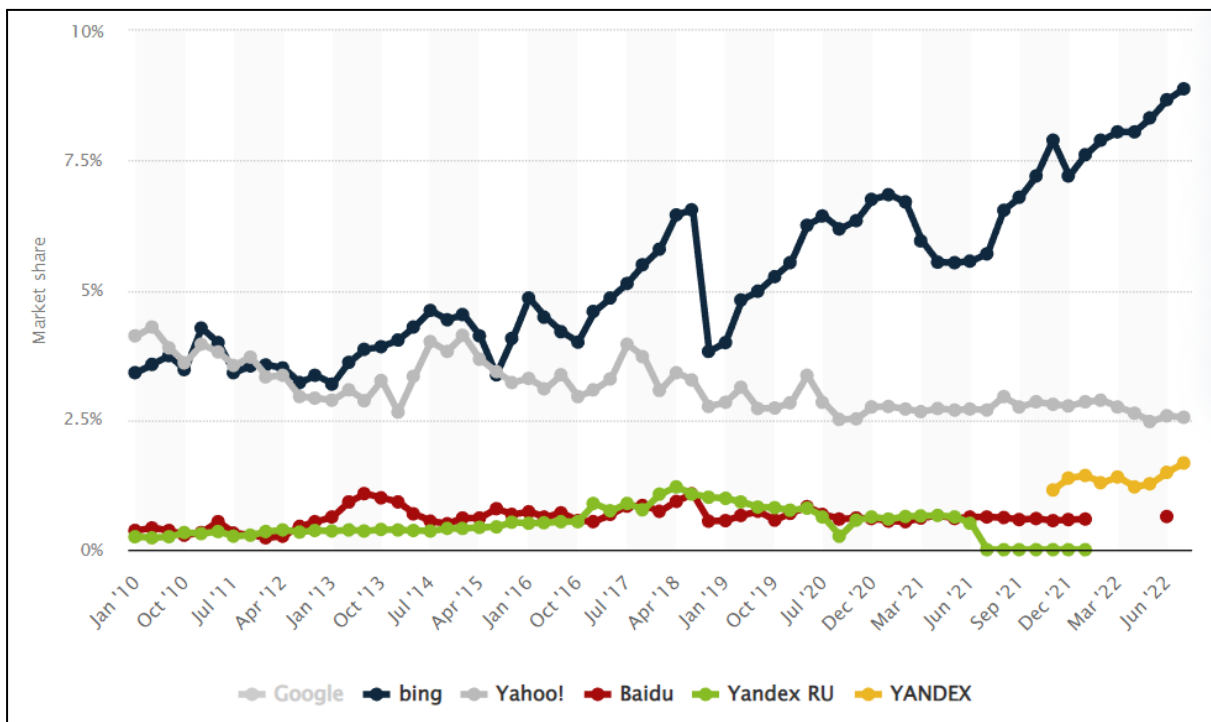
Inizialmente, l'algoritmo PageRank era la parte principale del processo di posizionamento di Google, e forse fu questa scelta vincente che gli permise di diventare uno dei software più usati dagli utenti ed attualmente il motore di ricerca più diffuso, con oltre 3 miliardi di utilizzatori nel mondo (Immagine 1) con uno share del mercato di oltre 83%, contro i suoi principali concorrenti che non superano la soglia del 10% di share (Immagine 2 e 3).



**Immagine 1:** Fonte: Statista.



**Immagine 2:** “Worldwide desktop market share of leading search engines from January 2010 to July 2022.”, focus globale - Fonte: Statista.



**Immagine 3:** “Worldwide desktop market share of leading search engines from January 2010 to July 2022.”, focus sui competitor di Google - Fonte: Statista.



Alla luce di tali dati, in questa tesi si cercherà di focalizzarsi su Google, essendo il principale motore di ricerca adottato dagli utenti a livello globale; tuttavia, è bene ricordare che se talune regole possano essere condivise da più motori, altre sono specifiche per il soggetto di studio.

### 1.3 Le tre fasi della ricerca di Google

Come specificato nella documentazione ufficiale del motore di ricerca <sup>7</sup>, il processo continuo attraverso il quale Google naviga il web e tiene aggiornato i propri *database* è il seguente:

#### 1. Scansione (*Crawling*)

In questa prima fase, Google esplora il web, navigando tra le pagine e scansionando link, testi, immagini e video grazie all'utilizzo di un software automatizzato. Questo programma, denominato genericamente *spider* o *crawler* o, nello specifico di Google, *Googlebot*, tesse una vera e propria rete di collegamento tra i link delle pagine web. In questa fase il proprietario di un sito web può parzialmente indirizzare il *crawler* nella navigazione, attraverso la pubblicazione di *sitemap* o file *robots.txt*, dei quali si parlerà più avanti nel capitolo.

Con il progredire dei linguaggi di programmazione, già da molti anni si è iniziato ad impiegare contenuti dinamici sui siti web, per esempio attraverso l'utilizzo di elementi Javascript; per questo motivo, il bot di Google, nel momento in cui raggiunge una pagina, attende il caricamento di tali contenuti, eseguendo un *rendering* attraverso software che simulino il caricamento della stessa, come succederebbe nella navigazione di un qualsiasi utente, permettendo quindi al bot di poter navigare tra tutti i contenuti presenti.

Tuttavia, per limitare la mole di dati impiegati per scansionare le pagine web, Google ha adottato un parametro denominato *Crawl Budget* <sup>8</sup>. Tale valore, mutabile nel tempo, definisce quanto a fondo e quanto spesso un sito web meriti di essere analizzato dal motore di ricerca. Se un *URL* è popolare e molto richiesto dagli utenti, si avrà un incremento della *crawl demand* e quindi della frequenza di scansione; mentre se un sito web è veloce a rispondere alle interrogazioni del *crawler*, senza pregiudicare la velocità delle connessioni parallele dell'esperienza utente, si avrà un *crawl rate* maggiore e quindi una maggiore velocità di scansione.

---

<sup>7</sup> <https://developers.google.com/search/docs/fundamentals/how-search-works>

<sup>8</sup> <https://www.seozoom.it/crawl-budget-seo-significato-ottimizzazione/>

## 2. Indicizzazione (*Indexing*)

In questa fase, Google cerca di capire di cosa si stia parlando nella pagina analizzata. Raccogliendo dati come ad esempio *Title tag* o *Alt tag* delle immagini, il *crawler* attribuisce un significato semantico ai dati raccolti, salvandoli in un *database* dal quale verranno estratte tali informazioni in base alla *query* di ricerca dell'utente. Come specifica la documentazione, non tutte le pagine scansionate vengono indicizzate, questo per motivi di contenuti oppure per rilevazione di pratiche ingannevoli, delle quali si parlerà successivamente.

## 3. Pubblicazione dei risultati di ricerca (*Ranking*)

Questa è l'ultima fase del processo e coincide con l'output di ricerca dell'utente. Infatti, a seguito di una domanda di ricerca, Google impiega diversi fattori per restituire una collezione di risultati il più possibile pertinenti a quanto richiesto. Maggiore è la corrispondenza tra i dati raccolti nelle due fasi precedenti con i criteri impiegati dal motore, migliore sarà il punteggio di *ranking* e di conseguenza migliore sarà il posizionamento nella pagina dei risultati.

Nei prossimi paragrafi si cercherà di analizzare quale sia il procedimento di *ranking* impiegato da Google, illustrando l'evoluzione dal semplice utilizzo di calcoli matematici al più complesso processo di analisi di anche fattori qualitativi; scoprendo infine quali migliorie si possano adottare per perfezionare tale parametro.

### 1.4 L'algoritmo PageRank di Google

*"Ipotezziamo che la pagina A abbia pagine T1...Tn che puntano ad essa (cioè, sono citazioni). Il parametro d è un fattore di smorzamento che può essere impostato tra 0 e 1. Di solito impostiamo d a 0,85. [...] Inoltre, C (A) è definito come il numero di link che escono dalla pagina A. Il PageRank di una pagina A è dato come segue:*

$$PR(A) = (1-d) + d (PR(T1)/C(T1) + ... + PR(Tn)/C(Tn))$$

*Tieni presente che i PageRank formano una distribuzione di probabilità sulle pagine web, quindi la somma dei PageRank di tutte le pagine web sarà uno".*<sup>9</sup>

---

<sup>9</sup> Tradotto da Brin, Page 1998: 109

Riscrivendo la formula dell'algoritmo nella sua versione più recente, si giunge al seguente risultato:

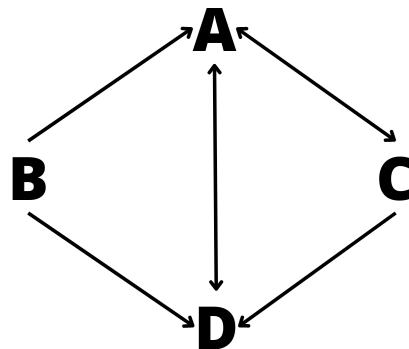
$$PR[A] = \frac{1-d}{N} + d \left( \sum_{k=1}^n \frac{PR[P_k]}{C[P_k]} \right)$$

Dove:

- $PR[A]$  è il valore PageRank della pagina A da determinare;
- $N$  è il numero totale delle pagine che compongono il sito web;
- $n$  è il numero totale delle pagine che contengono almeno un link in entrata verso la pagina A;
- $PR[P_k]$  è il valore PageRank di ogni pagina  $P_k$ . Con l'evolversi dell'algoritmo negli anni, il valore PageRank iniziale di ogni pagina è 0,25;
- $C[P_k]$  è il numero totale di link presenti in ogni pagina  $P_k$ ;
- $d$  è il fattore di smorzamento.

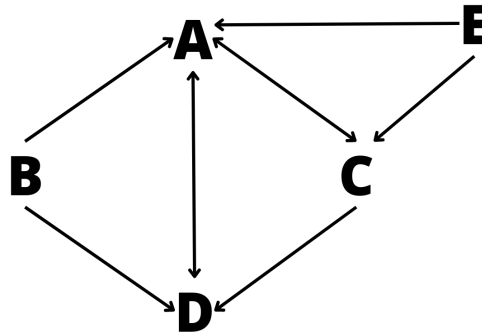
Si ipotizza ora il seguente esempio:

Dato un sito web con le pagine A, B, C e D, queste sono tra di loro collegate nella seguente maniera:



Applicando l'algoritmo Pagerank alla pagina A, partendo da dei valori PageRank di default per le altre pagine (equivalente a 0,25), si ha come risultato un valore arrotondato di 0,26.

Ipotizzando ora l'aggiunta di un'ulteriore pagina E, i nodi tra di loro risultano collegati nella seguente maniera.



Calcolando nuovamente il valore PageRank della pagina A, si ha ora come risultato 0,37.

Da questo semplice esempio si deduce che all'aumentare dei link in entrata verso una pagina, questa otterrà un valore PageRank maggiore.

È su questo assunto che si basa il ragionamento dietro l'algoritmo: una pagina è tanto più rilevante quante sono le altre pagine che si collegano ad essa.

Ciò però determina un problema, scaturito nei primi periodi di vita dell'algoritmo. Un utente può creare una moltitudine di pagine web con dei link in entrata verso la pagina desiderata oppure potrebbe convincere altri utenti ad aggiungere dei link sul proprio sito che indirizzino verso la pagina desiderata, con l'unico scopo di aumentarne il punteggio.

I fondatori di Google aggirarono la difficoltà aggiungendo un valore pesato ai link, e tale valore era determinato dal valore di PageRank delle pagine stesse; perciò un collegamento proveniente da una pagina autorevole, rendeva la pagina di destinazione più autorevole rispetto a quanto potessero fare più pagine con poca autorevolezza.

Tuttavia, la vulnerabilità di questo ragionamento sta nella sua stessa ciclicità: per essere autorevole, un sito deve essere collegato a siti autorevoli.

Per questo motivo l'algoritmo si basa anche su di un modello che permette la determinazione di un valore PageRank in seguito a diverse iterazioni di calcolo: il *Random Surfer Model*.

Tale modello viene utilizzato per determinare la possibilità per un utente di finire in maniera randomica su una pagina web, attraverso la navigazione dei link oppure direttamente dall'URL.

Il modello ipotizza quindi un *surfer* il quale, una volta giunto in un nodo, può seguire i link oppure raggiungere in maniera casuale un nodo diverso del grafo; il reset della sua posizione è necessario per poter visitare l'interezza della rete, in quanto potrebbero esserci dei cluster di nodi non collegati con il resto della rete e quindi irraggiungibili con una navigazione diretta tra link.

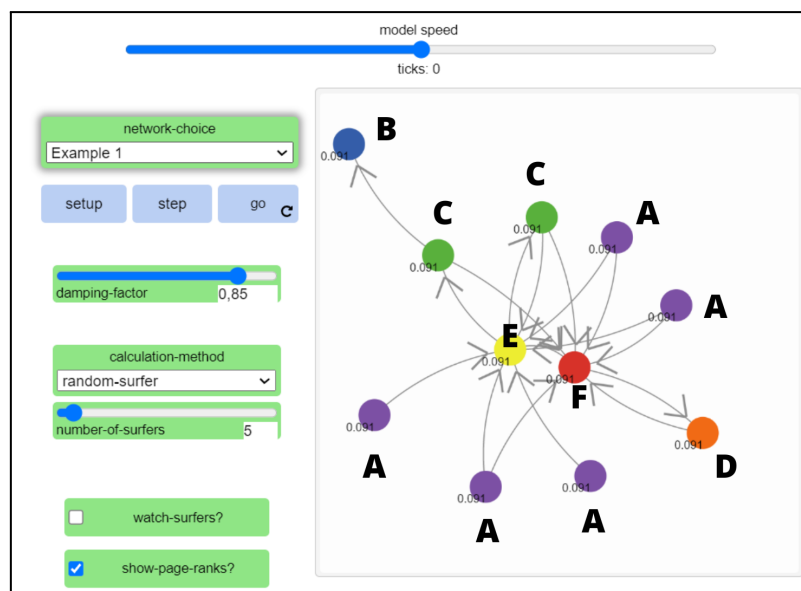
La discriminazione tra queste due possibili azioni varia in base al tipo di modello che si sceglie: se "*Step Walk With Self-Loop*", il *surfer* ad ogni iterazione del

modello ha una probabilità  $p$  di rimanere nel nodo e seguire i collegamenti ed una probabilità  $1 - p$  di essere spostato in un nodo casuale della rete; se invece “*Random Walk With Coin Flips*” ad ogni iterazione il surfer ha una probabilità fissa del 50% (*coin flip*) di rimanere oppure essere spostato. Dopo un certo numero di iterazioni, si determinerà l'importanza relativa del nodo in seguito a quante volte il surfer è giunto sullo stesso durante la sua navigazione.

Negli anni Google ha sempre aggiornato i suoi criteri di calcolo di posizionamento, giungendo recentemente all'introduzione del *Reasonable Surfer Model*, una revisione del modello precedente che tiene conto, nella determinazione della possibilità di azione del surfer, di un'ampia gamma di fattori, anche relativi alla *user experience* e all'accessibilità. Nella fattispecie, e anche più verosimilmente alla realtà dell'esperienza utente, più una pagina web rispetta delle linee guida di progettazione, maggiore sarà la possibilità che l'utente la navighi e minore quindi sarà la possibilità che questo rimbalzi su un altro risultato di ricerca in un'altra pagina web.

Per comprendere quindi in via definitiva il funzionamento dell'algoritmo nella sua interezza è stato utilizzato Netlogo, un ambiente di simulazione online che mette a disposizione diversi modelli di simulazione, tra cui il PageRank.

La simulazione è così composta:



**Immagine 4:** Interfaccia dell'ambiente di simulazione al setup - Fonte: Netlogo.

L'interfaccia nell'immagine 4 mostra sulla sinistra i parametri di simulazione e sulla destra l'ambiente della stessa. Dopo aver impostato il *damping-factor* a 0,85 come

da letteratura, si sceglie il metodo di calcolo *random-surfer*, impostando poi 5 *surfers* per rendere il procedimento di classificazione più veloce.

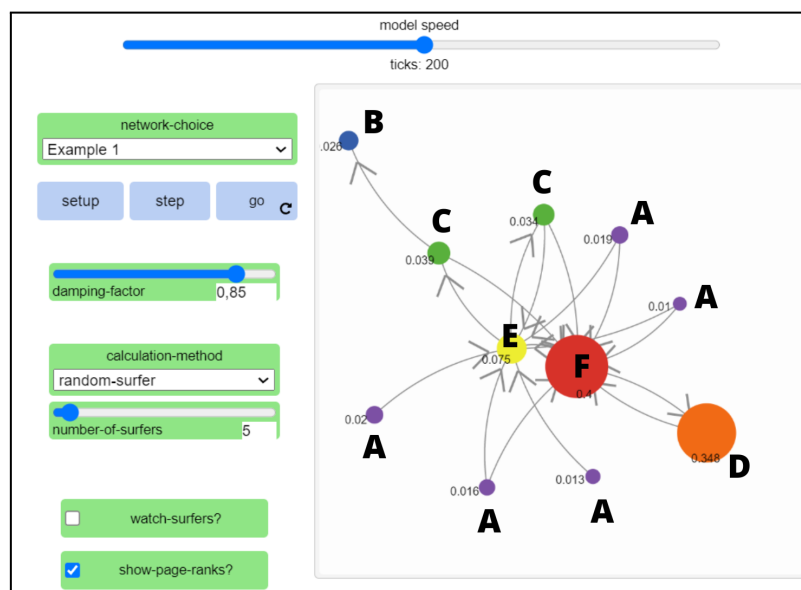
Avviando il setup dell'ambiente si possono notare le seguenti informazioni:

- Gli  $n$  nodi vengono inizializzati con uno stesso punteggio standard  $P$ , tale per

$$\text{cui } \sum_{k=1}^n P_k = 1$$

- I nodi A presentano solamente link in uscita;
- Il nodo B presenta solamente link in entrata;
- I nodi C presentano link in uscita verso i nodi E/F e link in entrata dal nodo E;
- Il nodo D presenta un link in entrata ed in uscita verso il nodo F;
- Il nodo E presenta link in entrata dai nodi A/C/F e link in uscita verso i nodi C/F;
- Il nodo F presenta link in entrata dai nodi A/C/D/E/F e link in uscita verso il nodo D.

A seguito di diverse iterazioni (200 in questo caso), l'ambiente raggiunge una distribuzione invariante e presenta i seguenti risultati:



**Immagine 5:** Interfaccia dell'ambiente di simulazione dopo 200 iterazioni di calcolo - Fonte: Netlogo.

Dai valori di output dell'algoritmo si può osservare quanto segue:

- I nodi con valore PageRank minore sono i nodi A in quanto non hanno nessun link in entrata;

- Il nodo B presenta un valore lievemente maggiore dei nodi A in quanto possiede un link in entrata dal nodo C;
- I nodi C hanno un valore maggiore del nodo B poiché, nonostante entrambi posseggono un solo link in entrata, i nodi C sono collegati ad un nodo di importanza maggiore (E);
- Il nodo E possiede il punteggio maggiore tra i nodi finora analizzati in quanto avente un valore di centralità maggiore ed essendo molto collegato ad altri nodi;
- Il nodo D presenta un valore nettamente superiore al valore del nodo E, essendo tuttavia collegato al solo nodo F. La spiegazione di quanto accade è da attribuire al nodo F stesso, il quale, essendo il nodo con maggior valore PageRank in termini assoluti, trasmette la sua importanza all'unico nodo al quale è collegato: il nodo D. Come già detto in precedenza, l'algoritmo di Brin e Page si fonda sul valore di importanza e perciò avere un collegamento con un nodo importante fornisce risultati migliori rispetto ad essere collegati con molti nodi poco importanti;
- Il nodo F possiede il valore maggiore tra tutti i nodi, in quanto nodo centrale di tutto il grafo e collegato pressoché a tutti gli altri nodi.

Da quanto osservato si può evincere che l'algoritmo PageRank sia un processo stocastico markoviano, ovvero un procedimento di calcolo dello stato di un sistema dipendente dallo stato immediatamente precedente del sistema stesso e non influenzato da come si è giunti a quella stessa condizione.

Infatti, una pagina web per divenire importante necessita sia di tempo, sia di processi (es. *link building*) volti all'aumento della sua reputazione o *link juice*.

Non importa se alla sua creazione, la pagina web fosse poco importante, importa quale sia la sua situazione nel presente del calcolo da parte del motore di ricerca e di come questa possa influenzare e farsi influenzare da altre pagine web.

## 1.5 Aggirare l'algoritmo

Fino al 2011 Google Toolbar è stata la *toolbar* del motore di ricerca omonimo e tra le varie funzionalità che questa offriva agli utenti era anche presente un visualizzatore del valore PageRank della pagina che si stava visitando in quel momento.

Tale funzionalità restituiva un valore di popolarità tra un minimo di 0 ed un massimo di 10. Gli utenti meno esperti, tuttavia, fraintesero questa metrica, pensando che un valore di PageRank alto equivallesse ad ottenere i primi posti nel posizionamento nei risultati di ricerca. Il PageRank era sì, una delle metriche principali di Google, ma non l'unica; i *webmaster* che riuscirono ad entrare in quest'ottica olistica del motore di ricerca non adottarono un insieme di pratiche finalizzate al temporaneo raggiungimento di migliori risultati, dal momento che ad un aumento del valore di

PageRank non corrispondeva un automatico miglior posizionamento nei risultati di ricerca.

Tale idiosincrasia era determinata dal fatto che queste pratiche, una volta identificate dal motore di ricerca, penalizzavano chi le compiva, con provvedimenti più o meno gravi, determinando quindi la perdita di popolarità acquisita dalle pagine.

Queste tecniche volte alla manipolazione dell'algoritmo o allo sfruttamento delle sue vulnerabilità sono denominate Black Hat SEO; alcune delle più comuni e riscontrate negli anni sono le seguenti:

- ***Link Farm***

Sono letteralmente delle “fabbriche di link”, ovvero uno o più network di siti web ben ottimizzati ma dal basso valore per gli utenti e che trattano dei temi più disparati, passando dallo sport all'economia o dall'attualità politica alle previsioni meteorologiche, con l'unico scopo di generare link in entrata verso i siti web che si vogliono ottimizzare.

- ***PageRank Spoofing***

Il valore PageRank della Google Toolbar era facilmente manipolabile ed attraverso un semplice *HTTP Redirect 302* era possibile trasferire il valore di popolarità da una pagina ad un'altra. In questo modo, la pagina contenente il codice di *redirect* poteva clonare il valore di autorevolezza di una pagina bersaglio, come ad esempio la *homepage* di Google, la quale possiede un punteggio massimo.

- ***Keyword Stuffing***

Era una pratica molto comune in passato, che consisteva nel riempire il testo di una pagina web di moltissime *keyword* per le quali essa intendeva posizionarsi. Spesso, i testi di queste pagine web perdevano qualsiasi significato per l'utente, in quanto piene di incongruenze semantiche e grammaticali.

In alternativa era possibile anche trovare siti web che avessero zone di testo dello stesso colore dello sfondo, riempite anch'esse di *keyword*, quindi invisibili per l'utente ma non per i motori di ricerca.

- ***Cloaking***

Tecnica che permette di ingannare il motore di ricerca nella fase di indicizzazione del sito web, mostrando una versione differente rispetto a quella che realmente vedrà l'utente, ottenendo così un miglior posizionamento.

- ***Pagine Doorway***

È una rete di pagine senza un vero e proprio contenuto, che servono per spingere utenti e motori di ricerca verso la navigazione di un sito web bersaglio.



- **Backlink nascosti**

Si riferisce alla presenza di *backlink* nascosti verso una o più pagine bersaglio. Tali link vengono nascosti nel *footer* oppure all'interno del testo, mimetizzandoli con un colore uguale al resto del testo e non evidenziandoli al passaggio del puntatore.

- **Compravendita e scambio di link**

Consiste nell'utilizzo di siti intermediari per posizionare dei link in entrata o in uscita.

- **Article Spinning**

Dal momento che i motori di ricerca penalizzano i contenuti clonati, questa pratica consiste nel parafrasare (anche in maniera automatizzata attraverso dei software) il contenuto di una pagina web e riscriverlo in maniera diversa in altre pagine.

- **Cybersquatting e Desert Scraping**

Si tratta di pratiche simili alla precedente. Nel caso del *cybersquatting* un utente acquista dei domini scaduti di siti web famosi, già indicizzati dai motori di ricerca e quindi con un'alta autorevolezza, per poi inserire i propri contenuti. Quando si parla di *desert scraping* invece, si intende riutilizzare dei contenuti di siti web con domini scaduti, apportando leggere modifiche.

Per questi ed altri motivi, Google decise di ritirare la propria *toolbar* in modo tale che gli utenti non utilizzassero sotterfugi per apparire nei primi posti, ma si concentrassero piuttosto sulla produzione di qualità di siti web. Questa politica aziendale che Google decise di adottare si è potuta notare anche nei suoi aggiornamenti più importanti, i quali si sono concentrati sempre maggiormente a premiare la qualità di produzione di un contenuto online.

## 1.6 Gli aggiornamenti di Google

L'algoritmo di ricerca di Google si è evoluto negli anni attraverso centinaia di aggiornamenti, gran parte dei quali rilasciati in via confidenziale e solamente alcuni di essi in via ufficiale <sup>10</sup>. Analizzando gli aggiornamenti più importanti rilasciati fino ad oggi, si può tracciare una interessante linea di *best practices* che Google ha costruito negli anni e che è bene sapere per capire quale sia il campo di gioco per la SEO.

---

<sup>10</sup> Joshi, Patel 2018: 10

- **Panda (2011)**

Si tratta di uno dei primi aggiornamenti più rivoluzionari. Con Panda, infatti, Google applicò un filtro alle pagine web assegnando un *quality score* che penalizzasse tutti i siti web con contenuti non utili agli utenti come contenuti qualitativamente bassi o duplicati, *keyword stuffing*, plagio, spam, ecc.

- **Penguin (2012)**

Si tratta di un aggiornamento nato con lo scopo di combattere le pratiche di Black Hat SEO. In caso di provvedimenti, la penalizzazione sarebbe stata circoscritta alla pagina web e non all'intero sito.

- **Pirate (2012)**

Con questo *update* si iniziarono ad applicare provvedimenti contro siti web che utilizzassero contenuti coperti da diritto d'autore, violando le normative sul copyright.

- **Hummingbird (2013)**

Lo slogan per questo aggiornamento era "*fast and precise*" <sup>11</sup>. Con Hummingbird, Google decise di introdurre un algoritmo predittivo che tenesse conto di oltre 200 fattori per il posizionamento dei siti web e che decifrasse il linguaggio naturale dell'utente e la sua comprensione del testo in fase di ricerca, restituendo dei risultati che non per forza includessero le keyword usate nella query di ricerca, quanto piuttosto dei risultati che l'utente si sarebbe aspettato.

- **Pigeon (2014)**

Questo fu un update sensazionale per la Local SEO. Esso ottimizzò i risultati di ricerca in relazione alla posizione dell'utente, cercando di offrire delle informazioni quanto più geolocalizzate possibili, senza bisogno di specificare alcun luogo nelle query.

- **Aggiornamento HTTPS/SSL (2014)**

Questo aggiornamento andò a favorire i siti web che utilizzassero protocolli di rete sicuri e criptati, definendo l'inizio di uno standard che ogni webmaster avrebbe poi dovuto iniziare a praticare.

- **E-A-T (2015)**

Si tratta di un sistema di posizionamento delle pagine web secondo i criteri di:

- *Expertise*: misura l'esperienza di chi offre dei contenuti in base alla propria attività lavorativa.

---

<sup>11</sup> Ivi: 11

- *Authoritativeness*: metrica che si riferisce all'autorevolezza e non all'autorità. Con questo Google intende premiare quei siti web che, in base al loro contenuto e dominio, sono capaci di essere ritenuti esperti in determinate materie dagli utenti.
- *Trustworthiness*: ovvero affidabilità; i contenuti devono essere veritieri e non *fake news* o informazioni poco accurate.

Le pagine trattanti temi sensibili per il motore di ricerca, come economia, salute, politica, felicità, ecc. vennero denominate “YMYL” (*Your Money Your Life*) e Google dimostrò di essere molto intransigente verso il loro posizionamento.

- **Aggiornamento *Mobile Friendly* o “*Mobilegeddon*” (2015)**

Il soprannome che tale aggiornamento si guadagnò è abbastanza esplicativo. Con esso, infatti, Google decise di premiare i siti web ottimizzati per la versione *mobile*, dal momento che i dati mostravano come la maggioranza degli utenti navigasse da cellulare. Di conseguenza, i siti web che non implementarono tale *feature* furono declassati nella pagina dei risultati di ricerca.

- ***RankBrain* (2015)**

È un sistema di apprendimento automatico che aiuta il motore a comprendere il significato delle query e a fornire i risultati di ricerca con la migliore corrispondenza. Era parte dell'aggiornamento Hummingbird, ma in seguito a modifiche dell'intelligenza artificiale fu posticipato. Secondo Google è il terzo fattore di ranking più importante.<sup>12</sup>

- ***Possum* (2016)**

Con questo aggiornamento si migliorò ulteriormente quanto introdotto con Pigeon, offrendo risultati più vicini alla posizione dell'utente a discapito di quelli più lontani, cercando però anche di aiutare i *business* localizzati fuori dai centri delle città.

- ***Fred* (2017)**

Questo aggiornamento andò a ridimensionare l'autorità di quei siti web generalisti, lo scopo dei quali era monetizzare tramite le pubblicità e l'elevato traffico dato dai molti temi trattati. Google non volle punire i siti generalisti *in toto*, in quanto è legittimo possedere dei blog che trattano di più argomenti, quanto piuttosto i siti web che offrivano contenuti di basso valore per gli utenti, *fake news* oppure pubblicità troppo invasiva, ma che data la moltitudine di articoli pubblicati, venivano posizionati comunque nei primi risultati.

---

<sup>12</sup> Ibidem

- **Aggiornamento *Mobile First Indexing* (2018)**

Con questo update Google volle dare sempre maggior importanza al dato che mostrava gli smartphone come *device* di navigazione primario degli utenti, andando ad indicizzare e scansionare prima di tutto la versione *mobile* dei siti web.

- ***Speed Update* (2018)**

Google dichiara che la velocità di caricamento di un sito viene considerata un fattore di posizionamento. Tutti i siti che impiegano più di 2-3 secondi per caricare vengono quindi penalizzati, in tal modo si favorisce la *user experience* (UX).

- ***Bert* (2019)**

La sigla significa *Bidirectional Encoder Representations from Transformers*, ovvero un sistema di elaborazione del *Natural Language Processing* (NLP) basato sul concetto di reti neurali. Con Bert, Google continua la sua evoluzione di un algoritmo di *machine learning* volto alla comprensione delle intenzioni dell'utente, iniziato con Hummingbird e proseguito con RankBrain.

- ***Core Web Vitals* (2021)**

Vengono introdotti tre nuovi parametri di posizionamento volti al miglioramento della *user experience*:

- *Largest Contentful Paint* (LCP): metrica che analizza il *rendering* del contenuto mostrato più grande, presupponendo che esso sia quindi l'elemento più importante. Serve per identificare se il contenuto di una pagina sia effettivamente un elemento utile all'utente oppure un elemento di disturbo come un pop-up o una pubblicità.
- *Cumulative Layout Shift* (CLS): misura lo spostamento degli elementi del *layout* che avvengono all'improvviso, dovuti, per esempio, al *rendering* successivo di una *ad*.
- *First Input Delay* (FID): ovvero il tempo trascorso tra la prima interazione dell'utente e la risposta della pagina web.

## 1.7 I 200 fattori di posizionamento di Google

Come si è potuto notare, Google ha implementato diversi criteri negli anni, passando da un primo algoritmo, prettamente focalizzato sull'aspetto matematico di autorevolezza, all'utilizzo di diversi algoritmi, i quali valutano al contempo diversi fattori qualitativi e quantitativi.

Attualmente il motore di ricerca applica oltre 200 criteri di analisi delle pagine web per piazzarle nei risultati di ricerca; alcuni di questi fattori sono stati resi pubblici dalla stessa azienda o da suoi portavoce, mentre altri risultano più controversi e frutto di analisi e speculazioni dei *webmaster* e *SEO specialist*.

Appare dunque impossibile definire con esattezza quali siano le chiavi che permettano di decifrare l'algoritmo di posizionamento di Google; tuttavia, è possibile raggruppare questi oltre 200 fattori in 9 macroaree:

- **Fattori di dominio**

Sono informazioni come l'anzianità del dominio, la durata della sua registrazione, uno storico di quanto accaduto (eventuali precedenti penalizzazioni), *keyword* presenti, informazioni sul servizio di registrazione, *country code* (per direzionare meglio il traffico degli utenti verso pagine del loro stesso paese).

- **Fattori a livello di pagina**

È una delle macroaree più sostanziose; infatti, se per Google la qualità è diventato un principio cardine di posizionamento, è altresì vero che una pagina deve possedere alcuni requisiti tecnici indispensabili per poter essere indicizzata, come per esempio *tag* HTML, *keywords*, gestione degli errori, presenza di una versione per *smartphone* e molti altri. Molti di questi criteri si posizionano nella cosiddetta SEO Tecnica, di cui si parlerà nel prossimo capitolo.

- **Fattori a livello di sito**

Riguardano la gestione delle risorse di un sito web, come per esempio la presenza di una tassonomia nell'architettura delle pagine web, la presenza di *sitemap* e file *robots.txt*, l'utilizzo di Google Analytics, Google Search Console e di un menù di navigazione *breadcrumb*.

- **Fattori relativi ai *Backlink***

Nonostante l'introduzione di diversi parametri legati alla UX, i link sono rimasti dei criteri essenziali per il posizionamento delle pagine web in quanto, tuttora, sono il principale canale di navigazione per poter scansionare il Web. La cosiddetta *Link Building* è la pratica di creazione di una rete di collegamenti verso e da il proprio sito web e attualmente resta una delle principali strategie SEO.

- **Interazioni degli utenti**

In seguito all'introduzione di *RankBrain*, Google ha iniziato a classificare le pagine web in funzione della loro rilevanza per ogni *query* di ricerca, attraverso anche l'uso di intelligenze artificiali. In quest'area si posizionano fattori quali il CTR (*click through rate*), *bounce rate*, *dwell time*, quanti utenti hanno aggiunto una determinata pagina ai preferiti, la navigazione organica su un sito web e l'eventuale presenza di commenti degli utenti.

- **Regole speciali dell'algoritmo di Google**

Questa è la macroarea più aperta a dubbi e speculazioni in quanto riguarda dei fattori coperti dal segreto aziendale. Alcuni dei criteri identificati dai professionisti del settore sono ad esempio la predilezione di nuovi contenuti rispetto a quelli vecchi come risposta alla *query* dell'utente, degli atteggiamenti più restrittivi per le pagine YMTL, un utilizzo dei dati di navigazione per ripresentare delle pagine già visitate dall'utente come risultato se pertinenti, ecc.

- **Segnali del *brand***

Google pare tenga in considerazione i contenuti prodotti dai diversi *brand* per esempio attraverso la rilevazione della presenza di canali *social*, informazioni sulla sede legale dell'attività, correlazione tra *keyword* di posizionamento e nome del marchio, ricerche *branded*, ecc.

- **Fattori di *webspam On-site* e Fattori di *webspam Off-site***

Queste ultime due categorie fanno parte di una lunga lista di azioni che possono essere classificate come pratiche di *Black Hat SEO* o *Gray Hat SEO*. Nel primo caso, come è già stato spiegato nelle pagine precedenti, si identificano un insieme di comportamenti scorretti, volti ad un apparente ed immediato aumento dei valori di autorevolezza di un sito web, mentre nel secondo caso si indicano tutte quelle pratiche non propriamente classificate come illecite da Google, ma che in un modo o nell'altro cercano di velocizzare il processo di incremento della popolarità di un sito, talvolta ricorrendo ad *escamotage*.

Ambedue le categorie di fattori portano quindi alla luce tutta una serie di provvedimenti disciplinari che il motore di ricerca applica per penalizzare coloro che adottano queste pratiche, disincentivando qualsiasi azione volta ad "ingannare" l'algoritmo.

## 1.8 Le penalizzazioni

Come già accennato, le penalizzazioni di Google sono dei provvedimenti disciplinari applicati all'insorgere di atti contrari alle *guidelines* del motore di ricerca.

Esse si possono classificare in due categorie:

- **Penalizzazioni manuali**

Nel primo caso si parla di penalizzazioni applicate direttamente da un *team* di controllo di Google che si occupa di rivedere singolarmente i siti web contrassegnati come potenzialmente sanzionabili.

- **Penalizzazioni algoritmiche**

Nel secondo caso si classificano dei processi automatizzati di ricalcolo del posizionamento di una pagina web, a seguito di tentativi del motore di ricerca di offrire risultati sempre più utili in relazione alle richieste degli utenti.

Per tale motivo, in questa categoria si posizionano anche i cali di traffico dovuti a dei nuovi metodi di calcolo dell'algoritmo, causati principalmente da nuovi aggiornamenti. Queste, tuttavia, non sono delle vere e proprie penalizzazioni, quanto più un ragionamento che Google applica considerando negativamente dei siti web basati su *guidelines* obsolete e quindi non meritevoli dei primi posti in classifica.

Le sanzioni che Google può applicare possono essere più o meno severe in base alla gravità che esso stesso rileva. Si può passare da azioni su determinate *keyword* per le quali il sito si è posizionato, passando a provvedimenti su alcuni URL o sull'intero sito web fino alla sua completa de-indicizzazione.

A tal proposito, si cita uno studio condotto da *Backlinko.com*<sup>13</sup> riguardo al tasso di *click* in funzione del posizionamento nei risultati di ricerca. Non sorprende scoprire che i primi tre risultati di ricerca ottengono oltre il 50% del totale dei *click* per la navigazione organica, più della metà dei quali (27,6%) appartiene solo al primo link. L'analisi prosegue mostrando come, sostanzialmente, l'aumento del CTR è esponenzialmente maggiore più ci si avvicina ai primi posti e, di conseguenza, esponenzialmente minore man mano che ci si allontana, crollando ad un 2,4% solamente al 10° posto in classifica. È chiaro, quindi, come il passare solamente alla seconda pagina dei risultati di ricerca determini una riduzione di traffico non irrilevante, causando una diminuzione del 75% della possibilità che un utente possa navigare su un sito web.

---

<sup>13</sup> <https://backlinko.com/google-ctr-stats>

Dati questi numeri è semplice capire come anche la più piccola penalizzazione possa determinare grandi effetti negativi per il traffico di un sito web, soprattutto per il posizionamento di *keyword* molto generiche. Per ovviare a questo problema un sito web penalizzato può decidere di posizionarsi per *long-tail keyword*, ovvero *query* di ricerca con più caratteri e quindi più specifiche e questo, come mostra la ricerca, può determinare un aumento di circa il 30% delle possibilità di *click*.

Il sistema di penalizzazioni si può quindi definire legittimo?

Una piattaforma come Google, la quale si è visto detenere oltre l'80% dello *share* di mercato delle ricerche online globali, è un motore di ricerca come altri, oppure dovrebbe considerarsi come un *gatekeeper* capace di filtrare i contenuti del web?

Questa ampia diffusione tra gli utenti dovrebbe determinare una maggiore regolamentazione del servizio?

*“Nonostante la sua [di Google ndr] natura ipertestuale consenta potenzialmente infiniti percorsi di lettura, le rotte di navigazione sul web sono di fatto fortemente condizionate dai risultati forniti dai motori di ricerca. [...] le pagine escluse diventano tanto più nascoste e inaccessibili quanto più si rafforza, nell'esperienza soggettiva degli utenti, la funzione dei motori come esclusive porte di accesso all'informazione.”*<sup>14</sup>

Questi quesiti sono simili a quelli che Dave Yost, procuratore generale dell'Ohio, si è posto a Giugno 2021, intentando una causa verso Google chiedendone la definizione come un'utilità pubblica soggetta alla regolamentazione dello Stato.

Yost cita gli studi condotti dalle società di ricerche sul mercato SparkToro<sup>15</sup> e The Markup<sup>16</sup>, sostenendo sul *The New York Times*<sup>17</sup> che almeno il 65% delle ricerche Google nel 2020 si fossero concluse con zero *click*, ovvero gli utenti si sarebbero accontentati dei risultati di ricerca e delle anteprime di quanto scritto nelle pagine web, senza tuttavia, mai aprire alcun link.

Il procuratore dell'Ohio quindi sostiene che il motore dovrebbe considerarsi come una utilità pubblica, dato l'elevato traffico di utenti utilizzatori del servizio, per poter *“offrire agli altri una possibilità migliore”*<sup>18</sup> e sperare in un mercato online *“più competitivo”*<sup>19</sup>.

Seppur le intenzioni di Yost possano risultare condivisibili, sia lo stesso *New York Times*<sup>20</sup> sia *Wired*<sup>21</sup> definiscono la causa come una “forzatura”, ricordando che per

---

<sup>14</sup> Paccagnella 2020: 217

<sup>15</sup> <https://sparktoro.com/blog/in-2020-two-thirds-of-google-searches-ended-without-a-click/>

<sup>16</sup> <https://themarkup.org/google-the-giant/2020/07/28/google-search-results-prioritize-google-products-over-competitors>

<sup>17</sup> <https://www.nytimes.com/2021/07/07/opinion/google-utility-antitrust-technology.html>

<sup>18</sup> Ibidem

<sup>19</sup> Ibidem

<sup>20</sup> <https://www.nytimes.com/2021/06/08/technology/google-ohio-public-utility.html>

<sup>21</sup> <https://www.wired.com/story/no-facebook-google-not-public-utilities/>



pubblica utilità si definisce un'azienda con un contratto governativo con lo scopo di erogare un bene o un servizio.

L'errore che Yost commette è quello di confondere un servizio pubblico per un valore pubblico, inteso come *“il valore che un'organizzazione apporta alla società a beneficio del bene comune.”*<sup>22</sup>.

Infatti, *“nella platform society, la creazione del valore pubblico finalizzato al bene comune è spesso confusa con la creazione di valore economico.”*<sup>23</sup>; tale valore economico, come scrivono Van Dijck, et. al., è determinato dall'efficienza della piattaforma stessa.

Una piattaforma funziona se riesce a connettere quanti più attori sociali possibili, datificando e filtrando le attività degli stessi e offrendo loro dei contenuti *ad hoc*, riuscendo a mercificare l'intero processo. In questo paradigma, Google certamente è riuscito a dimostrare quanto fosse importante la centralità dell'utente, offrendo dei risultati che potessero rispondere in maniera sempre più mirata alle domande di ricerca. C'è da chiedersi quindi se i dati citati da Dave Yost siano un sinonimo di efficienza del motore di ricerca oppure di ingiustizia sociale.

In ogni caso, è bene ricordare che l'ecosistema creato da Google è un ambiente codificato che non connette semplicemente utenti a siti web, ma determina anche *“il processo mediante il quale essi si connettono.”*<sup>24</sup> Queste regole sono le *policy* e le *guidelines* del motore di ricerca, per cui quando l'azienda decide di punire i trasgressori attraverso le penalizzazioni, anche fino al punto di poter escludere certi attori, persegue il suo interesse privato di preservazione del modello da lei creato e la SEO quindi, per sua definizione, sottostà a tali regole.

## 1.9 Cos'è la SEO?

Ora che si è finalmente delineato il quadro nel quale la SEO è collocata, si possono definire nel dettaglio quali sono gli aspetti più tecnici di tale materia.

Innanzitutto, la SEO si affianca alla *Search Engine Advertising* (SEA) per creare il *Search Engine Marketing* (SEM). SEO e SEA oggi concorrono per migliorare la visibilità di un contenuto su un motore di ricerca, mentre un tempo si sarebbe detto che esse servissero per migliorare il posizionamento di una pagina web su Google.

Questa differenza di termini deriva dal fatto che, nei relativamente pochi anni di vita di queste pratiche, molte cose sono cambiate: oggi sul web non si hanno solo pagine web ma anche contenuti di ogni tipo, come immagini e video, e questi spesso vengono utilizzati per migliorare la visibilità di un *brand* in maniera trasversale e orizzontale in diversi ambienti; esempi significativi sono la *Social Media Optimization*

---

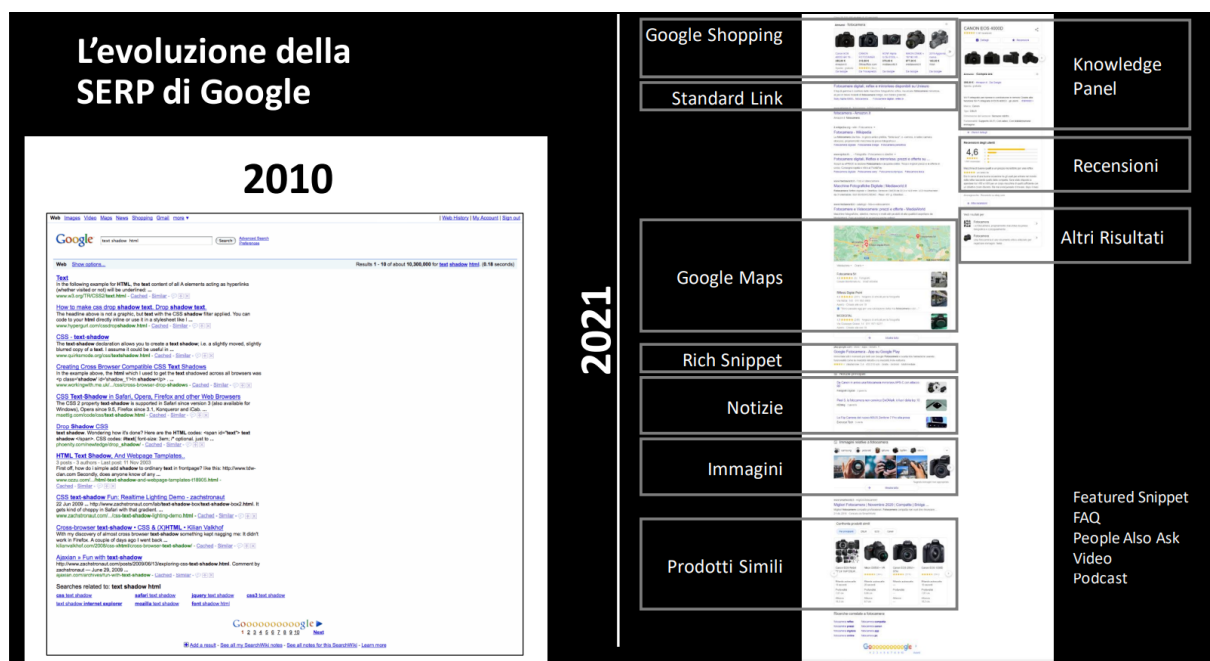
<sup>22</sup> Van Dijck 2019: 60

<sup>23</sup> Ivi: 61

<sup>24</sup> Ibidem

(SMO) e la *App Store Optimization* (ASO) che ben rendono l'idea di come Google, per quanto risulti uno strumento imprescindibile, non sia l'unico adottato per scopi di marketing.

SEO e SEA si riferiscono quindi all'aumento di visibilità nella *Search Engine Results Page* (SERP), ovvero l'insieme dei risultati di ricerca raccolti dal motore di ricerca. Sempre riferendosi a Google, esso ha modificato considerevolmente la SERP dai suoi primi anni di vita fino ad oggi, passando da una pagina contenente soli link ad una ricca collezione di *advertising*, immagini, recensioni, notizie e moltissimi altri contenuti collegati.



**Immagine 6:** Evoluzione della SERP di Google - Fonte: Learnn.

La SEO quindi si riferisce all'ottimizzazione di un sito web per migliorare il posizionamento organico tra i vari elementi della SERP come mostrato nell'immagine 6. La SEA invece, è atta a posizionare un contenuto attraverso l'utilizzo di campagne pubblicitarie a pagamento.

Inoltre, la SEO è suddivisibile in due grandi aree, tra di loro diverse, ma anche con tratti comuni, essendo due facce della stessa medaglia: la SEO Tecnica e la SEO Semantica.

## 1.10 La SEO Tecnica

Per SEO Tecnica si intendono tutte quelle ottimizzazioni maggiormente focalizzate sull'aspetto tecnico. Non tutte le successive pratiche influenzano direttamente il *ranking*, tuttavia molte di esse possono contribuire positivamente alla UX, elemento considerato altrettanto importante da Google. Le pratiche più importanti sono le seguenti:

- **Codici di stato HTTP**

Ogniquale volta un utente (*client*) naviga su Google e clicca su un link, esso esegue una *HTTP Request*, ovvero richiede a Google di contattare i *server* nei quali sono memorizzati i dati desiderati e, una volta instaurata la connessione, restituire una *HTTP Response*, ovvero il risultato di questa interrogazione.

L'*Internet Engineering Task Force* (IETF) ha definito nell'*RFC 2616*<sup>25</sup> i possibili codici di stato di una *HTTP Response*; di seguito si espongono quelli più rilevanti per l'argomento in questione:

- **Informational 1xx**

Come specificato nel documento, un *client* deve essere pronto a ricevere diversi codici 1xx come conferma che la richiesta è stata ricevuta e il server la sta processando.

- **Success 2xx**

Il gruppo dei codici 2xx informa al *client* che la sua richiesta è stata ricevuta con successo, compresa ed accettata. Il codice di risposta generico è 200.

- **Redirect 3xx**

Questa classe di codici indica che il *client* deve eseguire ulteriori azioni per soddisfare la *Request*, reindirizzando il traffico verso altri URL, potendo generare infiniti loop, fino a quando non si avrà in risposta un codice di *Success*. Lo IETF raccomanda di non impiegare più di 5 *redirect*, in quanto certi *client* potrebbero limitare il flusso dati a tale numero.<sup>26</sup>

I reindirizzamenti possono essere implementati per diversi motivi, per esempio una pagina web è in manutenzione, oppure è in corso un A/B test e in tal caso si può utilizzare il codice 302 per reindirizzare temporaneamente l'utente verso un altro sito; oppure perché una pagina web non è più disponibile e quindi per evitare di fermare la

---

<sup>25</sup> <https://www.ietf.org/rfc/rfc2616.txt>

<sup>26</sup> lvi: 60

navigazione utente in una pagina di errore, lo si reindirizza permanentemente con un codice 301 verso una pagina funzionante.

Dal punto di vista utente, i codici 301 e 302 producono lo stesso risultato, ovvero far rimbalzare l'utente in una pagina diversa da quella richiesta; tuttavia, in ottica SEO le differenze sono non poche. Per Google, un *redirect* temporaneo 302 non trasferisce l'*authority* della pagina iniziale verso il nuovo indirizzo, poiché si presume che a breve la navigazione verrà ripristinata e perciò non è necessario penalizzare il punteggio che la pagina aveva ottenuto nel tempo. Al contrario, un *redirect* 301 identifica un reindirizzamento permanente, e quindi l'autorevolezza della pagina iniziale viene trasferita alla nuova pagina di destinazione.

Che si tratti di *redirect* 301 o 302, le pratiche di reindirizzamento non devono essere utilizzate per aggirare l'algoritmo, come si è già detto nei precedenti paragrafi. Un codice 301 trasferisce il valore di *ranking* solamente se il motore di ricerca identifica omogeneità tra quanto era presente nella vecchia pagina e quanto c'è nella nuova. Deviare il traffico di una pagina autorevole verso una pagina differente può determinare l'attribuzione di penalizzazioni, cosiccome un utilizzo inappropriato del codice 302, non ripristinando la vecchia navigazione dopo un certo periodo e causando quindi il ripristino del valore di *authority*.

È infine bene ricordare che esistono diversi modi per reindirizzare il traffico utente verso una pagina di destinazione diversa, anche per motivi legittimi, ad esempio l'accesso ad un sito in seguito ad un login. Gli strumenti che possono permettere questo passaggio sono diversi, tuttavia, in ottica SEO il reindirizzamento non proveniente dal lato *server* o senza codici di stato HTTP potrebbero far insospettare il motore di ricerca, in quanto non viene fornita una motivazione per tale azione, potendo causare la perdita di autorevolezza della pagina web; è quindi buona pratica utilizzare reindirizzamenti solamente lato *server* o con codici HTTP, e quando non è possibile farlo, utilizzare *meta refresh* o indicare il link di destinazione del *redirect* nella *sitemap* del sito web e nella *<head>* della pagina iniziale.<sup>27</sup>

- **Client Error 4xx**

In questo raggruppamento si intendono degli errori causati dal *client* come errori di sintassi oppure richieste che non possono essere soddisfatte, per esempio, per problemi di autorizzazioni.

L'errore più noto è il *404 Page not found*, ovvero quando il *server* non è riuscito a trovare la risorsa richiesta dal *client*. Di per sé, un errore 404 non è un elemento da temere, in quanto può capitare che nel tempo un

---

<sup>27</sup> <https://developers.google.com/search/docs/crawling-indexing/301-redirects>

sito web cambi e con esso le risorse che ospitava, quindi è bene notificare all'utente che la pagina che sta cercando non esiste più. A tal proposito sarebbe buona pratica utilizzare il codice di errore 410 *Gone*, per indicare che la pagina non esiste più e non verrà ripristinata, oppure personalizzare la pagina di errore 404 per uniformare alla UX anche questa tipologia di navigazione.

Per la SEO questi tipi di errori non sono un problema, ma possono diventarlo; infatti, se un sito web è costellato di *error 404*, e molti di questi collegamenti sono presenti nella *sitemap*, oppure una pagina che generava molto traffico all'improvviso smette di funzionare, l'autorevolezza della stessa potrebbe subire penalizzazioni. Per evitare ripercussioni quindi, è bene monitorare il proprio sito web nel momento in cui si modifica, utilizzando per esempio, dei *redirect 301* o *302* quando una pagina viene eliminata, in quanto il motore di ricerca potrebbe impiegare anche alcune settimane per scansionare nuovamente tutto il sito web e perciò la pagina in questione potrebbe essere ancora indicizzata nella SERP.

- **Server Error 5xx**

Infine, gli errori che iniziano con la cifra "5" si rivolgono agli errori causati dal *server*, il quale non è riuscito a soddisfare una richiesta del *client* valida.

L'errore più generico è il 500 e non fornisce particolari dettagli; anche in questo caso è possibile e raccomandabile personalizzare la pagina di errore per evitare frustrazioni dell'utente. In ottica SEO ogni errore nel lungo periodo può divenire un motivo di penalizzazioni, perciò anche per quanto riguarda gli errori 5xx è opportuno tener monitorata l'infrastruttura del sito web.

- **File robots.txt**

Il file robots.txt fa capire ai crawler dei motori di ricerca a quali pagine o file possono o non possono accedere. È formato da regole che possono bloccare o consentire l'accesso di un determinato *crawler* a un percorso di file specificato nel sito web in questione. Per permettere a tutti i motori di ricerca di poter leggere le istruzioni specificate nel file, esistono delle regole da dover rispettare<sup>28</sup>:

- Il nome del file deve essere esattamente "robots.txt";
- Può esistere un solo file robots.txt per sito web;
- Il file robots.txt può risiedere in un sottodominio (<https://website.example.com/robots.txt>) ma non può esimersi dalla

---

<sup>28</sup> <https://developers.google.com/search/docs/crawling-indexing/robots/create-robots-txt>

regola di essere pubblicato nella directory principale dell'host del sito web a cui si applica (<https://www.example.com/robots.txt>) e non può essere presente nelle sottodirectory (<https://example.com/pages/robots.txt>);

- È obbligatorio codificare il file in UTF-8 in quanto certi motori di ricerca come Google potrebbero ignorare eventuali caratteri non presenti in tale codice;
- Il file può essere composto da diversi gruppi di direttive altrimenti, senza nessuna indicazione, i motori di ricerca hanno il permesso di scansionare l'intero sito web senza limitazioni. Le principali regole sono le seguenti :

- **User-agent:**

Questa regola obbligatoria deve essere ripetuta per ogni *crawler* di ogni motore di ricerca al quale si vogliono applicare determinate istruzioni; se si utilizza il valore “ \* ” le regole vengono estese a qualsiasi *crawler*;

- **Disallow:**

Anche questa regola va ripetuta almeno una volta e per ogni percorso di file del quale si vuole bloccare la scansione; se si utilizza come valore finale “ / ” ci si riferisce all'intera directory;

- **Allow:**

Questo tipo di regola serve per eseguire un *override* di una o più regole *disallow* specificate precedentemente;

- **Sitemap:**

Regola facoltativa che può indicare al motore di ricerca l'esistenza di uno o più file di *sitemap*.

- **Crawl-delay:**

Direttiva facoltativa utilizzata per far attendere il *crawler* al fine di evitare sovraccarichi del server;

- **Noindex e Nofollow:**

Coppia di regole facoltative che indicano al motore di ricerca rispettivamente di de-indicizzare una pagina e di non seguirne i link durante la scansione.

Riguardo a queste ultime tre direttive, Google nel 2019 <sup>29</sup> ha annunciato che non le supporterà più all'interno dei file robots.txt, raccomandando inoltre, per quanto riguarda le regole *noindex* e *nofollow*, l'utilizzo di *meta tag robots* <sup>30</sup>. Questi tag applicano le medesime direttive, ma sono inserite nelle *<head>* delle singole pagine web anziché nel file robots.txt.

Per la SEO, la presenza di un file robots.txt dettagliato e ben strutturato è utile al fine di ottimizzare la scansione del sito web in quanto potrebbero essere presenti pagine web utili all'utente ma non meritevoli di essere indicizzate poiché possibilmente penalizzabili (Google per esempio etichetta negativamente le pagine con meno di 300 caratteri) oppure poiché sono presenti duplicati di pagine (come per esempio diverse pagine per le diverse versioni di un unico prodotto di un e-commerce) e quindi si indica al *crawler* di non sprecare *crawl budget* per scansionare pagine cloni.

- **Pagine duplicate**

Proprio in merito al problema appena citato delle pagine duplicate, due attributi di *tag* molto importanti sono *<rel=canonical>* e *<hreflang>*.

Il primo viene utilizzato per indicare al motore di ricerca, in concerto con le direttive del robots.txt, quale sia la pagina web canonica tra le possibili copie della stessa, permettendo anche in questo caso il risparmio del *crawl budget* analizzandone una sola.

Il secondo attributo invece si riferisce alla possibili copie in diverse lingue di un sito web. In questo caso *<hreflang>* non si applica per indicare quale sia la versione che il *crawler* deve analizzare, ma permette al motore di ricerca di localizzare geograficamente la scansione del sito web in funzione della SERP che dovrà mostrare all'utente; per tale motivo le pagine in diverse lingue non sono considerate come dei duplicati. Tuttavia, una pratica di ottimizzazione di comune utilizzo per i siti web di piccole dimensioni è quello di produrre un'unica versione del sito e di utilizzare *<hreflang="x">* il quale indica al motore di ricerca di tradurre automaticamente i contenuti secondo la localizzazione della lingua scelta dall'utente.

- **File sitemap.xml**

Il file di *sitemap*, generalmente esportati con estensione XML, sono dei file contenenti un elenco dei principali URL di un sito web e che possono essere notificati ai motori di ricerca, indicando loro quali link scansionare.

È bene specificare che i *crawler* dei motori non seguono la priorità dei link forniti dal *webmaster*; i file *sitemap* infatti indicano quali sono le pagine di maggior rilevanza secondo la persona che ha progettato il sito web, tuttavia i motori non ne garantiscono l'indicizzazione. A differenza dei file robots.txt nei

---

<sup>29</sup> <https://developers.google.com/search/blog/2019/07/a-note-on-unsupported-rules-in-robotstxt>

<sup>30</sup> [https://developers.google.com/search/docs/advanced/robots/robots\\_meta\\_tag](https://developers.google.com/search/docs/advanced/robots/robots_meta_tag)

quali vengono indicati, i file *sitemap* possono essere molteplici in quanto alcuni motori come Google accettano un limite di peso di 50 MB e 50.000 link per file, è quindi possibile e consigliabile suddividere i link per raggruppamenti semantici, anche al fine di ottimizzare l'individuazione di eventuali errori.

Congiuntamente con le pratiche sopracitate, i file *sitemap* ai fini della SEO sono essenziali per consigliare ai motori di ricerca quali collegamenti navigare, al fine di ottenere buoni valori per il posizionamento nella SERP.

- **Livello di connessione HTTP**

Seppur la qualità di connessione non influenzi direttamente il *ranking* su Google, essa può beneficiare sotto diversi aspetti.

Innanzitutto, una connessione con protocollo HTTP è un insieme di regole che gestisce la comunicazione tra *client* e *server*. Grazie all'unione del protocollo HTTP e del protocollo di sicurezza *Secure Sockets Layer* (SSL) o del suo successore *Transport Layer Security* (TLS), nasce l'HTTPS, un protocollo di connessione cifrato e maggiormente sicuro.

La versione HTTP più longeva fu la 1.1, resa standard con la RFC 2616 nel 1999 e adottata fino al 2015; essa tuttavia si basava su un massimo di sei connessioni parallele per poter inviare i dati, poiché esse possono trasportare una singola richiesta alla volta in maniera unidirezionale. Questo potrebbe portare all'insorgere di una coda di richieste in attesa di essere processate fin quando una delle connessioni non si libera, determinando una possibile latenza nel caricamento della pagina web.

La IETF nel 2015 pubblica la RFC 7540 <sup>31</sup> presentando HTTP/2, una nuova versione del protocollo. Questo nuovo tipo di connessione incorpora, tra le sue varie parti, HTTPS e la tecnologia *multiplex*, la quale permette di gestire connessioni parallele che inviano e ricevono richieste senza aspettare la risoluzione della precedente, diminuendo notevolmente i tempi di caricamento della pagina.

Tempi di caricamento minori comportano un incremento del *crawl rate*, migliore UX, accessibilità e valore di *Page Speed*, del quale si parlerà di seguito.

- **Page Speed**

In seguito agli ultimi aggiornamenti di Google, il motore ha dichiarato che il tempo di caricamento della pagina è diventato ufficialmente uno dei criteri di *ranking*.

Un'ottimizzazione di questo parametro non beneficia solamente la SEO, ma anche le performance del sito web <sup>32</sup>. Una ricerca condotta da 55 e Deloitte nel 2019 <sup>33</sup> ha evidenziato come la riduzione di soli 0,1 secondi del

---

<sup>31</sup> <https://www.rfc-editor.org/rfc/rfc7540>

<sup>32</sup> <https://www.thinkwithgoogle.com/intl/it-it/strategie/app-e-mobile/migliorare-velocita-sito-mobile/>

<sup>33</sup> [https://www2.deloitte.com/content/dam/Deloitte/ie/Documents/Consulting/Milliseconds\\_Make\\_Millions\\_report.pdf](https://www2.deloitte.com/content/dam/Deloitte/ie/Documents/Consulting/Milliseconds_Make_Millions_report.pdf)



caricamento di una pagina web per smartphone ha determinato un incremento dell'8,4% di conversioni per delle rivendite al dettaglio e del 10,1% per dei siti di viaggi.

Altri dati <sup>34</sup> mostrano che gli utenti si aspettano un'attesa media di caricamento di una pagina web di 2 secondi e che la probabilità di *bounce* aumenti del 32%, per i siti che caricano entro 3 secondi, del 90% per i siti che ne impiegano fino a 5 e del 123% per le pagine che si caricano entro un massimo di 10 secondi.

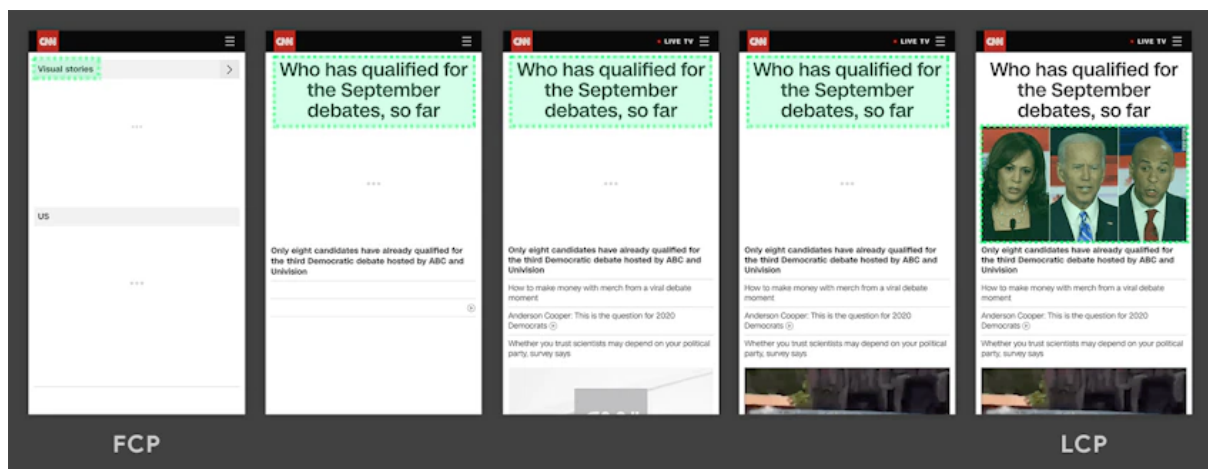
Per migliorare la velocità di caricamento si possono impiegare moltissime ottimizzazioni, le più significative sono l'utilizzo della *cache* del browser, la compressione di elementi multimediali, velocizzare le risposte del server, eliminare le risorse *render-blocking* (ovvero risorse critiche per il caricamento della pagina, le quali tuttavia interrompono lo stream di *rendering* dei contenuti finchè non vengono caricate) oppure impiegare la tecnica del *lazy loading* consistente nel caricare i contenuti man mano che l'utente scorre il sito e inserendo gli elementi più importanti in cima alla pagina web.

- **Core Web Vitals**

Come già esposto, con l'aggiornamento del 2021 i Core Web Vitals sono un insieme di fattori influenzanti il *ranking* delle pagine web. Diversi possono essere gli approcci di ottimizzazione, i principali sono i seguenti:

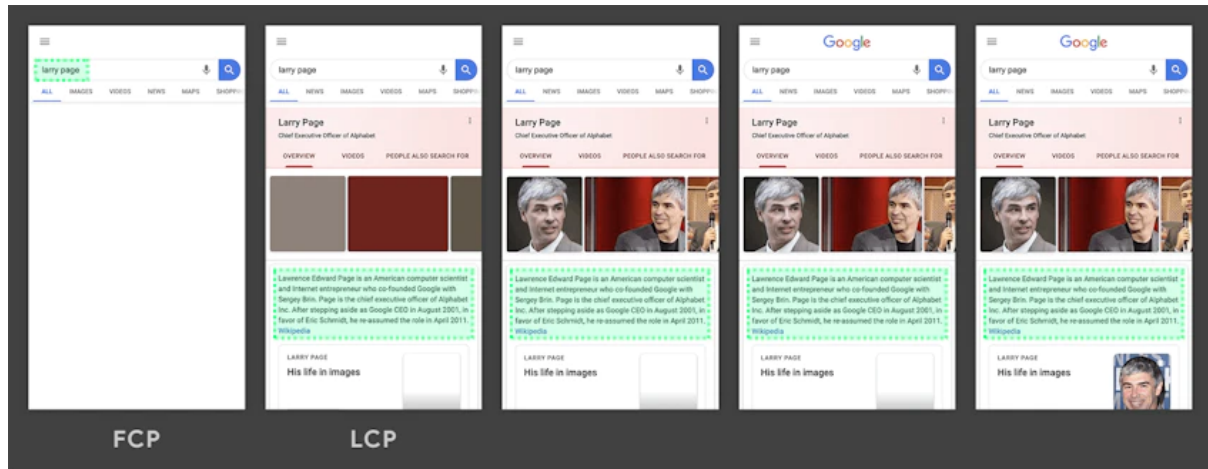
- **LCP**

Per migliorare il parametro del *Largest Contentful Paint*, il *webmaster* deve ottimizzarne il caricamento, posizionandolo tra i primi elementi che la pagina elabora, per permetterne un *rendering* senza inutili latenze, in quanto un buon punteggio di LCP si attesta su massimo 2,5 secondi.



<sup>34</sup><https://www.thinkwithgoogle.com/intl/en-154/marketing-strategies/app-and-mobile/need-mobile-speed-how-mobile-latency-impacts-publisher-revenue/>

**Immagine 7:** Analisi di caricamento di una pagina web della CNN - Fonte: Web.dev (<https://web.dev/lcp/>).



**Immagine 8:** Analisi di caricamento di una pagina di risultati Google - Fonte: Web.dev (<https://web.dev/lcp/>).

L'immagine 7 mostra come la pagina web non ottimizzi il LCP, rilevato dal motore di ricerca inizialmente nei titoli del menù e dell'articolo e, successivamente, nell'immagine che viene renderizzata solamente nell'ultimo *frame*.

Al contrario, nell'immagine 8 si può notare come la stessa Google ottimizzi il LCP già nella sua SERP, sincerandosi che questo venga caricato fin dai primi *frame*, coincidendo quasi con il *First Contentful Paint* (FCP).

- **CLS**

Sempre nell'immagine 8 si può notare come Google abbia posizionato dei *container* nel codice della pagina web, ottimizzando il *Content Layout Shift*. Questa ottimizzazione consente alla pagina di non subire nessuno *shift* del contenuto in quanto le immagini che si caricano dal terzo *frame* si posizionano negli spazi predefiniti, adattando la loro grandezza e senza interferire con altri elementi.

- **FID**

Una buona risposta del browser si attesta sui 100 millisecondi perciò l'ottimizzazione del *First Input Delay* deve mirare ad ottimizzare gli elementi di caricamento come per la *Page speed*.

In generale, delle buone pratiche per ottimizzare FID e LCP riguardano dei migliori tempi di risposta dei server, utilizzando DNS veloci, protocolli di rete moderni e predisponendo un sistema di *caching* lato server. Per quanto

riguarda invece il caricamento degli elementi delle pagine web come file di *script*, fogli di stile, *font*, o immagini e video, una buona strategia è quella di archiviare tali risorse nella memoria locale, minimizzandole quando possibile e precaricare elementi critici e ritardare il caricamento degli elementi secondari solamente quando l'utente ne avrà bisogno.

## 1.11 La SEO Semantica

La SEO Semantica è l'insieme di quelle pratiche maggiormente incentrate sull'ottimizzazione del significato semantico dei contenuti di un sito web, utili per meglio indicare al motore di ricerca e all'utente quali sono gli argomenti trattati. Come per la SEO Tecnica, non tutte le seguenti azioni rientrano direttamente nei fattori di *ranking*; nonostante ciò, esse forniscono molte migliorie per la *user experience*.

Inoltre, è necessario ribadire un concetto molto importante legato alla SEO Semantica soprattutto. Con l'aggiornamento *Hummingbird*, come già detto nel paragrafo 1.6, Google ha introdotto l'utilizzo di intelligenze artificiali che cercassero di comprendere il significato delle intenzioni dell'utente, cambiando l'approccio che l'azienda applicava nel fornire risultati di ricerca. Forse ispirati dalle parole di Tim Berners-Lee <sup>35</sup>, Google ha iniziato a creare un proprio database composto non solo da *link* (i quali rimangono tutt'oggi la base di Internet), ma anche di *linked-data*, ovvero informazioni tra di loro collegate che formano una rete di conoscenza semantica. È proprio grazie a questi dati che i motori di ricerca come Google riescono a prescindere dalle specifiche parole utilizzate dall'utente in fase di ricerca ma colgono al contempo il significato di quello che l'utente sta cercando. Se quindi, per esempio, si provasse a cercare su Google “capo di stato repubblica italiana” e “presidente della repubblica italiana”, ambedue le ricerche offrirebbero come primo risultato link su Sergio Mattarella.

La SEO Semantica si occupa di questo: fornire un significato al contenuto e renderlo facilmente leggibile ai motori di ricerca.

- **Structured data**

Gli *Structured data* sono delle linee di codice in formato JSON che forniscono ulteriori informazioni sui contenuti del sito web. I dati strutturati non sono elemento diretto di *ranking*, tuttavia offrono un'esperienza personalizzata all'utente, il quale scopre che tipo di contenuto lo attende prima ancora di cliccare il link nella SERP, potendo determinare un vantaggio competitivo verso i concorrenti.

---

<sup>35</sup> Tim Berners-Lee: *The next Web of open, linked data* (2009), [https://www.youtube.com/watch?v=OM6XlICm\\_qo](https://www.youtube.com/watch?v=OM6XlICm_qo)

Alcuni esempi di tipi di *Structured data* sono *article*, *book*, *FAQ*, *logo*, *video*, *product*, *app software* <sup>36</sup>.

- **Keyword**

Una *keyword* (o parola chiave) è qualsiasi termine immesso su un motore di ricerca come *query* di ricerca che genera una collezione di risultati in cui sono elencate le pagine web che si posizionano per quella stessa parola, ovvero che la utilizzano all'interno dei loro contenuti e dei metadati.

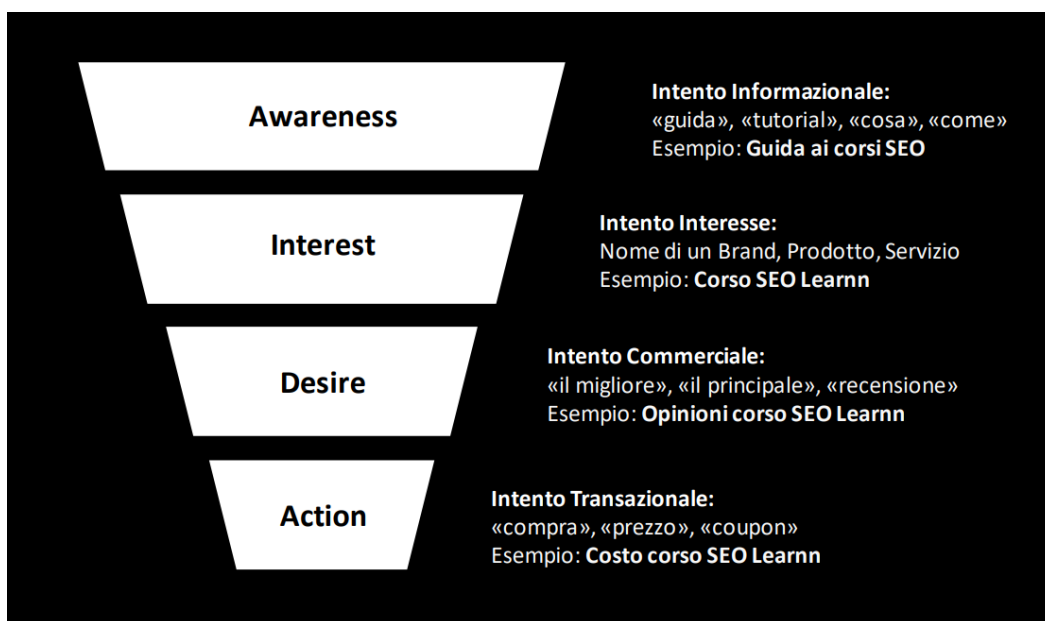
Le *keyword* hanno diverse caratteristiche:

- **Search Volume**

Il volume di ricerca indica quante volte una determinata *keyword* è ricercata dagli utenti.

- **Intento di ricerca**

L'intento di ricerca individua lo scopo di ricerca per l'utente, se generico o specifico.



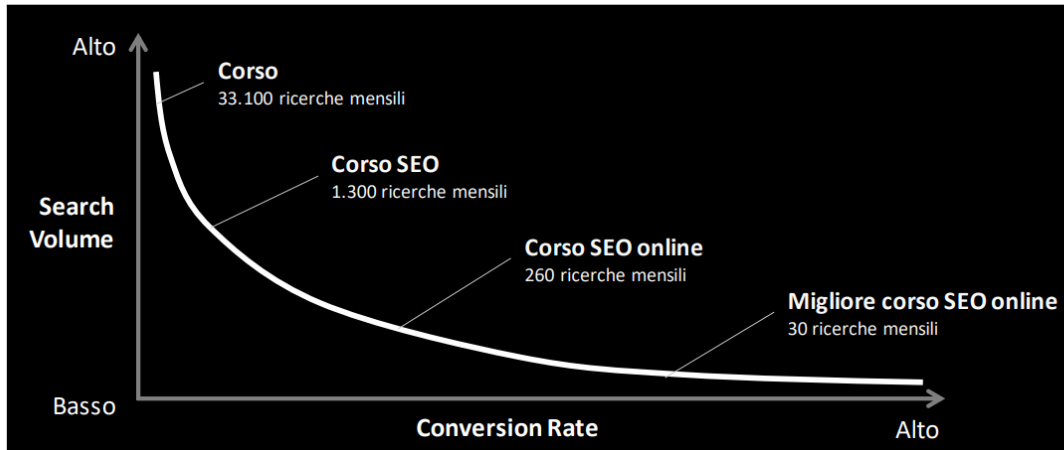
**Immagine 9:** Intento di ricerca nel modello di *funnel* AIDA - Fonte: Learnn.

Inoltre, l'intento di ricerca può essere anche utilizzato come indicatore del percorso di un utente in un *funnel* di un *customer journey* come indicato nell'immagine 9.

<sup>36</sup> <https://developers.google.com/search/docs/advanced/structured-data/intro-structured-data>

- **Numero di parole**

Metrica che si riferisce a quante parole sono presenti nella ricerca. Le parole di ricerca possono differenziarsi in *short-tail keyword* e *long-tail keyword*.



**Immagine 10:** Differenza tra *short-tail keyword* e *long-tail keyword* - Fonte: Learnn.

Come mostra l'immagine 10, le *short-tail keyword* sono parole di ricerca brevi, composte da 1-2 vocaboli, le quali generano un alto volume di ricerca con un generico intento di ricerca, mentre le *long-tail keyword* sono formate 4-7 parole e si posizionano per bassi *search volume* ma con intenti di ricerca più specifici.

- **Concorrenza e Costo per Clic (CPC)**

Queste ultime due caratteristiche si rivolgono prevalentemente alle strategie SEA; la concorrenza indica quanta offerta di siti web con una data *keyword* sia presente online, determinando di conseguenza un maggiore CPC per quei termini di ricerca maggiormente gettonati, poiché in molti saranno disposti a pagare per posizionarsi ai primi posti.

- **URL**

Tra i primi elementi che vengono analizzati dai *crawler* si posiziona l'URL del sito web in questione. Esso, esattamente come i contenuti della pagina, deve utilizzare un linguaggio comprensibile sia per l'utente sia per il motore di ricerca perciò, ad esempio, un URL <https://www.example.com/corso-seo/ottimizzare-url/> viene valutato positivamente a differenza di un URL <https://www.example.com/cartella1/in?var=url>.

Le *best practices* in tale merito riguardano l'utilizzo di *keyword* per le quali si vuole posizionare il sito, utilizzare il simbolo “ - ” per separare le parole ed il loro significato e il simbolo “ \_ ” per unirle, scegliere di far terminare o meno

tutti gli URL con il simbolo “ / ” ed evitare di impiegare URL troppo lunghi e poco informativi.

- **Meta tag Title e Description**

Successivamente all'analisi dell'URL, avviene quella del *Title* e della *Description* nel tag HTML `<head>` della pagina web, i quali offrono altre informazioni sul contenuto della pagina. Anche in questo esistono delle linee guida come ad esempio utilizzare *keyword* presenti nei contenuti della pagina senza tuttavia stravolgerne il significato ed impiegare al massimo 60 caratteri per il *Title* e 150 (120 per il *mobile*) per la *Description*.

- **Heading Tags**

Sono dei tag HTML utili per differenziare titoli (`<H1>`) e sottotitoli (`<H2>` - `<H6>`) all'interno della pagina. Google e gli altri motori di ricerca non definiscono dei comportamenti obbligatori da impiegare, quanto più delle *best practices* per aiutare i *crawler* a identificare i contenuti più rilevanti. È buona norma quindi utilizzare un solo tag `<H1>` nella pagina, strutturare nella maniera più semplice il testo ed include anche nei titoli le *keyword*.

- **Link**

I link dovrebbero essere ottimizzati anche a livello semantico. Gli *anchor text* utilizzati per i collegamenti dovrebbero essere esplicativi della destinazione a cui puntano, aiutando utenti e motori nella navigazione. Testi eccessivamente lunghi o generici (ad esempio “clicca qui”) dovrebbero quindi essere evitati.

- **Immagini e video**

Le immagini e i video potrebbero essere tra gli elementi più pesanti da caricare per una pagina web, è quindi necessario cercare di ridurre le dimensioni il più possibile, senza rinunciare alla qualità, pre caricandole con dei *preload* e ritardando il caricamento di contenuti secondari con *lazy loading*. Per quanto riguarda le immagini, si possono inoltre utilizzare formati vettoriali come SVG o WEBP.

Inoltre, i nomi delle immagini e video ed i corrispettivi tag `<alt>` dovrebbero essere esplicativi, anche in un'ottica di accessibilità del sito web.

# CAPITOLO 2

## 2.1 Domanda di ricerca

Definite quindi le caratteristiche della materia, è possibile realizzare un software che collezioni tali informazioni dai siti web per poi analizzarle?

## 2.2 Analisi del mercato

Sul mercato sono già presenti diversi *tool* che offrono agli utenti analisi dei siti web in chiave SEO. Tra i nomi più referenziati e autorevoli si trovano Semrush <sup>37</sup>, Ahrefs <sup>38</sup>, Google Search Console <sup>39</sup>, Moz <sup>40</sup>, Ubersuggest <sup>41</sup>, Screamingfrog <sup>42</sup> e tantissimi altri. Analizzando questi servizi è emerso un funzionamento comune, classificabile in *web application* o software installabili che permettono al *webmaster* di collegare il proprio sito web al programma per analizzare gli aspetti tecnici e semantici.

Tra le principali caratteristiche tecniche analizzate dai *tool* si trovano la velocità del sito, i link in entrata ed in uscita, le *keywords*, *http status* delle pagine del sito, estensione dei file, dati sul traffico e posizionamento nella SERP; mentre per gli aspetti semantici vengono raccolte informazioni sempre sulle *keywords*, presenza di elementi che contengono meta-informazioni e gerarchia delle pagine del sito.

Altra caratteristica condivisa è la presenza di un *paywall* per questi servizi oppure di un modello *freemium* che ne limita il funzionamento nella versione gratis, determinando quindi la necessità di doversi abbonare per poter usufruire pienamente del *tool*.

Molti dei software alternativi e gratuiti ai principali *player* del mercato risultano offrire servizi limitati a raccogliere specifici dati senza mirare a realizzare un *audit SEO* completo, portando quindi un *webmaster* che non dispone delle risorse per poter utilizzare dei software proprietari, a doversi munire di una moltitudine di strumenti non interagibili tra di loro.

---

<sup>37</sup> <https://www.semrush.com/>

<sup>38</sup> <https://ahrefs.com/>

<sup>39</sup> <https://search.google.com/search-console/about>

<sup>40</sup> <https://moz.com/>

<sup>41</sup> <https://app.neilpatel.com/en/dashboard>

<sup>42</sup> <https://www.screamingfrog.co.uk/>

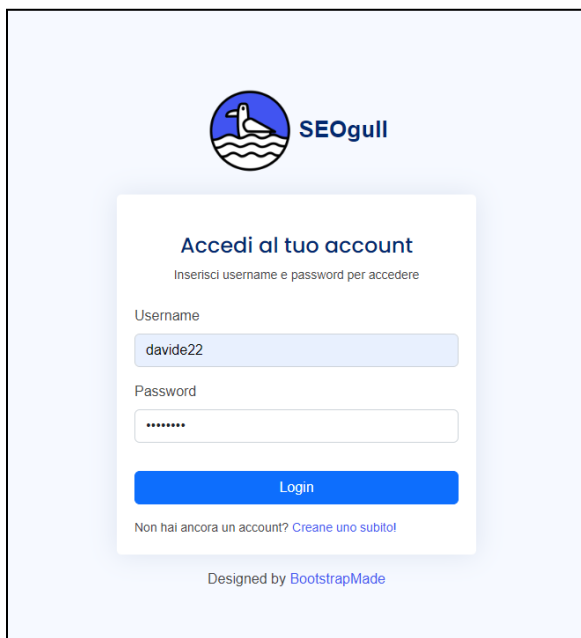
## 2.3 Realizzazione di SEOgull

Analizzate le caratteristiche del mercato e dei suoi principali attori, si è quindi deciso di realizzare una *web application* denominata SEOgull. Di seguito ne verranno spiegate le sue caratteristiche di progettazione ed il suo funzionamento. Il link per poter scaricare il codice è il seguente:

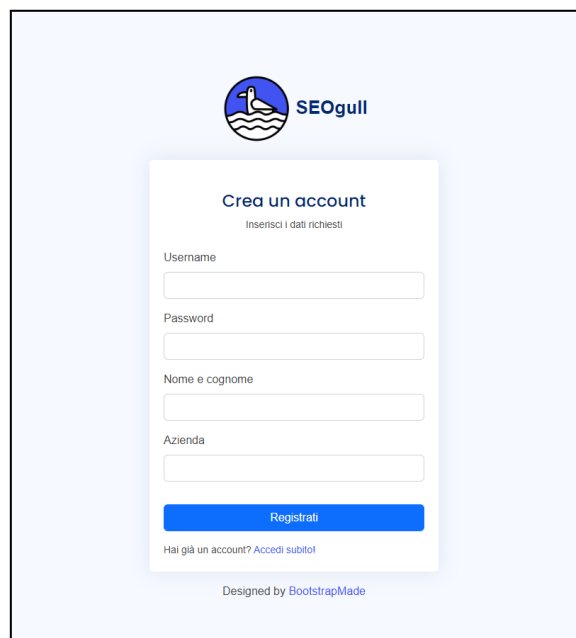
[https://drive.google.com/drive/folders/1YM8q7hwm4-w25GPHXqeC2IMS\\_j\\_Jlj9Y?usp=sharing](https://drive.google.com/drive/folders/1YM8q7hwm4-w25GPHXqeC2IMS_j_Jlj9Y?usp=sharing)

### 2.3.1 Front-end ed interazione con la GUI

Per la realizzazione del *front-end* sono stati utilizzati il *framework* Bootstrap, i linguaggi HTML, CSS e Javascript. In merito a quest'ultimo, è stata utilizzata la libreria jQuery per l'interazione e manipolazione dei dati del *Document Object Model* (DOM) e le librerie ApexCharts.js ed Echarts per la realizzazione dei grafici. La GUI si presenta quindi come mostrato nelle seguenti immagini.



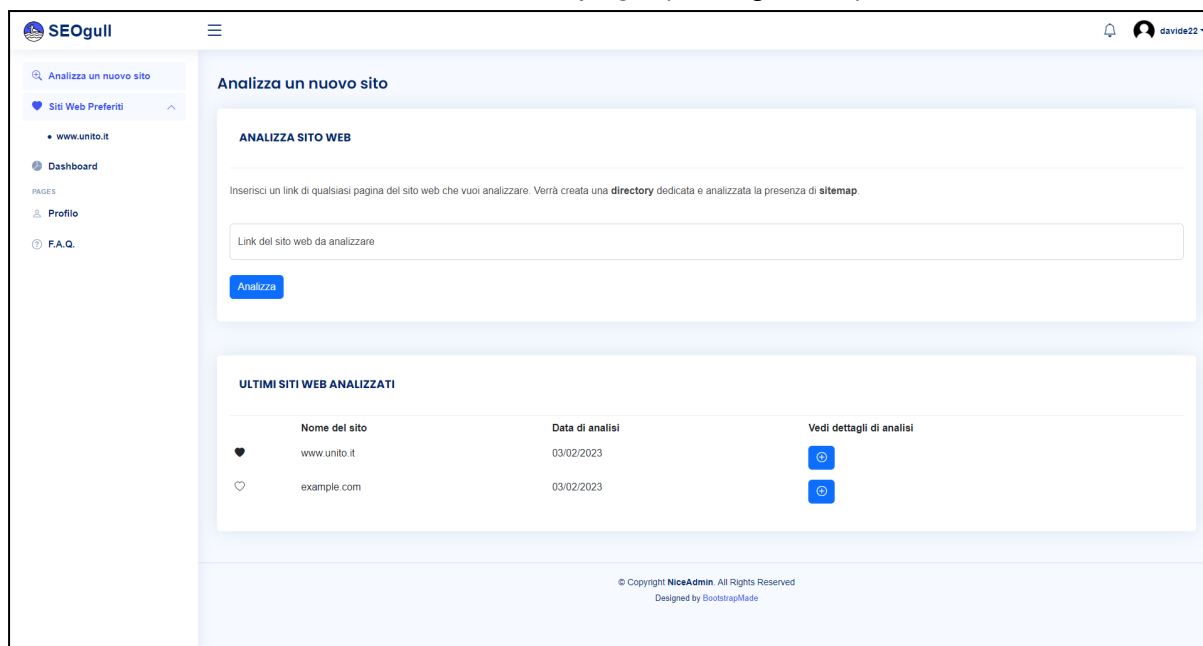
**Immagine 11:** Pagina di login (*login.php*).



**Immagine 12:** Pagina di registrazione (*registrazione.php*).



Attraverso questi due *form*, l'utente può accedere al sito (immagine 11) oppure creare un nuovo profilo e registrarsi (immagine 12); una volta eseguito l'accesso, l'utente verrà reindirizzato verso la *homepage* (immagine 13).



**Immagine 13:** Pagina di *homepage* (*homepage.php*).

Da questa pagina l'utente può compiere le due azioni principali dell'applicazione: analizzare un nuovo sito web oppure vedere i dati di un sito web già analizzato in precedenza.

Nel primo caso, inserendo un URL nel *form* della sezione “Analizza sito web” e cliccando sul bottone “Analizza”, partirà una coppia di chiamate AJAX (*Asynchronous JavaScript and XML*) nidificate, le quali innanzitutto creeranno una cartella dedicata al nuovo sito web nella *directory* dell'utente e, successivamente, cercheranno la presenza di una *sitemap*. L'URL inserito dall'utente potrà riferirsi a qualsiasi pagina del sito web, in quanto da esso verrà estratto il nome del dominio ed in base a questo verrà creata la nuova cartella dedicata; tuttavia, qualora esistesse già una cartella per il sito web nella *directory* dell'utente, apparirà un *pop-up* di richiesta conferma per la sovrascrittura della cartella esistente.

Per quanto concerne la sezione “Ultimi siti web analizzati”, l'utente potrà visualizzare gli ultimi 5 *website* analizzati in ordine decrescente di data. Attraverso le icone a forma di cuore, è possibile aggiungere o rimuovere il relativo sito web alla lista dei preferiti, sempre visibile ed accessibile nella barra laterale delle funzionalità. Cliccando sui bottoni “Vedi dettagli di analisi” oppure sui link dei preferiti, l'utente verrà reindirizzato alla pagina di *dashboard* già focalizzata sul sito web selezionato dall'utente (immagine 14.1) mostrandone i relativi dati, mentre se esso o essa accedono a questa pagina attraverso la barra laterale delle funzionalità, verrà inizialmente visualizzata la sola lista di tutte le analisi utente, dalla quale si potrà scegliere un sito web da visualizzare, rivelando la sezione dei grafici.

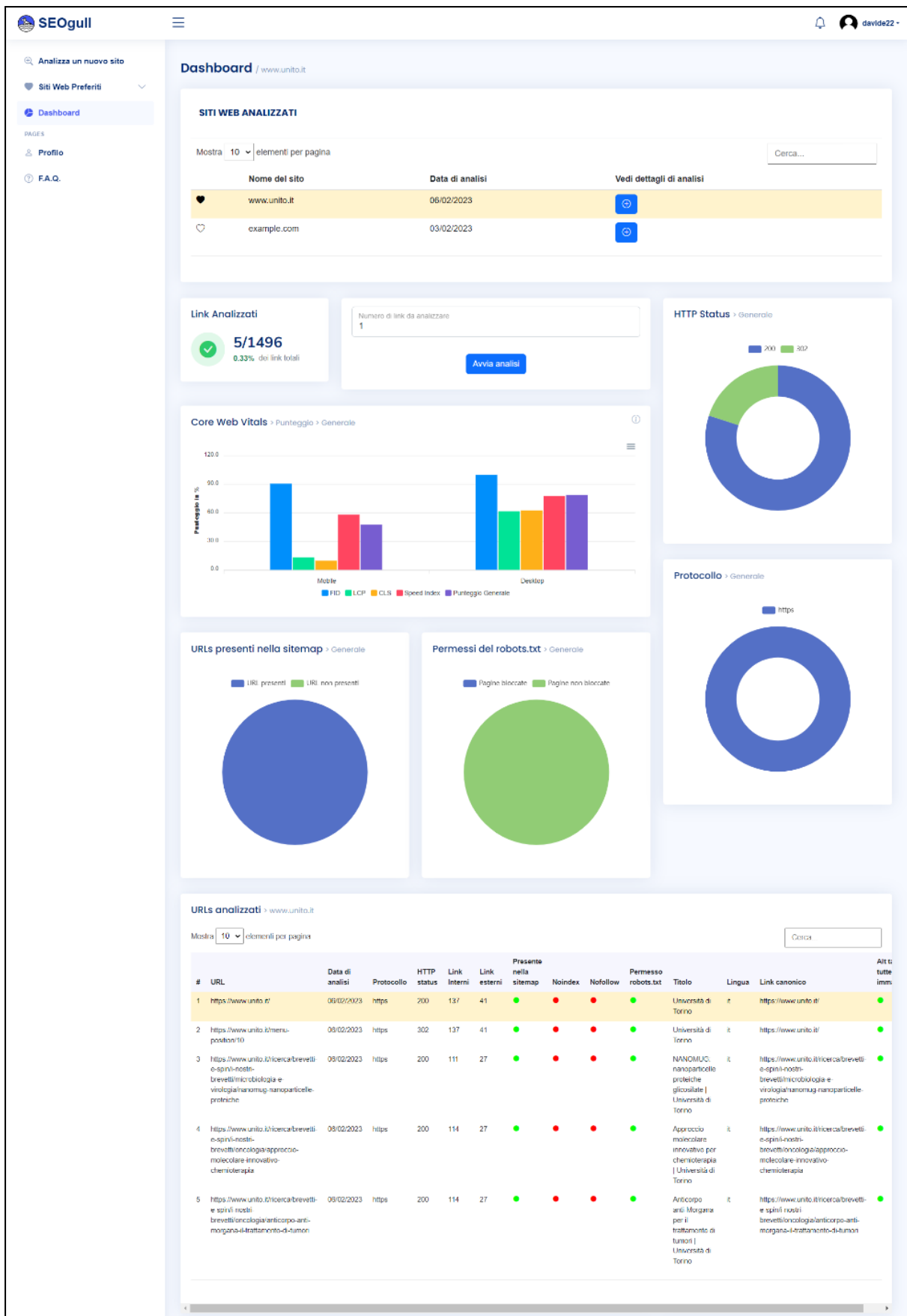
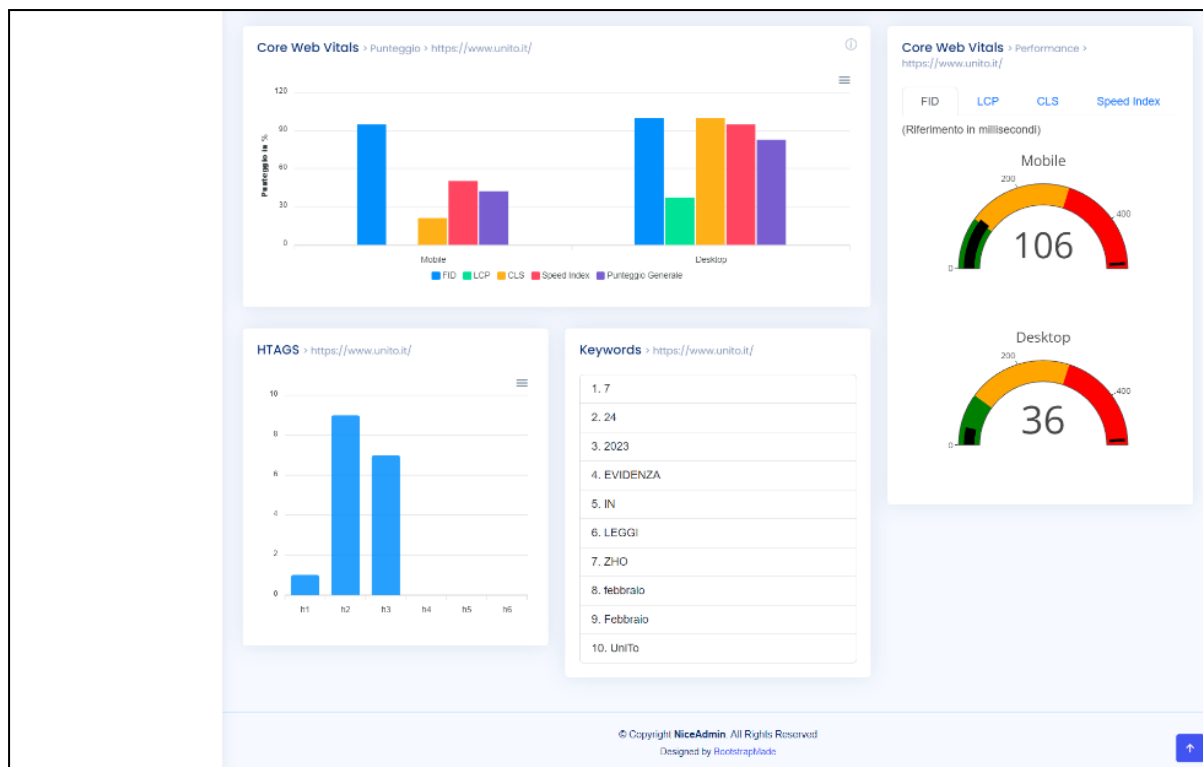


Immagine 14.1: Pagina di *dashboard* con vista espansa (*dashboard.php*).



**Immagine 14.2:** Pagina di *dashboard* con vista espansa (*dashboard.php*).

Nella pagina di *dashboard* sono quindi elaborate e mostrate le informazioni raccolte dal programma per ogni pagina web del sito. Inizialmente non esisterà nessun dato in quanto la creazione della cartella e la rilevazione di *sitemap* sono solamente delle azioni preparatorie al processo di analisi del *crawler*, del quale si parlerà nel dettaglio nel prossimo sottocapitolo.

L'interfaccia di questa pagina, come già accennato, è composta da diverse sezioni:

- **Tabella dei siti web analizzati:** contiene una lista ordinabile per diversi parametri di tutti le analisi dell'utente; cliccando su una riga essa espanderà o aggiornerà la vista della pagina, nel caso fosse già stata espansa, mostrando la prossima sezione;
- **Panoramica generale sul sito web:** in questa sezione è possibile visualizzare quante pagine sono state rilevate per il sito web e quante di esse sono già state analizzate; inoltre è presente un campo di *input* per impostare quanti link si desiderano analizzare con la successiva iterazione di calcolo del *crawler*, da un minimo di un solo link ad un massimo di dieci. Questa soglia imposta è dovuta ad un motivo di praticità, in quanto ogni link per essere analizzato impiega tra i 30 ed i 60 secondi ed è quindi risultato necessario limitare il tempo di attesa dell'utente. Una volta avviata l'analisi, l'utente può comunque continuare a visitare la pagina, interagendo con i grafici di questa sezione, i quali mostrano le diverse

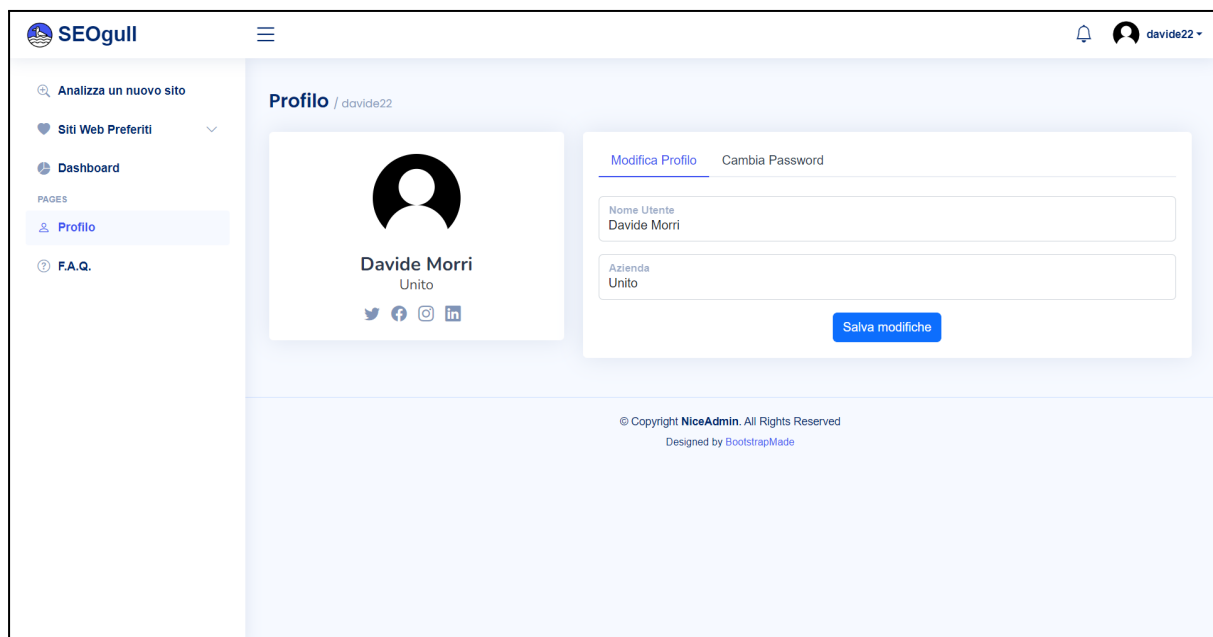
risposte HTTP delle pagine già analizzate, i protocolli utilizzati, la presenza o meno dei link delle pagine nella *sitemap* del sito, l'eventuale blocco di certe pagine dovute alle direttive del *robots.txt*, un grafico a barre con i punteggi medi del sito dei *Core Web Vitals* divisi per dispositivo ed infine una tabella con le principali informazioni delle pagine analizzate.

Cliccando su uno di queste righe, verrà visualizzata o aggiornata l'ultima sezione della pagina;

- **Performance della pagina:** nell'ultima sezione sono presenti dei grafici che mostrano i dati di performance della singola pagina web del sito analizzata, suddivisi in un grafico a barre che, in analogia con il sopracitato, mostra i punteggi dei *Core Web Vitals* divisi per dispositivo, mentre un grafico a cruscotto ne mostra le prestazioni in unità assolute. Sono altresì presenti un grafico che mostra la frequenza di *htags* ed una lista delle *keywords* presenti nella pagina.

La strutturazione della pagina di *dashboard* rappresenta in linea definitiva una forma ad imbuto della visualizzazione dei dati, passando dalla prima sezione generica per la cronologia di analisi dell'utente ad un'ultima specifica, focalizzata sulle prestazioni di una data pagina web.

Infine, per quanto riguarda il *front-end*, la sezione “*Pages*” della barra laterale delle funzionalità mostra due ultime pagine. Una di queste - la sezione delle F.A.Q. - ospiterà in futuro un *tutorial* sul funzionamento del programma ed eventuali risposte alle domande più comuni; mentre il link “Profilo” rimanda alla pagina di impostazioni del profilo utente (immagine 15), dal quale esso o essa possono modificare i propri dati personali ed aggiornare la *password*.



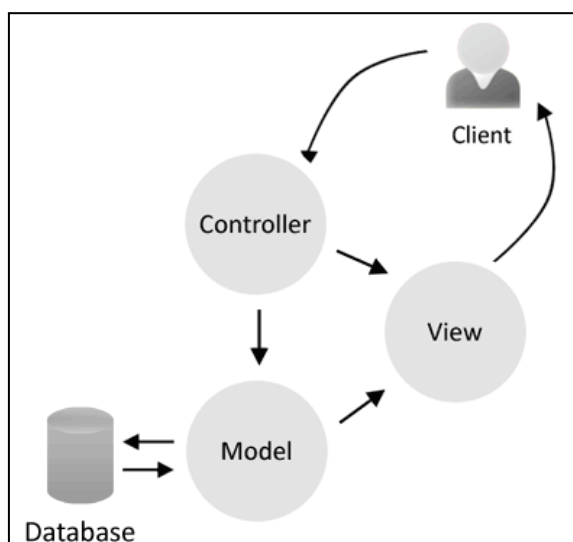
**Immagine 15:** Pagina di impostazioni del profilo utente (*profile.php*).

### 2.3.2 Back-end

Per il *back-end* dell'applicazione è stato impiegato il linguaggio di programmazione PHP, attraverso il *framework object-oriented* CodeIgniter versione 3.1.13<sup>43</sup>; la scelta nell'utilizzo di una versione meno recente è imputabile alla ricerca di una maggiore stabilità dovuta ad anni di aggiornamenti e contributo degli utenti.

Il *framework* scelto è basato su un *pattern* architetturale *Model-View-Controller* (MVC) il quale suddivide il sistema in tre parti distinte:

- **Model:** si occupa di definire i metodi dell'applicazione;
- **View:** si occupa di gestire l'interfaccia e l'interazione con l'utente;
- **Controller:** si occupa di ricevere i comandi dall'utente attraverso la View, manipolando i metodi del Model e aggiornando di conseguenza la View.



**Immagine 16:** Pattern MVC - Fonte: [html.it](http://html.it)<sup>44</sup>.

L'utilizzo del pattern MVC consente di separare la logica di business del sistema, imputabile al Model ed al Controller, dall'interfaccia, attribuita invece alla View.

Questa divisione del sistema incontra quindi i principi fondamentali della programmazione ad oggetti teorizzati da Grady Booch<sup>45</sup>:

<sup>43</sup> <https://codeigniter.com/userguide3/>

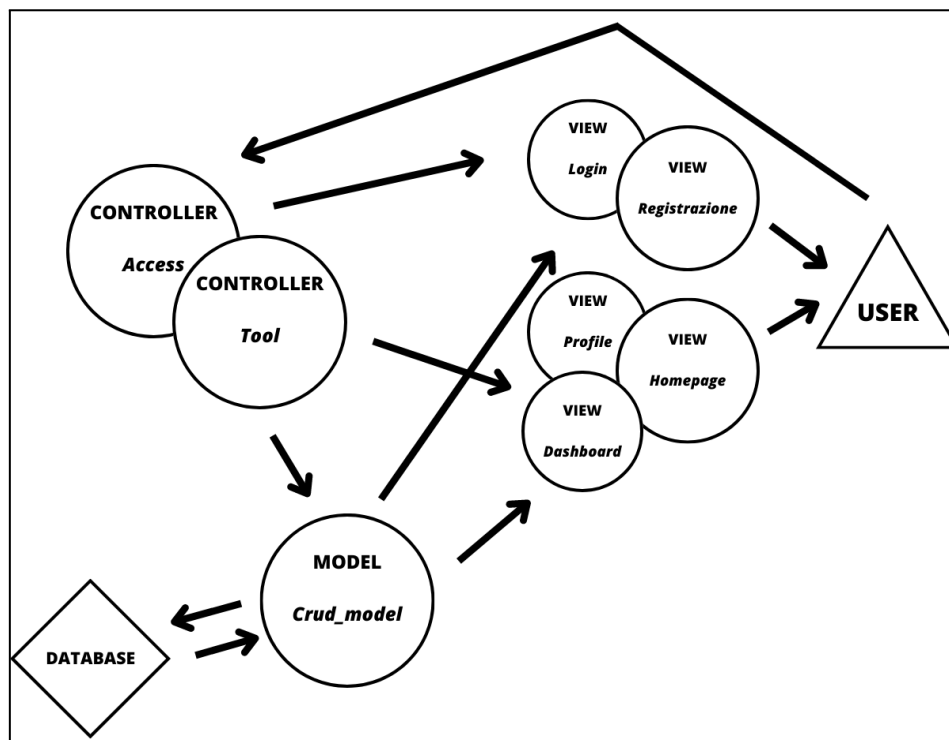
<sup>44</sup> <https://www.html.it/pag/18299/il-pattern-mvc/>

<sup>45</sup> Booch 1998: 37

- **Astrazione:** è la caratteristica essenziale degli oggetti che permette loro di distinguersi da altri oggetti; la capacità di osservare e riconoscere questi confini concettuali dipende dall'osservatore;
- **Gerarchia:** i sistemi complessi sono scomponibili in sotto-sistemi intercorrelati, a loro volta formati dal loro sottosistema fino al raggiungimento degli elementi più basilari. La gerarchia di un sistema è quindi l'ordinamento della sua astrazione;
- **Incapsulamento o *Information binding*:** è il processo di compartimentazione di un elemento, suddividendolo in un'interfaccia e nella sua implementazione;
- **Modularità:** è la proprietà di un sistema di essere composto da parti (moduli) indipendenti ma comunicanti.

Un approccio *object-oriented* quindi fornisce diversi vantaggi come una migliore organizzazione e strutturazione del software, maggiore robustezza e riutilizzabilità e migliore leggibilità e facilità nella manutenzione.

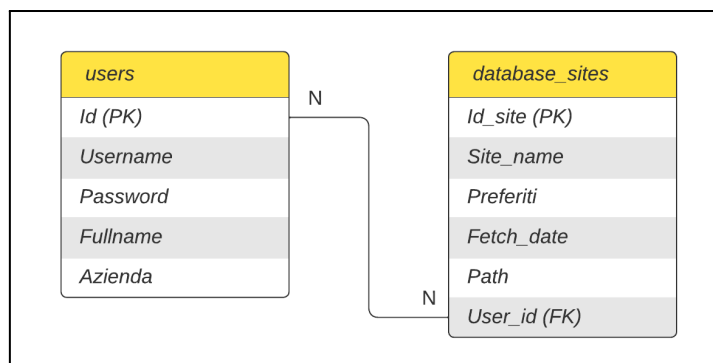
SEOGull è composto quindi da un Model che si occupa di definire i metodi *CRUD* (*Create, Read, Update e Delete*) di manipolazione dei dati con un *database*, due Controller che si occupano della gestione dell'accesso al sito e delle sue funzionalità e cinque View che definiscono la *Graphical User Interface* del progetto.



**Immagine 17:** Rappresentazione dell'architettura MVC di SEOGull.

Inoltre, attraverso il software multiplatforma XAMPP, si è utilizzato il *web server* locale Apache ed il *Database Management System* (DBMS) MariaDB e SQLite.

Il *database* realizzato è composto da sole due tabelle, una che ospita le informazioni degli utenti (*users*) e l'altra che contiene i dati dei siti analizzati (*database\_sites*), in relazione molti a molti. Infatti, un utente può analizzare più siti web ed uno stesso sito web può essere analizzato da più utenti diversi.



**Immagine 18:** Modello ER del *database*.

La tabella *users* memorizza i dati degli utenti in fase di registrazione, con l'unica modifica del valore della *password*, il quale viene criptato attraverso la funzione `password_hash` di PHP. La crittografia avviene tramite *bcrypt*, una funzione di *hash one-way* <sup>46</sup>, mentre per confrontare il valore di *password* criptografato del *database* con il valore inserito dall'utente in fase di *login* viene eseguito un controllo attraverso la funzione `password_verify` <sup>47</sup>, la quale cripta l'*input* utente e lo confronta con il secondo valore già criptato, restituendo un valore booleano *true* se le due *password* combaciano.

La tabella *database\_sites* invece non salva al suo interno le dirette informazioni dei siti analizzati, ma memorizza un percorso relativo di una cartella dedicata nella *directory* del software che indica al Controller dove poter reperire diversi file JSON contenenti i dati scaricati. Tali file JSON sono l'*output* di tre diversi file Python che consistono nel vero e proprio *core* dell'intero progetto.

### 2.3.3 Il Web Crawler

SEOGull infatti non è una sola *web application* che permette all'utente di potersi registrare e visualizzare dei dati, ma è composto da un vero e proprio *web crawler*. Questo *bot* è stato realizzato con il linguaggio di programmazione Python, il quale,

<sup>46</sup> <https://www.php.net/manual/en/function.password-hash.php>

<sup>47</sup> <https://www.php.net/manual/en/function.password-verify.php>

grazie alla sua versatilità e ampia collezione di librerie, è risultato essere un ambiente di facile sviluppo.

Il software è composto da tre file Python, richiamati ed eseguiti dal Controller del programma attraverso delle chiamate AJAX, le quali permettono di popolare la GUI in maniera asincrona al termine dei processi di calcolo. L'esecuzione dei file viene gestita da PHP, attraverso la sua funzione nativa `exec`<sup>48</sup>, la quale permette il richiamo di uno *script* esterno, specificando il percorso del compilatore di Python ed il percorso del file stesso; ulteriori informazioni vengono altresì inviate in coda alla stringa, separandole con uno spazio. I file python raccoglieranno tali dati attraverso il modulo `sys`, salvandoli in delle variabili.

Di seguito si illustreranno i file in questione, al fine di comprendere appieno il funzionamento del progetto svolto.

- **crea\_cartella.py**

È il primo file che viene richiamato ed è il responsabile della creazione della cartella dedicata al sito nella *directory* dell'utente.

Come informazioni in *input* vengono inviati l'URL specificato dall'utente ed il suo *username*; una volta ricavato il dominio dall'URL attraverso il modulo `urllib.parse`<sup>49</sup>, viene specificata la destinazione per la creazione della nuova cartella utilizzando le variabili appena citate.

Nel momento della creazione della cartella, vengono creati anche tre file JSON vuoti, i quali verranno aggiornati come processo di *output* dei restanti due codici python:

- **links\_sitemap.json**: in questo file verranno salvati i link della *sitemap* del sito se rilevata;
- **links\_to\_scrape.json**: questo file sarà composto dai link da analizzare e dal quale il *crawler* preleverà gli URL da navigare. Il primo elemento della lista, posizionato all'indice numero 0, è una coppia di variabili che contengono meta-informazioni del file, indicando quale dovrà essere l'*id* del prossimo URL da analizzare e quanti sono i link totali ricavati;
- **scraped\_links.json**: in questo file invece verranno memorizzate le informazioni delle pagine web navigate. Ogni elemento della lista sarà un oggetto che, in caso di navigazione completata con successo, conterrà le seguenti informazioni:
  - *Id*;
  - *URL*;

---

<sup>48</sup> <https://www.php.net/manual/en/function.exec.php>

<sup>49</sup> <https://docs.python.org/3/library/urllib.parse.html#module-urllib.parse>



- Data di analisi;
- Protocollo di trasferimento;
- Codice di risposta HTTP;
- Numero di link interni al sito;
- Numero di link esterni al sito;
- Presenza dell'URL navigato all'interno della *sitemap* del sito;
- Presenza di *robots meta-tag* Noindex;
- Presenza di *robots meta-tag* Nofollow;
- Permesso di navigazione dell'URL da parte del file *robots.txt*;
- *Core Web Vitals*;
- Le prime dieci *keywords* della pagina, rilevate per importanza;
- Titolo HTML dell'URL;
- Linguaggio HTML dell'URL;
- URL canonico se presente;
- Ricorrenza degli *htags* HTML;
- Presenza di un *alt tag* in tutte le immagini dell'URL.

Se il processo di *crawling* si risolverà con un esito negativo, le informazioni dell'oggetto si limiteranno ad Id, URL, Data di analisi e Codice di risposta HTTP.

- **cerca\_sitemap.py**

È il secondo file che viene richiamato in successione al precedente, all'interno della condizione di `success` della prima chiamata AJAX del codice che avvia l'analisi di un nuovo sito web.

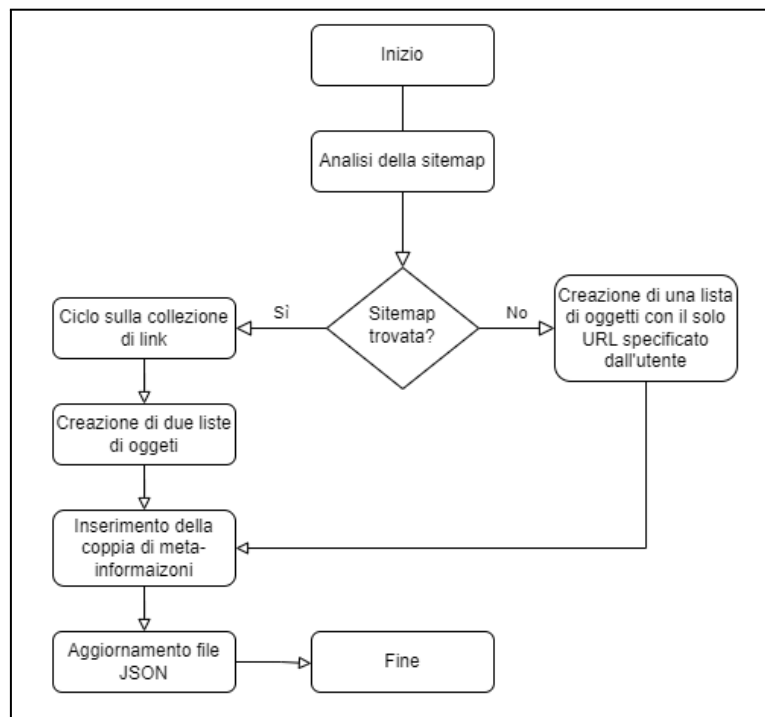
Il codice prende in *input* il dominio della pagina e lo *username* dell'utente per poter interagire con i file della *directory*, e l'URL del sito da navigare. Richiamando la funzione `sitemap_tree_for_homepage` del modulo `usp.tree`<sup>50</sup> attraverso la variabile contenente l'URL specificato dall'utente, verrà cercata la *sitemap* del sito web e se l'esito avrà successo, ne verranno ricavati tutti i link. Successivamente, attraverso un ciclo sulla lista dei link analizzati, ad ogni sua iterazione verranno popolate due ulteriori liste, composte da un oggetto contenente un id progressivo ed un link. Ad una delle due liste verrà successivamente preposto all'intera collezione di oggetti, un'ulteriore coppia contenente le meta-informazioni citate precedentemente.

Infine verranno aggiornati i file JSON: il file `links_sitemap.json` verrà riempito dalla lista contenente i soli oggetti con id e URL, mentre il file `links_to_scrape.json` verrà sovrascritto con i valori della lista con anche le meta-informazioni; qualora invece la ricerca della *sitemap* non producesse risultati, verrà aggiornato solamente il file `links_to_scrape.json` contenente una lista con le meta-informazioni e la coppia id, URL del link specificato dall'utente ed utilizzato per la ricerca.

---

<sup>50</sup> <https://ultimate-sitemap-parser.readthedocs.io/en/latest/usp.html#module-usp.tree>

Questa apparente creazione ridondante di due file simili trova motivazione nel loro stesso ruolo. Infatti, lo scopo del file `links_sitemap.json` è quello di visualizzare i link presenti nella *sitemap* del sito, se presente; da tale lista partirà la collezione di link dalla quale il *crawler* analizzerà il sito e, come verrà mostrato successivamente, ad ogni navigazione degli URL, solo il file `links_to_scrape.json` verrà aggiornato con nuovi link, permettendo una mappatura quanto più completa del sito web.



**Immagine 19:** Flowchart del codice `cerca_sitemap.py`

- **analisi\_crawler.py**

Questo file viene richiamato dalla *call to action* di analisi nella pagina di *dashboard* e si occupa di gestire l'analisi effettiva del *crawler*.

Come per il file `cerca_sitemap.py`, anche in questo vengono raccolti come dati di *input* il dominio del sito web e lo *username* per poter navigare nella *directory* e poter recuperare la lista dei link da dover navigare; in aggiunta a queste due informazioni, viene inviato anche il numero massimo di link navigabili, il quale diventerà il numero di iterazioni del ciclo di analisi.

Questo file si presenta in maniera più articolata rispetto ai primi due in quanto è composto da diverse funzioni e impiega diversi moduli per poter eseguire le azioni richieste. La funzione principale che viene richiamata è `crawl`, la quale innanzitutto inizializza delle liste vuote e definisce delle variabili per controlli successivi, come ad esempio due liste contenenti rispettivamente tutti i link della *sitemap* e dei link da navigare da poter confrontare con i dati dell'URL navigato.

Viene successivamente definito un ciclo di tante iterazioni quanto il numero specificato dall'utente, fino ad un massimo di dieci, all'interno delle quali, attraverso un costrutto `try-catch`, si tenta di navigare un URL. Il link da navigare, come già accennato, viene recuperato attraverso le meta-informazioni del file `links_to_scrape.json`, attraverso le quali il codice ne recupera l'id. Se il recupero avrà successo, l'URL verrà richiamato in una sessione HTML attraverso il modulo `requests_html`<sup>51</sup>; esso si basa sul modulo `requests`<sup>52</sup> che permette di inviare delle richieste GET con protocollo HTTP/1.1. Il codice, sempre grazie al modulo `requests_html`, provvede successivamente al *rendering* Javascript della pagina per poterne caricare eventuali dati dinamici, attendendo 5 secondi per lasciare elaborare il processo che possiede un *timeout* di 180 secondi, al termine del quale il caricamento della pagina risulterà fallito.

Il *rendering* della richiesta viene eseguito attraverso Selenium, il quale ricarica la stessa attraverso delle API di Chromium, un *browser open-source* realizzato da Chrome<sup>53</sup>. Selenium è uno strumento di automazione, utilizzato per simulare il comportamento di interazione degli utenti con i *browsers* ed esso dispone della funzionalità di navigazione attraverso un *Headless Browser*, ovvero un normale *browser* senza la sua GUI, manipolabile attraverso delle linee di comando<sup>54</sup>. La scelta di utilizzare un *Headless Browser* anziché un browser normale deriva dal fatto che questa tipologia di strumento rispetta il principio di *Information binding* di Booch ed inoltre è una pratica molto utilizzata per il *web scraping*; infatti, il risultato di riuscito caricamento della pagina da parte del codice consiste in un oggetto *request* contenente tutte le informazioni dell'URL navigato ed esplorabile dal modulo `requests_html`.

A questo punto inizia la vera e propria analisi dei dati:

- **Title ed *htags*:** grazie alla funzione `find` del modulo, richiamata sull'oggetto *request*, è possibile specificare un *tag* HTML e la funzione produrrà una lista con tutti gli elementi trovati<sup>55</sup>. Grazie a questa funzione vengono rilevati direttamente i *tag* `<Title>` e gli *htags* da `<H1>` a `<H6>`;
- **Linguaggio della pagina, *alt tag*, link canonico e *robots meta-tags*:** per queste informazioni risulta necessario un ulteriore passaggio rispetto al punto precedente. Infatti, tali elementi sono degli attributi di altri *tag*, perciò, con la funzione `find` vengono rilevati questi ultimi, e tramite la proprietà `attrs` si specifica l'attributo richiesto, salvando i

---

<sup>51</sup> <https://requests.readthedocs.io/projects/requests-html/en/latest/>

<sup>52</sup> <https://requests.readthedocs.io/en/latest/>

<sup>53</sup> <https://www.chromium.org/chromium-projects/>

<sup>54</sup> Shariff, et. al. 2019: 15

<sup>55</sup> <https://requests.readthedocs.io/projects/requests-html/en/latest/>

risultati in variabili distinte. Una piccola differenza si applica per gli *alt tag*, in quanto prima di tutto vengono trovati tutti i *tag* delle immagini della pagina e, successivamente, si analizza se ognuno di essi possiede l'attributo cercato, popolando una lista di valori booleani come risposta. Se tutta la lista risulterà composta da valori *true*, significherà che ogni immagine possiede un attributo *alt tag* non vuoto e di conseguenza il valore della variabile di questa informazione indicherà *true*, in caso contrario sarà *false*.

- **Link interni ed esterni:** attraverso l'attributo `absolute_links` sull'intero testo dell'oggetto vengono recuperati tutti i link assoluti e salvati in una lista. Attraverso un ciclo su tale lista, viene determinato il dominio di ogni link e confrontato con quello della pagina navigata, salvando poi l'intero URL dell'iterazione nella lista dei link interni, se i due domini combaciano, o nella lista di quelli esterni in caso contrario. Le due liste vengono infine conteggiate, determinando quanti link interni ed esterni sono presenti nella pagina web.  
Con la lista dei link interni viene inoltre aggiornata la lista degli URL del file `links_to_scrape.json`, aggiungendo i link che non sono già presenti nel file ed attribuendo loro un id, modificando in seguito le meta-informazioni del file;
- **HTTP status:** per rilevare questa informazione viene semplicemente richiamata la funzione `status_code` del modulo `request` sull'oggetto *response*;
- **Presenza nella *sitemap*:** viene eseguito un semplice controllo per verificare la presenza dell'URL navigato all'interno di una lista con i link del file `links_sitemap.json`;
- **Blocco di navigazione per le condizioni del *robots.txt*:** prima di tutto è necessario reperire il file *robots.txt*. Per ricavarlo si elimina qualsiasi elemento successivo al dominio, nell'URL analizzato, e si aggiunge la stringa `"/robots.txt"`; questo perchè, come detto nello scorso capitolo, il funzionamento del file è vincolato alla sua presenza nella *directory* principale del sito. Attraverso il modulo `Protego`<sup>56</sup> si analizza il file, se trovato, e se ne raccolgono le direttive generiche per qualsiasi *user-agents*; infine, attraverso la funzione `can_fetch` del modulo si riceve un valore booleano di risposta;

---

<sup>56</sup> <https://pypi.org/project/Protego/>

- **Core Web Vitals:** per calcolare questi valori, ci si connette alle API del servizio Google PageSpeed Insights <sup>57</sup> tramite una chiave segreta del proprio account Google. L'URL alla quale si invierà la richiesta sarà composto da diversi valori:

- **Una parte di URL che specifica il dominio ed il percorso del file:**

*<https://www.googleapis.com/pagespeedonline/v5/runPagespeed>*

- **La strategia di analisi:** questa può differenziarsi in *desktop* oppure *mobile*;
- **L'URL da analizzare;**
- **La chiave segreta.**

Definiti due URL ai quali eseguire la *request*, uno per ogni tipo di strategia, si ricavano due *report* di PageSpeed Insights dai quali si filtrano i valori assoluti dei *Core Web Vitals* (CLS, FID, LCP e Speed Index), i relativi valori percentuali ed anche un punteggio del sito web analizzato.

- **Keywords:** per rilevare quali siano le principali *keywords* all'interno del testo della pagina, si è utilizzata l'equazione TD-IDF (*Term Frequency–Inverse Document Frequency*). L'applicazione di questo metodo di calcolo non definisce quali siano le parole più utilizzate basandosi sulla *keyword density*, ma anzi, attribuisce un valore ponderato alle parole in funzione della loro importanza in un testo. Per applicare l'equazione si determina prima di tutto il corpo di testo della pagina web attraverso il modulo BeautifulSoup <sup>58</sup>, il quale viene ripulito dai caratteri non alfanumerici e da una libreria di *stopwords* inglese oppure italiana a seconda del valore inserito nell'attributo di lingua della pagina HTML. Si individua in seguito il numero totale di sentenze e parole, calcolando i valori TF ed IDF di queste ultime, moltiplicandoli poi tra di loro; dalla lista finale di *keywords* si individuano infine i primi dieci termini.

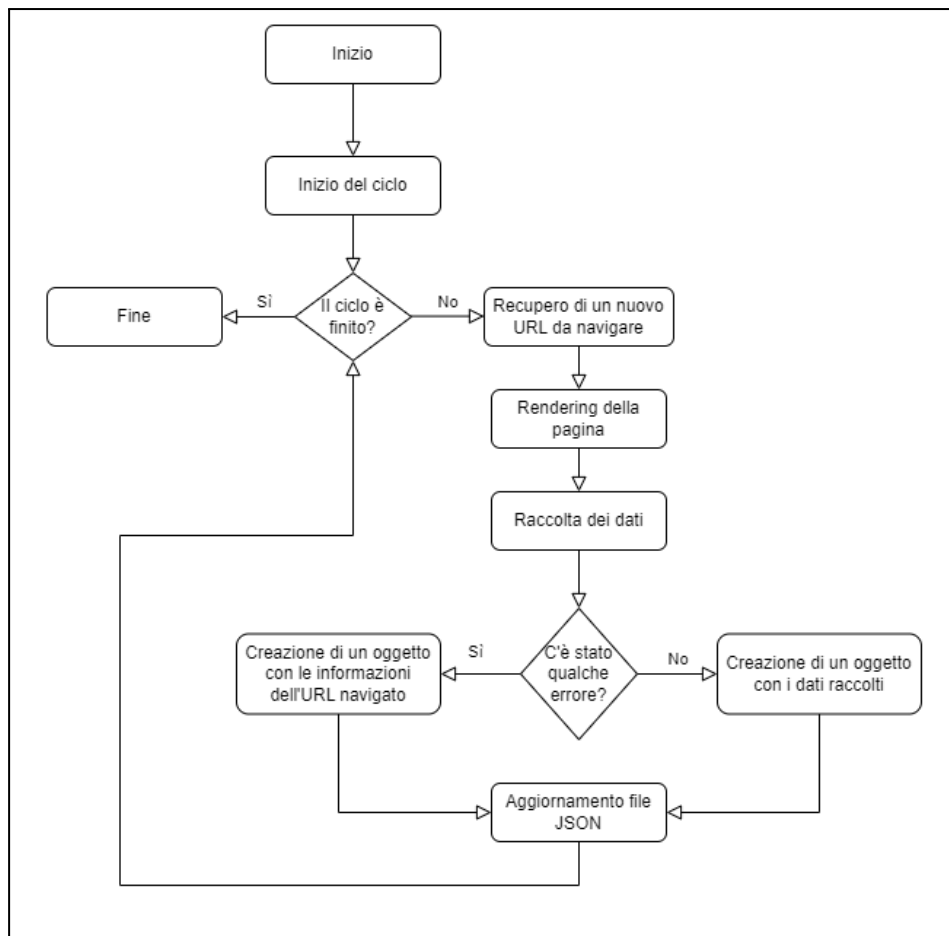
Una volta collezionate tali informazioni, esse verranno memorizzate all'interno di un oggetto che verrà poi aggiunto al file `scraped_links.json`. Se tuttavia il costrutto `try-catch` dovesse generare una qualsiasi eccezione durante l'esecuzione di uno dei precedenti calcoli, l'oggetto creato conterrà solamente

---

<sup>57</sup> <https://developers.google.com/speed/docs/insights/v5/get-started?hl=it>

<sup>58</sup> <https://beautiful-soup-4.readthedocs.io/en/latest/>

le principali informazioni sull'URL navigato, lasciando ad eventuali funzioni di *debug* il compito di scoprire la causa dell'errore nell'analisi della pagina.



**Immagine 20:** *Flowchart* del codice `analisi_crawler.py`

## 2.4 Risposta alla domanda di ricerca

Per rispondere alla domanda di ricerca: sì, è stato possibile realizzare un software di analisi di un sito web che potesse collezionare dati utili alla SEO; è stata altresì realizzata una GUI ed una gestione di account multipli grazie all'impiego di un database.

Nel prossimo capitolo si affronteranno i limiti del progetto realizzato e le sue prospettive di sviluppo e miglioramento.

# CAPITOLO 3

## 3.1 Limiti del progetto

Il progetto realizzato presenta molte limitazioni e margini di miglioramento in quanto il software creato non si vuole intendere come finito, quanto piuttosto un *Minimum Viable Product* (MVP), ovvero una versione minimale di un prodotto, sviluppata quanto basta per poter essere sottoposta a test di analisi utili al fine di raccogliere spunti di perfezionamento.

Di seguito si indicheranno i limiti già noti per poi passare alla spiegazione di un'analisi effettuata su di un sito web ed ai relativi *feedback* del *webmaster*.

- **Limiti noti**

L'applicazione realizzata non si vuole certo porre come *competitor* dei *tool* di analisi citati ad inizio dello scorso capitolo, innanzitutto per la sua permanenza in locale su una macchina di calcolo e non *online*. Non potendosi intendere come una *web application* terminata, si è preferito mantenere *offline* il progetto, in quanto numerosi potrebbero essere i problemi riscontrabili.

- **Dati di laboratorio e dati di utenti reali**

L'analisi delle informazioni maggiormente legate all'aspetto qualitativo della navigazione dell'utente ad alla sua UX, come ad esempio i *Core Web Vitals*, impiega questa distinzione di dati.

I dati di laboratorio sono dei dati raccolti in un ambiente di test, vincolato al periodo ed alle caratteristiche del dispositivo e della connessione ad Internet che esegue l'analisi. Inoltre, gli emulatori che Google utilizza per simulare i dispositivi *desktop* e *mobile* si riferiscono ad una sola tipologia di *device*. Queste informazioni raccolte sono perciò riferite allo specifico contesto nel quale vengono prodotte e non si riferiscono agli effettivi dati riscontrabili nella navigazione di utenti reali. Tali dati sono certamente più informativi, ma sono altresì difficili da rilevare, soprattutto attraverso gli strumenti che Google offre, come ad esempio Google Search Console, il quale ad esempio si limita a raccogliere i dati di navigazione degli utenti che utilizzano Google Chrome come *browser*.

I dati degli utenti reali inoltre sono meno immediati da analizzare, in quanto Google può impiegare fino a 28 giorni di tempo per rendere effettiva una modifica adottata sul sito web, ed inoltre è necessario dover attendere ulteriore tempo per permettere agli utenti di navigare il sito per poterne analizzare le sessioni.

Per questi motivi appena elencati, SEOgull utilizza esclusivamente dati di laboratorio.

- **Hosting e web-scraping**

Come appena scritto, la bontà dei dati raccolti con SEOgull si limita allo specifico contesto e momento in cui essi vengono raccolti; la qualità dell'analisi dipende anche dalla connessione Internet ed alla velocità di risposta del *server* nel quale i codici sono ospitati. Un'esecuzione del software su di una macchina locale abbassa notevolmente i tempi di latenza a differenza di quanto potrebbe succedere se esso fosse reso disponibile *online* su di un server commerciale.

Molti server, inoltre, non permettono di ospitare software di *bot* o software di analisi ed inoltre i motori di ricerca penalizzano pratiche di *web-scraping* se richieste in successioni troppo brevi. Risulterebbe quindi necessario revisionare il codice, inserendo la possibilità di variegare le richieste di analisi, modificando ad esempio lo *user-agent* per evitare che il software venga bloccato.

- **Sicurezza**

Seppur sia stata applicata una pratica di crittografia delle *password* degli utenti, questo non risulta sufficiente per poter garantire la sicurezza da possibili attacchi informatici. Risulterebbe necessario definire uno o più alberi di attacco dell'applicazione per rilevare i punti deboli e sanificare campi di *input* per evitare iniezioni SQL o attacchi *Cross-site Scripting* ad esempio.

- **Natural Language Processing (NLP)**

Come precedentemente scritto, una delle funzioni dello strumento è volta all'analisi delle *keywords* presenti nella pagina secondo il modello TF-IDF; tuttavia, dall'aggiornamento BERT di Google del 2019, il motore di ricerca ha intrapreso un percorso nell'elaborazione del linguaggio naturale, cercando di capire il significato delle *query* dell'utente piuttosto che limitarsi a risultati contenenti le sole parole chiave utilizzate. La stessa Google ha rilasciato uno strumento per l'analisi NLP di un testo <sup>59</sup>, richiamabile attraverso delle API; offrendo quindi un'interessante prospettiva di miglioramento.

- **Informazioni raccolte**

Un altro limite di SEOgull riguarda la varietà dei dati raccolti. Per la versione MVP di questo progetto si è deciso di limitarsi a raccogliere pochi dati, concentrandosi su quelli più informativi e semplici da rilevare.

---

<sup>59</sup> <https://cloud.google.com/natural-language?hl=it>



- **Limiti non noti**

Inoltre, l'applicazione è stata testata attraverso l'analisi del sito web "<https://leimbucate.it/>", condividendone i risultati con le proprietarie per raccogliere *feedback* su l'MVP. I dati raccolti sono i seguenti e sono disponibili in formato JSON all'interno della cartella contenente il codice sorgente del programma:

- Il sito web è composto da 116 pagine;
- Il protocollo utilizzato è *HTTPS*;
- Le direttive del file *robots.txt* non hanno bloccato la navigazione di nessuna pagina web;
- Sei pagine possiedono un *redirect* permanente 301, mentre le restanti pagine generano un *http status* 200;
- Le precedenti sei pagine non sono altresì presenti nella *sitemap*;
- I punteggi dei *Core Web Vitals* indicano una buona progettazione del sito web per la sua versione *desktop* (95.47%) ed una necessità di miglioramento per la versione *mobile* (66.52%), indicando in particolare azioni di manutenzione per quanto concerne il LCP (44.19%);
- La quasi totalità delle pagine del sito contengono immagini senza un *alt tag*.

La condivisione di questi dati è stata valutata in maniera positiva dalle proprietarie del sito web, in quanto ha fornito loro un utile campanello di allarme sulla necessità di ottimizzare aspetti come le immagini e la versione *mobile*, aspetti che, come si è scritto nel primo capitolo, risultano essere fondamentali per un buon posizionamento nella SERP del motore di ricerca. Inoltre, sono state notate la mancanza della possibilità di poter scaricare i dati raccolti in un formato elaborabile da un foglio di calcolo come ad esempio *.csv* e di uno strumento che potesse fornire un supporto nella realizzazione di testi *SEO-friendly*.

Gli spunti di miglioramenti per SEOGull sono risultati interessanti, determinando un buon punto di partenza per il suo sviluppo.

# CONCLUSIONI

La domanda di ricerca posta riguardava la creazione di un software che permettesse di analizzare in chiave SEO i siti web.

Per la riuscita del progetto è stato innanzitutto indispensabile comprendere la storia e l'evoluzione della SEO, determinando inoltre quale fosse l'ambito applicativo della materia. Tra questi ambiti ci si è focalizzati sui motori di ricerca, Google in particolare, essendo il principale attore del mercato, detenente oltre l'80% dello *share* a livello globale.

Si è quindi studiata la storia dell'algoritmo PageRank di Google, spartiacque nella storia della SEO, e la sua evoluzione fino ai giorni nostri, comprendendo quali siano diventati i *topic* della materia, in modo da sapere quali dati cercare e quali pratiche evitare.

Una volta compreso il funzionamento teorico di un *crawler*, si è iniziato a progettare uno attraverso Python, un ambiente di sviluppo che si è rivelato adeguato grazie alle sue numerose librerie e moduli. Una volta creato un *software* funzionante si è passati alla realizzazione di un'interfaccia e di un *pattern* architetturale che potesse gestire il tutto. Per realizzare ciò si è deciso di impiegare il *framework* CodeIgniter, il quale ha permesso la strutturazione di una *web application* che permettesse di interagire con l'utente e di avviare il *software*, manipolando i suoi dati di *output*.

Giunti al rilascio di un MVP del *software* di analisi si è infine analizzato un sito web, condividendo il *report* realizzato con le proprietarie del sito analizzato e stabilendo una lista dei limiti del progetto, sancendo quindi il termine dello stesso ai fini di questa tesi di laurea.

Lo studio compiuto ha chiarito quanto la SEO sia un efficace *driver* per una strategia di *marketing*: investire risorse su una corretta realizzazione di un sito web, applicazione o contenuto *online* garantisce risultati nel lungo periodo, fornendo ai motori di ricerca dei contenuti di qualità da offrire agli utenti e contribuendo a migliorare la reputazione di un *brand*.

Conoscere il funzionamento dei motori di ricerca e seguirne le direttive di progettazione non significa sottostare alle volontà di un'autorità digitale monopolistica, ma al contrario significa comprendere quali siano le ragioni del successo di questi nuovi intermediari dell'informazione al fine di sfruttarne le potenzialità.

Concludendo con un pensiero del sociologo positivista Auguste Comte, “*vedere per prevedere, prevedere per provvedere*”.

# RINGRAZIAMENTI

Ringrazio tutte le persone con le quali ho potuto vivere e condividere questi anni universitari. Ringrazio Floriana per essere una costante in mezzo a tantissime variabili. Ringrazio la mia famiglia che mi ha sempre accompagnato in ogni mio percorso. Ringrazio gli amici di CIME ed i bellissimi lavori di gruppo fatti insieme. Ringrazio tutti i ragazzi e le ragazze del quarto piano per aver condiviso con me opinioni, conoscenze, risate ed insicurezze e per avermi dato un posto a Torino da poter chiamare casa. Ringrazio infine Torino per i bellissimi ricordi che mi ha dato e che forse continuerà a darmi in futuro.

# BIBLIOGRAFIA

## Manuali:

BOOCH, Grady. *Object-oriented analysis and design with applications (2nd ed.)*. Benjamin-Cummings Publishing Co., Inc., USA. 1993.

PACCAGNELLA, Luciano. *Sociologia della comunicazione nell'era digitale*. Società editrice il Mulino spa, 2020.

RICK, Levine, et al. *The Cluetrain manifesto*. 2000.

VAN DIJCK, Josè; POELL, Thomas; DE WAAL, Martijn. *Platform Society: valori pubblici e società connessa*. Edizione italiana a cura di Giovanni Boccia Artieri e Alberto Marinelli. Guerini scientifica, 2019.

## Articoli scientifici:

BLUM, Avrim; CHAN, TH Hubert; RWEBANGIRA, Mugizi Robert. A random-surfer web-graph model. In: *2006 Proceedings of the Third Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*. Society for Industrial and Applied Mathematics, 2006. p. 238-246.

BRIN, Sergey; PAGE, Lawrence. The anatomy of a large-scale hypertextual web search engine. In: *Computer networks and ISDN systems*, 1998, 30.1-7: 107-117.

BURKE, C. J.; ROSENBLATT, M. A Markovian function of a Markov chain. In: *The Annals of Mathematical Statistics*, 1958, 29.4: 1112-1122.

PAGE, Lawrence. Method for node ranking in a linked database. In: *USA Patent*, 1997, 6.

JOSHI, Anuj; PATEL, Priyanka. Google Page Rank Algorithm and It's Updates. In: *International Conference on Emerging Trends in Science, Engineering and Management, ICETSEM-2018*. 2018.

SHANNON, Claude E. A mathematical theory of communication. In: *The Bell system technical journal*, 1948, 27.3: 379-423.

SHARIFF, Shahnaz Mohammadi, et al. Improving the testing efficiency of selenium-based load tests. In: *2019 IEEE/ACM 14th International Workshop on Automation of Software Test (AST)*. IEEE, 2019. p. 14-20.

## Report Statista:

StatCounter(2021), *Web Browsers* [Dossier],  
<https://www.statista.com/study/68729/web-browsers/> (ultimo accesso 14/11/2022).

StatCounter(2022), *Search engines: alternatives to Google* [Dossier],  
<https://www.statista.com/study/85980/search-engines-alternatives-to-google/> (ultimo accesso 14/11/2022).

## Sitografia (ultimo accesso per tutti i link 12/02/2023) :

<http://www.giancarlopalavicini.it/economia/marketing/principi-di-marketing>

<https://ahrefs.com/>

<https://app.neilpatel.com/en/dashboard>

<https://backlinko.com/google-ctr-stats>

<https://beautiful-soup-4.readthedocs.io/en/latest/>

<https://cloud.google.com/natural-language?hl=it>

<https://codeigniter.com/userguide3/>

<https://developers.google.com/search/blog/2019/07/a-note-on-unsupported-rules-in-robotstxt>

[https://developers.google.com/search/docs/advanced/robots/robots\\_meta\\_tag](https://developers.google.com/search/docs/advanced/robots/robots_meta_tag)

<https://developers.google.com/search/docs/crawling-indexing/301-redirects>

<https://developers.google.com/search/docs/crawling-indexing/robots/create-robots-txt>

<https://developers.google.com/speed/docs/insights/v5/get-started?hl=it>

<https://docs.python.org/3/library/urllib.parse.html#module-urllib.parse>

<https://it.semrush.com/blog/pagerank-di-google/>

<https://moz.com/>

<https://pypi.org/project/Protego/>

<https://requests.readthedocs.io/en/latest/>

<https://requests.readthedocs.io/projects/requests-html/en/latest/>

<https://search.google.com/search-console/about>

<https://sparktoro.com/blog/in-2020-two-thirds-of-google-searches-ended-without-a-click/>

<https://themarkup.org/google-the-giant/2020/07/28/google-search-results-prioritize-google-products-over-competitors>

<https://ultimate-sitemap-parser.readthedocs.io/en/latest/usp.html#module-usp.tree>

<https://web.dev/vitals/>

<https://www.chromium.org/chromium-projects/>

<https://www.evemilano.com/blog/>

<https://www.evemilano.com/pagerank/>

<https://www.evemilano.com/wp-content/uploads/pdf/Myth-Of-The-Google-Toolbar-Ranking.pdf>

<https://www.html.it/articoli/guida-fattori-seo-ranking-algoritmo-google/>

<https://www.html.it/pag/18299/il-pattern-mvc/>

<https://www.ietf.org/rfc/rfc2616.txt>

<https://www.insidemarketing.it/glossario/definizione/black-hat-seo/>

<https://www.nytimes.com/2021/07/07/opinion/google-utility-antitrust-technology.html>

<https://www.php.net/manual/en/function.exec.php>

<https://www.php.net/manual/en/function.password-hash.php>

<https://www.php.net/manual/en/function.password-verify.php>

<https://www.rfc-editor.org/rfc/rfc7540>

<https://www.screamingfrog.co.uk/>

<https://www.semrush.com/>

<https://www.seozoom.it/la-guida-ai-200-fattori-di-ranking-su-google/>

<https://www.thinkwithgoogle.com/intl/en-154/marketing-strategies/app-and-mobile/need-mobile-speed-how-mobile-latency-impacts-publisher-revenue/>

<https://www.thinkwithgoogle.com/intl/it-it/strategie/app-e-mobile/migliorare-velocita-sito-mobile/>

[https://www.youtube.com/watch?v=OM6XIIcm\\_qo](https://www.youtube.com/watch?v=OM6XIIcm_qo)

[https://www2.deloitte.com/content/dam/Deloitte/ie/Documents/Consulting/Milliseconds\\_Make\\_Millions\\_report.pdf](https://www2.deloitte.com/content/dam/Deloitte/ie/Documents/Consulting/Milliseconds_Make_Millions_report.pdf)

## **Modello Netlogo:**

Stonedahl, F. and Wilensky, U. (2009). NetLogo PageRank model. <http://ccl.northwestern.edu/netlogo/models/PageRank>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Wilensky, U. (1999). NetLogo. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

## **Video-corsi:**

LUPPARELLI, F., *SEO*, <https://learnn.com/corso/seo/> (ultimo accesso 05/12/2022)

OLOVRAP, L., *SEO Core Web Vitals e ottimizzazione velocità sito web*, <https://my.learnn.com/corso/139> (ultimo accesso 05/12/2022)