



**UNIVERSITÀ
DEGLI STUDI
DI UDINE**
hic sunt futura

**Dipartimento di Scienze
Matematiche, Informatiche e Fisiche**

TESI DI LAUREA IN
INFORMATICA

Sito web statico per gruppo di ricerca accademico: analisi e implementazione

CANDIDATO

Federico Dittaro

RELATORE

Prof. Marino Miculan

CORRELATORE

Dott. Matteo Paier

Anno accademico 2022-2023

CONTATTI DELL'ISTITUTO

Dipartimento di Scienze Matematiche, Informatiche e Fisiche

Università degli Studi di Udine

Via delle Scienze, 206

33100 Udine — Italia

+39 0432 558400

<https://www.dmif.uniud.it/>

Ringraziamenti

Sommario

Indice

1	Introduzione	1
2	Analisi del software	3
2.1	SSG	3
2.2	principali SSG	3
2.2.1	Hugo	3
2.2.2	Jekyll	5
2.2.3	Gridsome	6
2.2.4	Eleventy	7
2.2.5	Pelican	7
2.3	Differenze tra siti statici e dinamici	7
2.4	Hugo	7
3	Realizzazione di un sito web	9
4	Caso di studio	11
5	Conclusioni	13
A	Glossario	15

1

Introduzione

In hac habitasse platea dictumst. Vestibulum consectetur dictum pellentesque. Suspendisse nunc neque, commodo ac imperdiet nec, sollicitudin vitae libero. Donec bibendum vel nunc vitae pharetra. In vel volutpat odio, et interdum dui. Duis mauris ligula, congue eget molestie at, tincidunt nec diam. Nam vitae eros nec arcu suscipit vehicula. Aliquam consectetur imperdiet elit, eget pretium arcu fringilla at. Maecenas [1] sed libero pulvinar, mattis tortor vel, fermentum enim.

2

Analisi del software

2.1 SSG

Gli SSG (in inglese, Static Site Generator) sono dei tool che permettono la creazione di tutti i contenuti presenti nei siti web a partire da file di configurazione e contenuti scritti in formati più generali (tipicamente markdown). La caratteristica principale di tali siti è che a fronte di una richiesta da parte dell'utente di visualizzare determinati contenuti del sito, il web server fornisce pagine statiche, delle quali l'utente non è in grado di modificare il contenuto né possiede alcun tipo di stato che ne permetta la personalizzazione. Non esiste quindi una elaborazione back-end sul lato server e non esistono database, qualsiasi funzionalità "dinamica" associata al sito statico viene eseguita sul lato client. I principali vantaggi riguardanti la scelta di utilizzo di un SSG sono:

- Ottimizzazione delle prestazioni: avendo poche o nessuna parte dinamica sono più facili da ottimizzare ed il caricamento è molto rapido;
- Richiesta di meno risorse al server: dato che non è richiesta nessuna elaborazione lato server, quest'ultimo svolge meno lavoro migliorando prestazioni e scalabilità;
- Servizio di hosting molto economico: possono essere utilizzati per la pubblicazione servizi di host completamente gratuiti come GitHub Pages (esattamente come nel caso di studio);
- Maggiore sicurezza: non utilizzando server o database sono molto sicuri da eventuali attacchi esterni.

2.2 principali SSG

In questa sezione verranno menzionati i cinque principali SSG e discusse le principali differenze tra loro.

2.2.1 Hugo

Hugo è un generatore di siti web statici scritto in Go ideato inizialmente da Steve Francia nel 2013 e successivamente sviluppato da Bjørn Erik. L'ultima versione è stata rilasciata a Luglio 2023. Per utilizzare Hugo non è necessario conoscere Go in quanto il sito web viene creato attraverso file HTML



Figura 2.1: Hugo logo

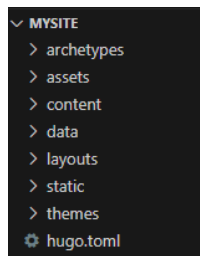


Figura 2.2: Struttura delle cartelle di Hugo

e CSS, inoltre c'è una separazione tra il contenuto e la presentazione permettendo così di modificare l'aspetto senza modificarne il contenuto. Oltre ai tipi di file citati precedentemente, Hugo supporta anche file di tipo javascript, Markdown, TOML, YAML e JSON. Le informazioni necessarie per creare, modificare, stilizzare o eliminare pagine e/o contenuti sono racchiuse all'interno di specifiche cartelle che possono essere successivamente estese. La struttura generale è la seguente:

- la cartella *archetypes* contiene i file che vengono utilizzati come template per la creazione di nuovi contenuti del sito in modo da standardizzare la struttura ed il formato;
- la cartella *content* è forse la più importante in quanto contiene tutto il contenuto del sito. Al suo interno tutti i file sono in formato Markdown;
- la cartella *data* contiene esclusivamente file di tipo JSON, TOML, YAML o XML utilizzati per aggiungere strutture specifiche al sito;
- la cartella *layouts* contiene file HTML usati per creare l'aspetto visivo del sito;
- la cartella *static* contiene file statici come ad esempio immagini, file CSS, file Javascript;
- la cartella *themes* contiene i file che definiscono il tema del sito ed il suo aspetto visivo;
- il file *config.toml* oppure *config.yaml* è fondamentale in quanto rappresenta il file di configurazione e contiene informazioni globali come il titolo del sito, la sua descrizione e molto altro.

Hugo è noto per la sua velocità ed inoltre supporta una grande varietà di temi scaricabili direttamente dal sito ufficiale. A differenza di altri SSG, Hugo non è indicato solamente per la creazione di blog ma anche per la creazione di siti generici come ad esempio siti aziendali o, come nel caso di studio, per siti accademici. Hugo mette inoltre a disposizione una grande varietà di Plugin molto utili come ad esempio il servizio per la rappresentazione delle icone social o il supporto multilingua.



Figura 2.3: Jekyll logo

2.2.2 Jekyll

Jekyll è un generatore di siti web statici, ideato da Tom Preston-werner, la prima versione del software risale al 2008 mentre l'ultima, la 4.1.0, è uscita il 27 maggio 2020. Jekyll si basa sul linguaggio Ruby, perciò richiede un'installazione ed una configurazione corretta e funzionante di tale ambiente. Successivamente si scarica la versione desiderata di Jekyll e si segue la procedura di installazione, così come descritta sulla documentazione. In Jekyll tutti i contenuti e i layout del sito vengono salvati localmente e vengono classificati in una struttura a cartelle, principalmente orientata alla costruzione di blog. Una volta creato il sito, la struttura trovata sarà la seguente:

- la cartella `_posts` contiene gli articoli del sito (composti da file Markdown);
- i contenuti delle pagine, sempre composti da file Markdown, sono salvati nella cartella `root` del sito, in alternativa si può decidere di creare una gerarchia di sottocartelle per una migliore organizzazione dei contenuti;
- la cartella `_layouts` contiene i vari template del sito che decidono la grafica delle singole pagine e dei singoli articoli (questi file sono sempre di tipo HTML);
- la cartella `_site` contiene tutte le informazioni necessarie per esportare il sito funzionante nel dominio del sito o in sistemi cloud;
- la cartella `_data` può essere creata per contenere dei file JSON in cui saranno costruiti dei database per immagazzinare stringhe, numeri e altri dati simili;
- la cartella `assets` contiene immagini, pdf o altri file statici per il sito.

Come Hugo anche Jekyll mette a disposizione centinaia di temi prefabbricati per aiutare lo sviluppo del sito web, ed entrambi forniscono degli shortcode, ovvero funzioni che permettono la comunicazione tra i layout delle pagine con i loro contenuti (ad esempio le template actions per Hugo). Anche Jekyll presenta una moltitudine di Plugin che possono essere integrati attraverso Ruby, permettendo di aggiungere e

semplificare la costruzione di determinati servizi per il sito web. Una delle differenze principali di Hugo rispetto a Jekyll è che il primo non è legato ad ambienti esterni, infatti dopo aver scaricato la versione desiderata ed estratto il contenuto nella cartella prescelta il software è pronto per essere usato, mentre Jekyll si deve appoggiare a Ruby. In conclusione, Jekyll è un'ottima scelta se si ha familiarità con l'ambiente Ruby o se si vuole costruire un sito complesso usando gli innumerevoli Plugin e template messi già a disposizione.

2.2.3 Gridsome

Gridsome è un SSG molto recente, è stato infatti ideato da Johannes Schickling nel 2018 subendo poi miglioramenti negli anni successivi fino all'ultima versione disponibile, la 0.7.23, rilasciata a settembre 2021. Si tratta di un framework basato su Vue.js e GraphQL che permette di creare una configurazione "headless", consentendo così di sfruttare la separazione dei contenuti dalla loro presentazione. Le cartelle di lavoro possono variare in base alla configurazione specifica di un progetto, ma in generale sono strutturate nel modo seguente: Come si vede in Fig. 2.4



Figura 2.4: Gridsome logo

- *src* è la cartella principale al cui interno si trova il contenuto sorgente. È suddivisa in sottocartelle come *assets* per file statici, *components* per componenti Vue.js, *layouts* per i layout del sito, *pages* per le pagine principali e *templates* per i template utilizzati per la generazione di pagine dinamiche;
- *static* è la cartella utilizzata per i file statici che verranno serviti direttamente, come immagini, file CSS o JavaScript;


```

my-gridsome-project/
├─ node_modules/
├─ src/
│  ├─ assets/
│  ├─ components/
│  ├─ layouts/
│  ├─ pages/
│  └─ templates/
├─ static/
├─ .gridsome/
├─ gridsome.config.js
├─ package.json
├─ README.md
└─ ...

```

Figura 2.5: Struttura delle cartelle di Gridsome

- *.gridsome* è la cartella che contiene le configurazioni specifiche di Gridsome. Include i file di configurazione e i dati temporanei generati durante la compilazione del sito;
- *gridsome.config.js* è il file di configurazione principale, definisce le impostazioni globali e le opzioni del progetto.
- *package.json* è il file che definisce le dipendenze del progetto e gli script personalizzati.

Anche Gridsome offre una vasta raccolta di plugin che possono essere utilizzati per estendere le funzionalità del generatore ma a differenza degli altri SSG presentati non è così adatto ai principianti, necessita infatti di una certa esperienza nello sviluppo web per poter riuscire a trarre il massimo da questo software.

2.2.4 Eleventy

2.2.5 Pelican

2.3 Differenze tra siti statici e dinamici

2.4 Hugo

3

Realizzazione di un sito web

4

Caso di studio

5

Conclusioni

A

Glossario

Bibliografia

- [1] Donald E. Knuth. *The TeXbook*. Addison-Wesley, 1986.