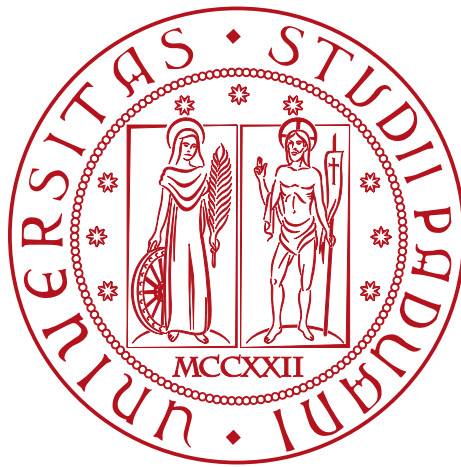


Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

CORSO DI LAUREA IN INFORMATICA



Legacy modernization di un blog WordPress: il case study di Girolibero

Tesi di laurea triennale

Relatore

Prof. Davide Bresolin

Laureando

Francesco Furno

Matricola 2042327

ANNO ACCADEMICO 2023-2024

Dedicato ai miei nonni Lina e Angelo.

Sommario

Il documento presenta il lavoro che ho svolto presso l'azienda Girolibero srl dal 29-04-2024 al 24-06-2024.

Gli obiettivi di questa esperienza formativa sono stati: il rinnovamento del blog aziendale e, nello specifico, la definizione di un nuovo tema WordPress per il sito, l'aggiornamento dell'interfaccia grafica e l'inserimento di diverse funzionalità.

Gli obiettivi prefissati sono stati quasi tutti raggiunti: ho realizzato il backend e il frontend del nuovo blog aziendale, collegando i dati dei viaggi provenienti dal sito principale di Girolibero, inserendo un form di iscrizione alla newsletter e integrando Cloudinary come DAM.

Life seems so much simpler when you're fixing things.

— Anakin Skywalker

Ringraziamenti

Innanzitutto vorrei esprimere la mia gratitudine al Prof. Davide Bresolin, relatore della mia tesi, per la supervisione durante la stesura di questo lavoro.

Desidero ringraziare i miei genitori, Annalisa e Roberto, per l'affettuosa presenza e gli incoraggiamenti.

Infine ringrazio Lucrezia, che mi ha supportato e sopportato durante l'ultimo anno di studi.

Vicenza, luglio 2024

Francesco Furno

Indice

1. Introduzione	1
1.1. L'azienda	1
1.1.1. Il team di lavoro	1
1.2. La proposta di stage	2
1.3. Obiettivi	3
1.4. Organizzazione del testo	4
1.4.1. Convenzioni stilistiche	5
2. Strumenti e metodologie di lavoro	7
3. Analisi dei requisiti	9
3.1. Introduzione ai casi d'uso	9
3.2. Attori individuati	9
3.3. WordPress	10
3.4. Elenco dei casi d'uso	10
3.4.1. UC1 - Inserimento coordinate geografiche	11
3.4.2. UC2 - Visualizzazione lista coordinate geografiche	12
3.4.3. UC3 - Rimozione coppia di coordinate geografiche	14
3.4.4. UC4 - Inserimento blocco "indice"	14
3.4.5. UC5 - Rimozione blocco "indice"	15
3.4.6. UC6 - Visualizzazione blocco "indice"	15
3.4.7. UC7 - Inserimento blocco "carosello viaggi"	16
3.4.8. UC8 - Rimozione blocco "carosello viaggi"	16
3.4.9. UC9 - Visualizzazione blocco "carosello viaggi"	17
3.4.10. UC10 - Inserimento viaggio nel blocco "carosello viaggi"	20
3.4.11. UC11 - Visualizzazione errore: codweb non presente	20
3.4.12. UC12 - Rimozione viaggio dal blocco "carosello viaggi"	21
3.4.13. UC13 - Inserimento descrizione autore in fondo all'articolo	21
3.4.14. UC14 - Rimozione descrizione autore dal fondo dell'articolo	22
3.4.15. UC15 - Visualizzazione mappa geografica degli articoli	22
3.4.16. UC16 - Visualizzazione lista articoli	24
3.4.17. UC17 - Visualizzazione articoli più recenti	28
3.4.18. UC18 - Visualizzazione articoli per categoria	28
3.4.19. UC19 - Visualizzazione articoli più popolari	28
3.4.20. UC20 - Visualizzazione articoli per autore	29
3.4.21. UC21 - Visualizzazione articoli per tag	29
3.4.22. UC22 - Visualizzazione singolo articolo	29
3.4.23. UC23 - Ricerca all'interno del blog	32
3.4.24. UC24 - Iscrizione alla newsletter	33
3.4.25. UC25 - Visualizzazione errore newsletter	36

3.5. Requisiti	37
4. Tecnologie	47
4.1. WordPress	47
4.2. Twig e Timber	48
4.3. Sass	49
4.4. jQuery	50
4.5. Node.js	50
4.6. React	51
4.7. Cloudinary	51
4.8. MapBox	52
5. Sviluppo	53
5.1. Il tema	53
5.1.1. Struttura del tema	53
5.2. Menu	56
5.2.1. Barra di ricerca	56
5.2.2. Pagine statiche	56
5.2.3. Categorie	56
5.2.4. Menu mobile	57
5.3. Lista degli articoli	58
5.3.1. Creazione della lista	59
5.3.2. Caricamento di altri elementi	59
5.4. Articolo singolo	60
5.4.1. Tassonomie	61
5.4.2. Sezione autore	61
5.4.3. Il blocco indice	62
5.5. Gestione delle immagini	63
5.6. Pagine statiche	64
5.7. Newsletter	65
5.7.1. Funzionamento	65
5.7.2. Visualizzazione finale	65
5.8. Mappa degli articoli	66
5.8.1. Coordinate	66
5.8.2. Preparazione dei dati	67
5.8.3. Inserimento dei punti	67
5.8.4. Selezione di un elemento	68
5.8.5. Visualizzazione della mappa	68
5.9. Il carosello viaggi	69
5.9.1. Struttura del plugin	70
5.9.2. Collegamento all'API viaggi di Girolibero	71
5.9.3. Creazione del blocco wordpress	71
5.9.4. Visualizzazione nell'articolo	72
6. Revisione	73
6.1. Interventi	73
6.1.1. Errori di formattazione	73
6.2. Linee guida per la formattazione	74

6.2.1. Introduzione	74
6.2.2. Contenuto	74
6.2.3. Conclusione	75
6.2.4. Altre direttive	76
6.3. Accessibilità	76
6.3.1. Navigazione	76
6.3.2. Immagini	77
6.3.3. Link	77
6.3.4. Form	78
6.3.5. Colori e contrasti	78
7. Conclusioni	81
7.1. Consuntivo	81
7.2. Raggiungimento degli obiettivi	81
7.3. Stato dei requisiti	82
7.4. Conoscenze acquisite	85
7.5. Valutazione personale	85
Glossario	XI
Acronimi	XII
Bibliografia	XV

Elenco delle figure

Figura 1: Logo dell'azienda Girolibero srl	1
Figura 2: Logo del blog happytobehere	2
Figura 3: Architettura del progetto	3
Figura 4: Diagramma dei casi d'uso UC1 - 3	11
Figura 5: Diagramma dei sottocasi di UC1	11
Figura 6: Diagramma dei sottocasi di UC2	13
Figura 7: Diagramma dei sottocasi di UC2.1	13
Figura 8: Diagramma dei casi d'uso UC4 - 6	14
Figura 9: Diagramma dei casi d'uso UC7 - 12	16
Figura 10: Diagramma dei sottocasi di UC9	17
Figura 11: Diagramma dei sottocasi di UC9.1	18
Figura 12: Diagramma dei sottocasi di UC9.1.1	18
Figura 13: Diagramma dei casi d'uso UC13 - 14	21
Figura 14: Diagramma del caso d'uso UC15	22
Figura 15: Diagramma dei sottocasi di UC15	23
Figura 16: Diagramma dei sottocasi di UC15.1	23
Figura 17: Diagramma dei casi d'uso UC16 - 23	24
Figura 18: Diagramma dei sottocasi di UC16	25
Figura 19: Diagramma dei sottocasi di UC16.1	25
Figura 20: Diagramma dei sottocasi di UC22	30
Figura 21: Diagramma dei casi d'uso UC24 - 25	33
Figura 22: Diagramma dei sottocasi di UC24	34
Figura 23: Logo di WordPress	48
Figura 24: Logo di PHP	48
Figura 25: Logo di MySQL	48
Figura 26: Logo di Twig	49
Figura 27: Logo di Timber	49
Figura 28: Logo di Sass	50
Figura 29: Logo di Bootstrap	50
Figura 30: Logo di jQuery	50
Figura 31: Logo di Node.js	51
Figura 32: Logo di React	51
Figura 33: Logo di Cloudinary	51
Figura 34: Logo di MapBox	52
Figura 35: Menu desktop del blog	56
Figura 36: Menu mobile del blog	57
Figura 37: Visualizzazione di una lista di articoli	58
Figura 38: Pulsante utilizzato nella lista di articoli	60
Figura 39: Icona di caricamento per la lista di articoli	60
Figura 40: Header di un articolo	60
Figura 41: Visualizzazione descrizione dell'autore nell'articolo	61

Figura 42: Inserimento descrizione dell'autore dall'editor	62
Figura 43: Indice dei contenuti di un articolo del blog	63
Figura 44: Caricamento di un media su Cloudinary dall'editor Gutenberg .	64
Figura 45: Il form di iscrizione alla newsletter del blog	66
Figura 46: Inserimento coordinate di un articolo nell'editor Gutenberg	67
Figura 47: Punto geografico personalizzato	67
Figura 48: Cluster di punti geografici della mappa	68
Figura 49: Popup di un elemento selezionato nella mappa	68
Figura 50: Visualizzazione della mappa degli articoli	69
Figura 51: Il blocco del carosello viaggi nell'editor Gutenberg	71
Figura 52: Il carosello viaggi nel corpo di un articolo	72

Elenco delle tabelle

Tabella 1: Obiettivi fissati per lo stage.	3
Tabella 2: Pianificazione del lavoro	3
Tabella 3: Requisiti di funzionalità	37
Tabella 4: Requisiti di qualità	44
Tabella 5: Requisiti di vincolo	45
Tabella 6: Riepilogo dei requisiti	46
Tabella 7: Contrasti dei colori dell'interfaccia grafica del blog	79
Tabella 8: Raggiungimento obiettivi fissati per lo stage.	81
Tabella 9: Stato finale dei requisiti	82
Tabella 10: Riepilogo finale dei requisiti raggiunti	84

Capitolo 1.

Introduzione

Il capitolo introduttivo presenta l'azienda ospitante, lo scopo dello stage e gli obiettivi progettuali.

1.1. L'azienda

L'azienda Girolibero è attiva come operatore turistico dal 1998. Ha sede a Vicenza, e si è affermata negli ultimi anni a livello internazionale come punto di riferimento nel settore delle vacanze attive e sostenibili, alcune delle quali in bicicletta.

Girolibero offre itinerari completamente organizzati che includono alloggio, trasporto bagagli tra le tappe e un accompagnatore di fiducia, a cui si aggiunge l'assistenza attiva 7 giorni su 7. Il cliente può richiedere la fornitura della bicicletta e scegliere tra touring bike e e-bike.

All'interno del loro sito principale[1] vengono aggiornati costantemente i numerosi viaggi disponibili, con la possibilità di effettuare rapidamente dei preventivi online. Attualmente, la manutenzione e lo sviluppo di questo sito è affidata a un team *IT (Information Technology)*_[g] interno, che lavora a stretto contatto con il reparto di marketing.

Nella sede di Vicenza, luogo in cui è stato svolto lo stage, sono presenti gli uffici manageriali, booking, marketing, *IT* e un'officina e deposito di biciclette impiegate nei tour.



Figura 1: Logo dell'azienda Girolibero srl

1.1.1. Il team di lavoro

Lo stage è stato svolto affiancando le figure di sviluppo web e marketing dell'azienda. Il team di lavoro era composto dai seguenti professionisti:

- Nicola Cecchetto - Tutor - Sviluppatore web;
- Elena Riatti - Tutor - Responsabile marketing;
- Ilaria Cubalchini - Componente marketing;

1.1. L'AZIENDA

- Simone Fontana - Sviluppatore web;
- Antonio Giuliani - Sviluppatore web;
- Marco Matteazzi - Sviluppatore web;
- Caterina Romio - Project manager e *UX (User Experience)*_[g] designer;
- Claudia Zigliotto - Componente marketing.

Durante lo stage, la principale figura a cui ho fatto riferimento è stata quella di Nicola Cecchetto, che ha supervisionato il mio progetto.

1.2. La proposta di stage

Happy to be here^[2], o happytobehere, è il blog di Girolibero dove vengono pubblicati con cadenza settimanale articoli relativi ai viaggi organizzati, approfondimenti sulle mete e curiosità sul panorama ciclistico.



Figura 2: Logo del blog happytobehere

Il blog, pur essendo ancora in uso, utilizza tecnologie software ormai superate, come funzioni deprecate, che non ricevono più supporto e sono meno performanti rispetto quelle attuali.

In accordo con il team di marketing e il team *IT*, è stato individuato come obiettivo prioritario l'aggiornamento del blog al fine di acquisire maggiore traffico di utenza sul sito principale dell'azienda. L'idea portante è stata quella di progettare e strutturare nuovamente l'intera architettura mantenendo il precedente database, secondo lo schema mostrato nella **Figura 3**.

Il *backend*_[g] del blog continuerà ad essere gestito utilizzando il *CMS (Content Management System)*_[g] WordPress e verrà integrato e connesso ai dati dei viaggi provenienti da girolibero.it. A questo scopo verrà predisposto un *blocco*_[g] apposito personalizzato, contenente i dati dei viaggi organizzati, che un autore potrà inserire nel corpo dei suoi articoli.

Anche l'archivio cloud Cloudinary, utilizzato per gestire la consegna e il recupero dei media sul sito principale, entrerà in uso nel blog.

Sul lato *frontend*_[g], lo stile subirà un completo restyling: seguendo le indicazioni dell'*UX* designer verrà cambiata completamente e modernizzata l'interfaccia grafica, che appare visivamente datata, al fine di richiamare il più possibile lo stile che contraddistingue il marchio di Girolibero e del sito principale.

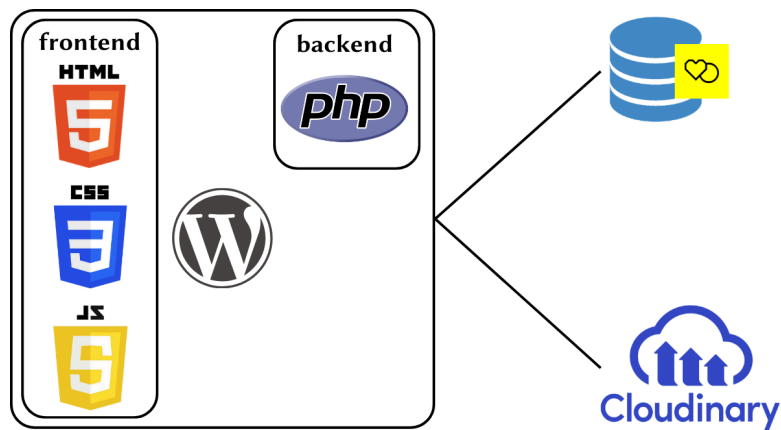


Figura 3: Architettura del progetto

1.3. Obiettivi

Gli obiettivi del tirocinio sono elencati nella [Tabella 1](#) e identificati da un codice secondo la seguente notazione:

- OB per i requisiti obbligatori, vincolanti in quanto obiettivo primario richiesto dall'azienda committente;
- DE per i requisiti desiderabili, non vincolanti o strettamente necessari, ma dal riconoscibile valore aggiunto;
- FA per i requisiti facoltativi, rappresentanti valore aggiunto non strettamente competitivo.

Le sigle precedentemente indicate saranno seguite da un numero, identificativo del requisito.

Codice	Descrizione
OB1	Realizzazione backend e frontend
OB2	Collegamento dei dati provenienti dal sito principale
OB3	Integrazione del blog con Cloudinary
DE1	Realizzazione form di iscrizione alla newsletter _[g]
DE2	Tracciamenti Google Analytics
FA1	Analisi accessibilità del blog ed eventuale adeguamento

Tabella 1: Obiettivi fissati per lo stage.

La pianificazione del lavoro iniziale è mostrata nella [Tabella 2](#).

1.3. OBIETTIVI

Fase introduttiva: 60 ore	
20h	Introduzione agli ambienti di sviluppo, <i>framework</i> _[9] e linguaggi utilizzati
20h	Introduzione a Wordpress e al <i>backend</i> dell'attuale blog Happy to be here
10h	Introduzione alla console Cloudinary, <i>DAM (Digital Asset Management)</i> _[9] utilizzato per l'archiviazione, gestione e consegna al sito delle immagini
20h	Definizione dei requisiti dell'azienda
Fase operativa: 200 ore	
40h	Progettazione
40h	Sviluppo backend
40h	Sviluppo <i>frontend</i>
40h	Collegamento dei dati provenienti da girolibero.it e predisposizione all'uso per gli autori del blog
Fase di verifica: 50 ore	
50h	Verifica, correzione di bug e di interfaccia, ottimizzazione del sistema di upload

Tabella 2: Pianificazione del lavoro

Gli esiti dei traguardi sono riportati nella sezione **7.2. Raggiungimento degli obiettivi**; solo gli obiettivi DE2 e FA1 non sono stati raggiunti.

L'attività di stage ha seguito la pianificazione stabilita, tuttavia la fase di verifica conclusiva è stata dedicata esclusivamente all'analisi dell'accessibilità del blog e alla correzione degli articoli presenti.

1.4. Organizzazione del testo

Vengono di seguito presentati in sintesi i capitoli del documento.

- **Il primo capitolo** illustra gli strumenti e le metodologie di lavoro utilizzati durante il periodo di stage in azienda;
- **Il secondo capitolo** riporta lo studio dei casi d'uso e l'analisi dei requisiti del progetto;
- **Il terzo capitolo** introduce le tecnologie utilizzate durante lo sviluppo;
- **Il quarto capitolo** riporta lo sviluppo e l'implementazione del progetto;
- **Il quinto capitolo** riepiloga gli interventi di revisione eseguiti al termine dello sviluppo;

1.4. ORGANIZZAZIONE DEL TESTO

- **Il sesto capitolo** riporta gli esiti degli obiettivi e dei requisiti di stage.

In fondo al documento sono presenti due sezioni esterne ai capitoli:

- **Il Glossario** definisce gli acronimi, le abbreviazioni e i termini ambigui o di uso non comune menzionati all'interno del documento;
- **La Bibliografia** riporta le fonti materiali e digitali consultate per la stesura di questo documento.

1.4.1. Convenzioni stilistiche

Vengono di seguito elencate le convenzioni stilistiche adottate per la stesura del testo:

- i collegamenti a capitoli, sezioni e paragrafi del documento sono indicati come segue: **Capitolo**;
- i link a pagine esterne del documento sono segnalati con questa modalità:
URL: www.esempio.it;
- per la prima occorrenza dei termini riportati nel **Glossario** viene utilizzata la seguente nomenclatura: *termine*_[g] ;
- tutti i termini già menzionati presenti nel glossario presenteranno lo seguente stile: *termine*.

Capitolo 2.

Strumenti e metodologie di lavoro

Questo capitolo presenta gli strumenti e le modalità di lavoro e comunicazione adottate per collaborare con il team.

Lo stage si è svolto interamente in presenza, all'interno dell'ufficio marketing aziendale.

Ad eccezione del tutor Nicola Cecchetto, presente tutti i giorni in ufficio, il team **IT** presenziava fisicamente solo il martedì di ogni settimana, così come l'**UX** designer.

Il primo giorno di stage sono stati delineati con i tutor gli strumenti di comunicazione con il team, che vengono di seguito elencati.

Slack

Slack¹ è una piattaforma di messaggistica istantanea che ha lo scopo di favorire comunicazioni veloci tra i membri di un team di lavoro. Consente di organizzare le comunicazioni tramite la creazione di canali specifici dedicati ad alcuni membri del team denominate Aree di Lavoro, canali generali o chat private. Offre, inoltre, la possibilità di effettuare incontri online.

Con l'avvio dello stage, sono stato inserito nell'area di lavoro Slack del team di marketing, in cui erano presenti anche i membri del gruppo **IT**. Questa modalità di comunicare è stata usata spesso per risolvere dubbi legati allo sviluppo o richiedere feedback su quanto svolto quando gli sviluppatori non erano presenti fisicamente in ufficio. Tramite Slack sono stati effettuati degli incontri online anche il martedì, per consentire la partecipazione a distanza anche dei componenti eventualmente assenti in sede.

Gmail

Gmail è un servizio di posta elettronica di proprietà di Google.

Nel contesto aziendale è stato utilizzato solo per le comunicazioni importanti di carattere logistico, come eventuali necessità di permessi in ingresso e uscita.

Ogni martedì, i diversi componenti del team si confrontavano sul lavoro svolto e, a seconda delle necessità imposte dalla Responsabile di Marketing, delineavano gli obiettivi della settimana corrente. Durante questa riunione aggiornavo

¹ URL: www.slack.com

il team sullo stato di avanzamento del mio progetto e, nel caso ne avessi avuto bisogno, coglievo questa opportunità per esporre e chiarire eventuali dubbi legati alle tecnologie.

Sempre in questa giornata, con Caterina Romio, UX designer, Elena Riatti, tutor e responsabile marketing e Nicola Cecchetto, tutore e sviluppatore web, facevo il punto della situazione su quanto svolto e programmavo gli obiettivi da raggiungere in vista dell'incontro successivo.

Durante il primo giorno di stage sono stati configurati gli strumenti di lavoro progettuali, che vengono di seguito elencati.

BitBucket

BitBucket² è un servizio di hosting per progetti software che utilizzano sistemi di versionamento Mercurial o Git. È stato utilizzato nel corso dello stage per tutta la durata dello sviluppo del progetto, consentendo anche ai membri del team IT di controllare il suo andamento.

Git

Git³ è un software di versionamento utilizzabile da linea di comando. È ampiamente diffuso tra i servizi di hosting per progetti software presenti sul mercato.

Visual Studio Code

Visual Studio Code⁴ è un editor di codice sorgente, utilizzato durante tutte le fasi del progetto. Grazie all'aggiunta di estensioni per i diversi linguaggi esistenti, facilita la scrittura del codice grazie a suggerimenti e controlli sulla sintassi.

XAMPP

XAMPP⁵ è una piattaforma software multiplatforma formata da server HTTP Apache, DBMS MariaDB e i linguaggi di programmazione PHP e Perl. È stata utilizzata per creare un web hosting locale del blog durante il suo sviluppo.

² URL: www.bitbucket.org

³ URL: www.git-scm.com

⁴ URL: code.visualstudio.com

⁵ URL: www.apachefriends.org

Capitolo 3.

Analisi dei requisiti

Questo capitolo riepiloga i casi d'uso e i requisiti individuati in occasione dell'analisi del prodotto e dei colloqui effettuati con l'azienda.

3.1. Introduzione ai casi d'uso

I casi d'uso che ho individuato sono 25 e sono di seguito descritti seguendo questo modello:

- Attori coinvolti: soggetti che svolgono determinate azioni;
- Descrizione: titolo e introduzione al caso d'uso;
- Precondizioni: stato del sistema prima del caso d'uso;
- Postcondizioni: stato del sistema dopo il caso d'uso;
- Scenario principale: elenco dettagliato delle azioni che l'attore deve svolgere per completare un caso d'uso;
- Estensioni: l'estensione è una relazione tra due casi d'uso in cui il caso d'uso esteso può essere modificato o arricchito dal caso d'uso estensore. L'estensione è rappresentata da una freccia con linea tratteggiata che punta dal caso d'uso estensore al caso d'uso esteso;
- Inclusioni: l'inclusione è una relazione tra due casi d'uso in cui il caso d'uso incluso è eseguito sempre all'interno del caso d'uso che include. L'inclusione è rappresentata da una freccia con linea tratteggiata che punta dal caso d'uso che include al caso d'uso incluso;
- Generalizzazioni: la generalizzazione è una relazione tra due casi d'uso in cui il caso d'uso generalizzato rappresenta una categoria più ampia rispetto al caso d'uso specifico, dove quest'ultimo è un'istanza. La generalizzazione è rappresentata da una freccia con linea continua che punta dal caso d'uso specifico al caso d'uso generalizzato.

3.2. Attori individuati

Ogni caso d'uso è associato a un certo numero di attori, che sono stati individuati durante l'analisi dei requisiti e sono riportati di seguito.

- **Visitatore:** un utente qualsiasi che visita il blog happytobehere. Il visitatore, non necessariamente cliente di girolibero, potrà navigare all'interno del blog per visualizzare gli articoli presenti, ordinati per categorie, tag e autori;

3.2. ATTORI INDIVIDUATI

- **Autore:** un componente del team di marketing dell'azienda Girolibero che scrive un articolo per il blog. Per instradare il traffico del blog verso il sito principale, lo scrittore potrà pubblicizzare i viaggi organizzati di girolibero, inserendo all'interno degli articoli dei blocchi personalizzati.

Oltre agli attori elencati precedentemente, sono stati individuati anche alcuni attori secondari: non sono persone vere e proprie, ma entità di terze parti che interagiscono con il blog.

- **Cloudinary:** gestore di immagini e video in cloud che fornisce la copertina e le immagini di contenuto per i post;
- **API Girolibero:** *API (Application Programming Interface)*^[9] che risponde alle richieste con un file strutturato contenente tutti i dati dei viaggi organizzati presenti sul database;
- **Mailchimp:** servizio email marketing utilizzato ogni volta che un visitatore desidera iscriversi alla [newsletter](#) di Girolibero;
- **MapBox:** fornitore di mappe online personalizzate per siti web.

3.3. WordPress

Prima di procedere con l'analisi dei requisiti, è stato condotto un approfondimento sul precedente [backend](#) del blog per comprendere quali funzionalità fossero già presenti.

Come riportato nella [Tabella 5](#), relativa ai requisiti di vincolo, l'utilizzo di WordPress come **CMS** per questo progetto rappresenta un requisito obbligatorio. Pertanto, all'interno di questo documento, non è stato riportato lo studio delle diverse funzionalità fornite da WordPress e già presenti nel precedente [backend](#) del blog.

Le funzionalità già presenti riguardano esclusivamente gli Autori: sono un esempio la pubblicazione o la rimozione di un articolo o la gestione dei contenuti, come testo, link e immagini nel corpo degli articoli. Si tratta di funzionalità già presenti in diversi CMS come WordPress e non è stato quindi necessario approfondirle.

3.4. Elenco dei casi d'uso

Vengono di seguito casi d'uso i requisiti individuati durante lo studio delle funzionalità del prodotto.

3.4.1. UC1 - Inserimento coppia di coordinate geografiche correlate all'articolo

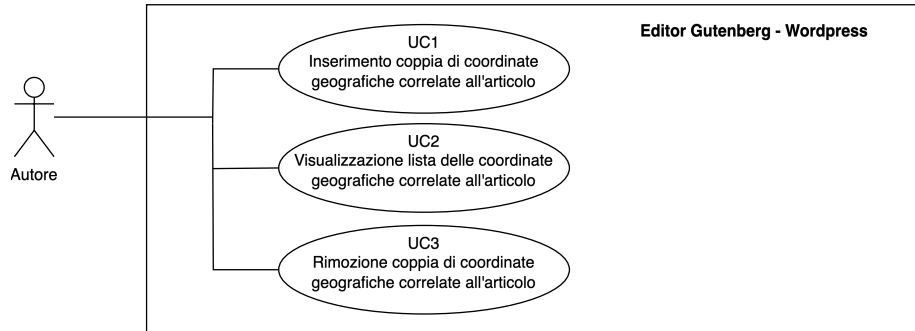


Figura 4: Diagramma dei casi d'uso UC1 - 3

- **Attori principali:** Autore;
- **Descrizione:** Un autore del blog happytobehere desidera aggiungere la coppia di coordinate geografiche di un luogo correlato all'articolo nell'apposito campo metadata all'interno dell'editor;
- **Precondizioni:** La pagina contenente l'editor dell'articolo è stata caricata correttamente sul browser dell'autore;
- **Postcondizioni:** La coppia di coordinate geografiche è stata inserita correttamente all'interno dei *metadati_[g]* dell'articolo;
- **Scenario principale:**
 1. L'autore compila l'apposito campo e inserisce la coppia di coordinate geografiche all'interno dei metadati dell'articolo.

3.4.1.1. UC1.1 - Inserimento latitudine coordinata

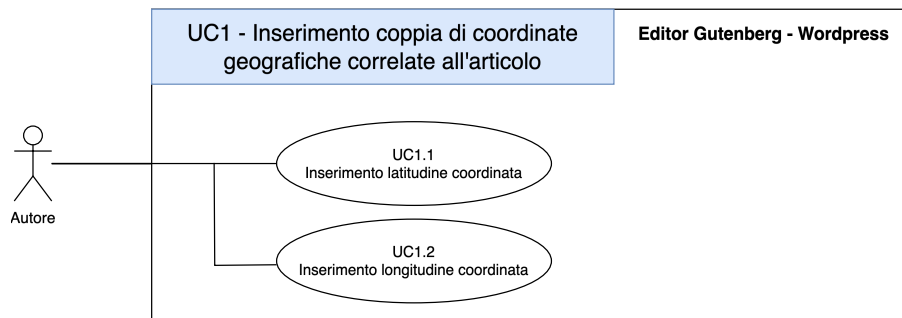


Figura 5: Diagramma dei sottocasi di UC1

- **Attori principali:** Autore;
- **Descrizione:** Un autore del blog happytobehere desidera inserire la latitudine di un luogo correlato all'articolo nell'apposito campo metadata all'interno dell'editor;

3.4. ELENCO DEI CASI D'USO

- **Precondizioni:** La pagina contenente l'editor dell'articolo è stata caricata correttamente sul browser dell'autore;
- **Postcondizioni:** La latitudine del luogo è stata inserita correttamente all'interno dei [metadati](#) dell'articolo;
- **Scenario principale:**
 1. L'autore compila l'apposito campo e la latitudine di un luogo correlato all'articolo all'interno dei metadati dell'articolo.

3.4.1.2. UC1.2 - Inserimento longitudine coordinata

- **Attori principali:** Autore;
- **Descrizione:** Un autore del blog happytobehere desidera inserire la longitudine di un luogo correlato all'articolo nell'apposito campo metadata all'interno dell'editor;
- **Precondizioni:** La pagina contenente l'editor dell'articolo è stata caricata correttamente sul browser dell'autore;
- **Postcondizioni:** La longitudine del luogo è stata inserita correttamente all'interno dei [metadati](#) dell'articolo;
- **Scenario principale:**
 1. L'autore compila l'apposito campo e la longitudine di un luogo correlato all'articolo all'interno dei metadati dell'articolo.

3.4.2. UC2 - Visualizzazione lista delle coordinate geografiche correlate all'articolo

- **Attori principali:** Autore;
- **Descrizione:** Un autore del blog happytobehere desidera visualizzare la lista delle coordinate geografiche dei luoghi correlati all'articolo utilizzando l'editor;
- **Precondizioni:** La pagina contenente l'editor dell'articolo è stata caricata correttamente sul browser dell'autore;
- **Postcondizioni:** La lista delle coordinate geografiche correlate all'articolo viene visualizzata correttamente dall'autore;
- **Scenario principale:**
 1. L'autore visualizza la lista delle coordinate geografiche dei luoghi correlati all'articolo utilizzando l'editor.

3.4.2.1. UC2.1 - Visualizzazione singola coppia di coordinate



Figura 6: Diagramma dei sottocasi di UC2

- **Attori principali:** Autore;
- **Descrizione:** Un autore del blog happytobehere desidera visualizzare un elemento della lista delle coordinate geografiche dei luoghi correlati all'articolo utilizzando l'editor;
- **Precondizioni:** La pagina contenente l'editor dell'articolo è stata caricata correttamente sul browser dell'autore;
- **Postcondizioni:** L'elemento della lista delle coordinate geografiche correlate all'articolo viene visualizzato correttamente dall'autore;
- **Scenario principale:**
 1. L'autore seleziona la coppia di coordinate desiderata presente nell'apposita lista e la visualizza utilizzando l'editor.

3.4.2.2. UC2.1.1 - Visualizzazione latitudine

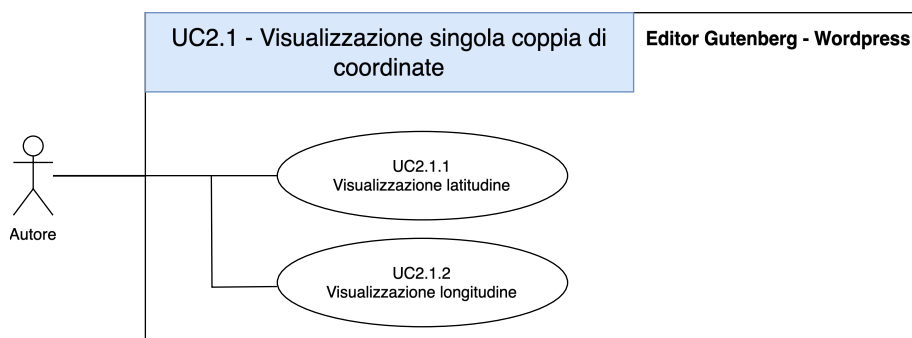


Figura 7: Diagramma dei sottocasi di UC2.1

- **Attori principali:** Autore;
- **Descrizione:** Un autore del blog happytobehere desidera visualizzare la latitudine di un elemento della lista delle coordinate geografiche dei luoghi correlati all'articolo utilizzando l'editor;
- **Precondizioni:** La pagina contenente l'editor dell'articolo è stata caricata correttamente sul browser dell'autore;
- **Postcondizioni:** La latitudine dell'elemento della lista delle coordinate geografiche correlate all'articolo viene visualizzata correttamente dall'autore;

3.4. ELENCO DEI CASI D'USO

- **Scenario principale:**
 1. L'autore visualizza la latitudine dell'elemento desiderato presente nell'apposita lista utilizzando l'editor.

3.4.2.3. UC2.1.2 - Visualizzazione longitudine

- **Attori principali:** Autore;
- **Descrizione:** Un autore del blog happytobehere desidera visualizzare la longitudine di un elemento della lista delle coordinate geografiche dei luoghi correlati all'articolo utilizzando l'editor;
- **Precondizioni:** La pagina contenente l'editor dell'articolo è stata caricata correttamente sul browser dell'autore;
- **Postcondizioni:** La longitudine dell'elemento della lista delle coordinate geografiche correlate all'articolo viene visualizzata correttamente dall'autore;
- **Scenario principale:**
 1. L'autore visualizza la longitudine dell'elemento desiderato presente nell'apposita lista utilizzando l'editor.

3.4.3. UC3 - Rimozione coppia di coordinate geografiche correlate all'articolo

- **Attori principali:** Autore;
- **Descrizione:** Un autore del blog happytobehere desidera rimuovere la coppia di coordinate geografiche di un luogo correlato all'articolo dai suoi [metadati](#) utilizzando l'editor;
- **Precondizioni:** La pagina contenente l'editor dell'articolo è stata caricata correttamente sul browser dell'autore ed è presente almeno una coppia di coordinate geografiche nei metadati dell'articolo;
- **Postcondizioni:** La coppia di coordinate geografiche è stata rimossa correttamente dai metadati dell'articolo;
- **Scenario principale:**
 1. L'autore rimuove dai metadati dell'articolo la coppia di coordinate geografiche desiderata utilizzando l'editor.

3.4.4. UC4 - Inserimento blocco "indice"

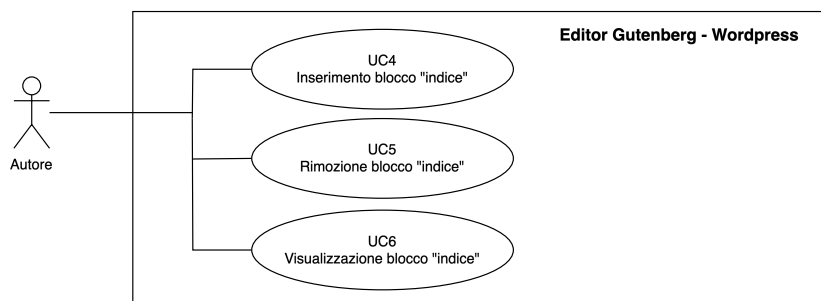


Figura 8: Diagramma dei casi d'uso UC4 - 6

3.4. ELENCO DEI CASI D'USO

- **Attori principali:** Autore;
- **Descrizione:** Un autore del blog happytobehere desidera inserire all'interno del corpo di un articolo un **blocco** contenente l'indice dei paragrafi;
- **Precondizioni:** La pagina contenente l'editor dell'articolo è stata caricata correttamente sul browser dell'autore e il blocco è configurato correttamente all'interno dell'editor Gutenberg di WordPress;
- **Postcondizioni:** Il blocco viene inserito correttamente all'interno del corpo dell'articolo dall'autore;
- **Scenario principale:**
 1. L'autore seleziona il blocco indice dalla lista dei blocchi disponibili nell'editor e lo posiziona all'interno dell'articolo.

3.4.5. UC5 - Rimozione blocco "indice"

- **Attori principali:** Autore;
- **Descrizione:** Un autore del blog happytobehere desidera rimuovere dal corpo di un articolo il **blocco** contenente l'indice dei paragrafi;
- **Precondizioni:** la pagina contenente l'editor dell'articolo è stata caricata correttamente sul browser dell'autore e il blocco è configurato correttamente all'interno dell'editor Gutenberg di WordPress. E' presente un blocco "indice" nel corpo dell'articolo dall'autore;
- **Postcondizioni:** Il blocco viene rimosso correttamente dal corpo dell'articolo;
- **Scenario principale:**
 1. L'autore seleziona il blocco indice nel corpo dell'articolo e lo rimuove usando l'editor.

3.4.6. UC6 - Visualizzazione blocco "indice"

- **Attori principali:** Autore;
- **Descrizione:** Un autore del blog happytobehere desidera visualizzare un **blocco** contenente l'indice dei paragrafi presente nel corpo di un articolo;
- **Precondizioni:** la pagina contenente l'editor dell'articolo è stata caricata correttamente sul browser dell'autore e il blocco è configurato correttamente all'interno dell'editor Gutenberg di WordPress;
- **Postcondizioni:** Il blocco viene visualizzato correttamente nel corpo dell'articolo dall'autore;
- **Scenario principale:**
 1. L'autore visualizza il blocco indice dal corpo dell'articolo usando l'editor.

3.4. ELENCO DEI CASI D'USO

3.4.7. UC7 - Inserimento blocco "carosello viaggi"

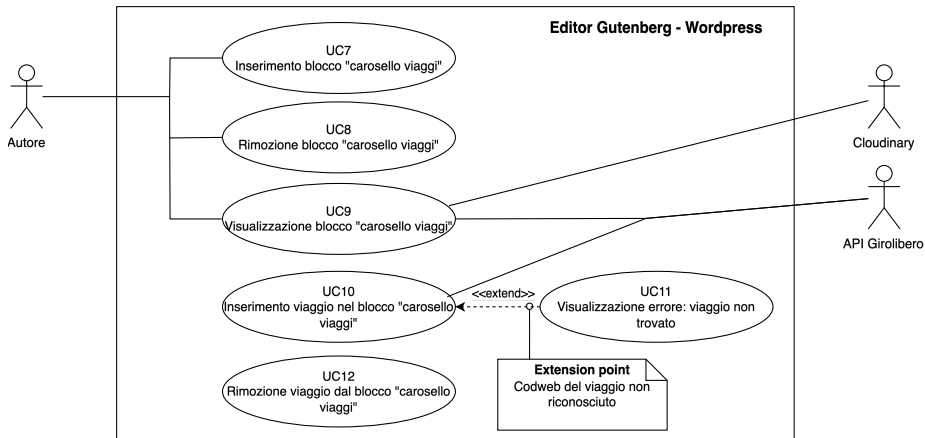


Figura 9: Diagramma dei casi d'uso UC7 - 12

- **Attori principali:** Autore;
- **Descrizione:** Un autore del blog happytobehere desidera inserire all'interno di un articolo un **blocco** contenente un **carosello**_[g] di viaggi organizzati da Girolibero;
- **Precondizioni:** Il plug-in contenente il blocco è configurato correttamente all'interno di WordPress;
- **Postcondizioni:** Il blocco viene inserito correttamente all'interno del corpo dell'articolo dall'autore;
- **Scenario principale:**
 1. L'autore seleziona il blocco contenente il carosello viaggi dalla lista dei blocchi disponibili nell'editor e lo posiziona all'interno dell'articolo.

3.4.8. UC8 - Rimozione blocco "carosello viaggi"

- **Attori principali:** Autore;
- **Descrizione:** Un autore del blog happytobehere desidera rimuovere dal corpo di un articolo un **blocco** contenente un **carosello** di viaggi organizzati da Girolibero;
- **Precondizioni:** Il plug-in contenente il blocco è configurato correttamente all'interno di WordPress ed è presente almeno un blocco "carosello viaggi" nel corpo dell'articolo;
- **Postcondizioni:** Il blocco viene rimosso correttamente dal corpo dell'articolo dall'autore;
- **Scenario principale:**
 1. L'autore rimuove il blocco contenente il carosello viaggi dal corpo dell'articolo usando l'editor.

3.4.9. UC9 - Visualizzazione blocco “carosello viaggi”

- **Attori principali:** Autore, [API Girolibero](#), Cloudinary;
- **Descrizione:** Un autore del blog happytobehere desidera visualizzare un [blocco](#) contenente un [carosello](#) di viaggi organizzati da Girolibero presente nel corpo di un articolo;
- **Precondizioni:** Il plug-in contenente il blocco è configurato correttamente all'interno di WordPress ed è presente almeno un blocco “carosello viaggi” nel corpo dell'articolo;
- **Postcondizioni:** Il blocco viene visualizzato correttamente nel corpo dell'articolo dall'autore;
- **Scenario principale:**
 1. L'autore visualizza il blocco contenente il carosello viaggi, presente nel corpo dell'articolo, usando l'editor.

3.4.9.1. UC9.1 - Visualizzazione lista viaggi presenti nel “carosello viaggi”

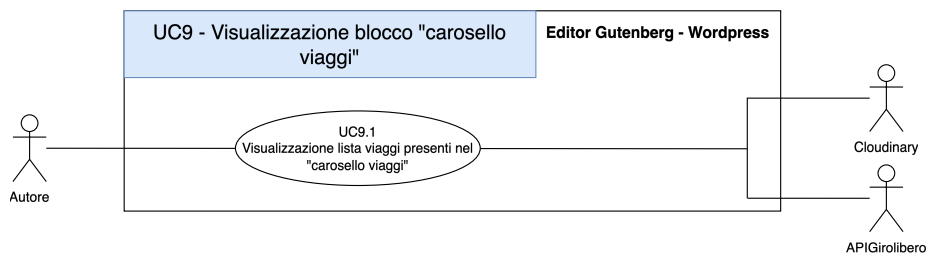


Figura 10: Diagramma dei sottocasi di UC9

- **Attori principali:** Autore, [API Girolibero](#), Cloudinary;
- **Descrizione:** Un autore del blog happytobehere desidera visualizzare la lista dei viaggi inseriti all'interno del blocco del carosello viaggi;
- **Precondizioni:** Il plug-in contenente il blocco è configurato correttamente all'interno di WordPress e il blocco è inserito all'interno del corpo dell'articolo;
- **Postcondizioni:** la lista dei viaggi presenti all'interno del carosello viene visualizzata correttamente all'interno del blocco dall'autore;
- **Scenario principale:**
 1. Il blocco recupera le informazioni della lista dei viaggi del carosello tramite l'API Girolibero;
 2. L'autore visualizza la lista dei viaggi inseriti all'interno del blocco, presente nel corpo dell'articolo, usando l'editor.

3.4. ELENCO DEI CASI D'USO

3.4.9.1.1. UC9.1.1 - Visualizzazione singolo viaggio

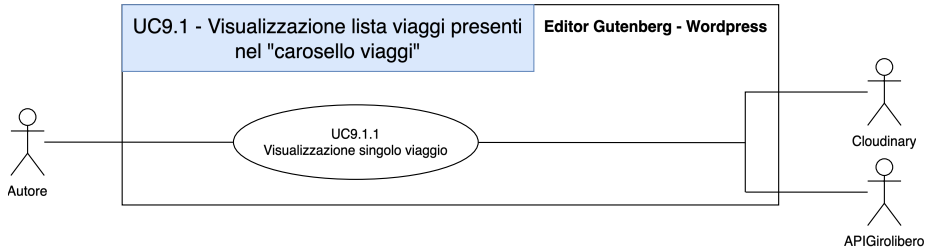


Figura 11: Diagramma dei sottocasi di UC9.1

- **Attori principali:** Autore, API Girolibero;
- **Descrizione:** Un autore del blog happytobehere desidera visualizzare un elemento della lista dei viaggi presenti all'interno del carousel contenuto nel blocco nel corpo dell'articolo;
- **Precondizioni:** Il plug-in contenente il blocco è configurato correttamente all'interno di WordPress e il blocco è inserito all'interno del corpo dell'articolo;
- **Postcondizioni:** Il singolo viaggio viene visualizzato correttamente all'interno del blocco;
- **Scenario principale:**
 1. Il blocco recupera le informazioni di un singolo viaggio tramite l'API Girolibero;
 2. L'autore visualizza il singolo viaggio all'interno del blocco, presente nel corpo dell'articolo, usando l'editor.

3.4.9.1.2. UC 9.1.1.1 - Visualizzazione titolo

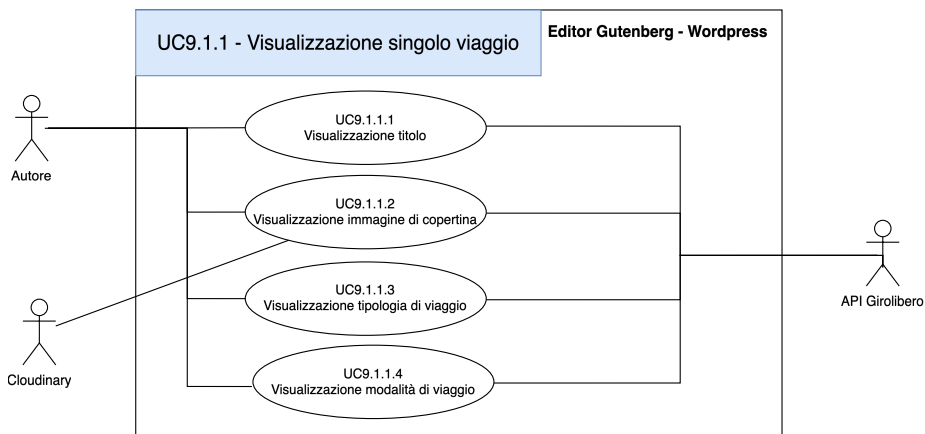


Figura 12: Diagramma dei sottocasi di UC9.1.1

- **Attori principali:** Autore, API Girolibero;
- **Descrizione:** Un autore del blog happytobehere desidera visualizzare il titolo di un viaggio della lista presente all'interno del blocco carousel viaggi;

3.4. ELENCO DEI CASI D'USO

- **Precondizioni:** Il plug-in contenente il blocco è configurato correttamente all'interno di WordPress e il blocco è inserito all'interno del corpo dell'articolo;
- **Postcondizioni:** il titolo del viaggio viene visualizzato correttamente all'interno del blocco;
- **Scenario principale:**
 1. Il blocco recupera il titolo del viaggio tramite l'API Girolibero;
 2. L'autore visualizza il titolo del viaggio all'interno del blocco, presente nel corpo dell'articolo, usando l'editor.

3.4.9.1.3. UC 9.1.1.2 - Visualizzazione immagine di copertina

- **Attori principali:** Autore, API Girolibero, Cloudinary;
- **Descrizione:** Un autore del blog happytobehere desidera visualizzare l'immagine di copertina di un viaggio della lista presente all'interno del blocco carosello viaggi;
- **Precondizioni:** Il plug-in contenente il blocco è configurato correttamente all'interno di WordPress e il blocco è inserito all'interno del corpo dell'articolo;
- **Postcondizioni:** l'immagine di copertina del viaggio viene visualizzato correttamente all'interno del blocco;
- **Scenario principale:**
 1. Il blocco recupera l'url dell'immagine tramite l'API Girolibero;
 2. L'immagine viene recuperata dal cloud Cloudinary e inserita all'interno del blocco;
 3. L'autore visualizza l'immagine di copertina del viaggio all'interno del blocco, presente nel corpo dell'articolo, usando l'editor.

3.4.9.1.4. UC 9.1.1.3 - Visualizzazione tipologia di viaggio

- **Attori principali:** Autore, API Girolibero;
- **Descrizione:** Un autore del blog happytobehere desidera visualizzare la *tipologia_[g]* (bici, trekking..) di un viaggio della lista presente all'interno del blocco carosello viaggi.
- **Precondizioni:** Il plug-in contenente il blocco è configurato correttamente all'interno di WordPress e il blocco è inserito all'interno del corpo dell'articolo;
- **Postcondizioni:** la tipologia del viaggio viene visualizzata correttamente all'interno del blocco;
- **Scenario principale:**
 1. Il blocco recupera la tipologia del viaggio tramite l'API Girolibero;
 2. L'autore visualizza la tipologia del viaggio all'interno del blocco, presente nel corpo dell'articolo, usando l'editor.

3.4. ELENCO DEI CASI D'USO

3.4.9.1.5. UC 9.1.1.4 - Visualizzazione modalità di viaggio

- **Attori principali:** Autore, API Girolibero;
- **Descrizione:** Un autore del blog happytobehere desidera visualizzare la *modalità_[g]* (indipendente, in gruppo..) di un viaggio della lista presente all'interno del blocco carosello viaggi.
- **Precondizioni:** Il plug-in contenente il blocco è configurato correttamente all'interno di WordPress e il blocco è inserito all'interno del corpo dell'articolo;
- **Postcondizioni:** la modalità del viaggio viene visualizzata correttamente all'interno del blocco;
- **Scenario principale:**
 1. Il blocco recupera la modalità del viaggio tramite l'API Girolibero;
 2. L'autore visualizza la modalità del viaggio all'interno del blocco, presente nel corpo dell'articolo, usando l'editor.

3.4.10. UC10 - Inserimento viaggio nel blocco "carosello viaggi"

- **Attori principali:** Autore, API Girolibero;
- **Descrizione:** Un autore del blog happytobehere desidera inserire i dati di un viaggio all'interno del blocco contenente il carosello viaggi, presente nel corpo di un articolo;
- **Precondizioni:** Il plug-in contenente il blocco è configurato correttamente all'interno di WordPress e il blocco è inserito all'interno del corpo dell'articolo;
- **Postcondizioni:** Il viaggio selezionato viene inserito correttamente all'interno del carosello viaggi;
- **Scenario principale:**
 1. L'autore inserisce il *codweb_[g]* identificativo del viaggio desiderato tramite l'editor;
 2. Il blocco verifica che il codweb inserito sia corretto tramite l'API Girolibero;
 3. Il viaggio viene inserito all'interno del blocco, presente nel corpo dell'articolo, usando l'editor.
- **Estensioni:**
 - Errore: codweb non presente.

3.4.11. UC11 - Visualizzazione errore: codweb non presente

- **Attori principali:** Autore;
- **Descrizione:** un autore del blog happytobehere ha tentato di inserire il *codweb* di un viaggio non esistente all'interno del blocco contenente il carosello viaggi.

- **Precondizioni:** Il plug-in contenente il blocco è configurato correttamente all'interno di WordPress e il blocco è inserito all'interno del corpo dell'articolo. L'autore ha tentato di inserire un codweb non presente all'interno dell'API Girolibero.
- **Postcondizioni:** il codweb errato fornito non viene inserito all'interno del carosello viaggi e l'autore visualizza un messaggio di errore;
- **Scenario principale:**
 1. L'autore visualizza un messaggio di errore, in quanto il codweb inserito nel blocco non ha trovato corrispondenza con nessun viaggio.

3.4.12. UC12 - Rimozione viaggio dal blocco "carosello viaggi"

- **Attori principali:** Autore;
- **Descrizione:** Un autore del blog happytobehere ha l'intenzione di rimuovere un viaggio dal **carosello** viaggi contenuto in un **blocco** all'interno di un articolo;
- **Precondizioni:** Il plug-in contenente il blocco è configurato correttamente all'interno di WordPress. Il blocco è inserito nel corpo dell'articolo ed è presente al suo interno almeno un viaggio;
- **Postcondizioni:** Il viaggio selezionato viene correttamente rimosso dal carosello viaggi;
- **Scenario principale:**
 1. L'autore cancella il **codweb** identificativo del viaggio da rimuovere dal blocco;
 2. Il viaggio viene rimosso correttamente dal blocco.

3.4.13. UC13 - Inserimento descrizione autore in fondo all'articolo

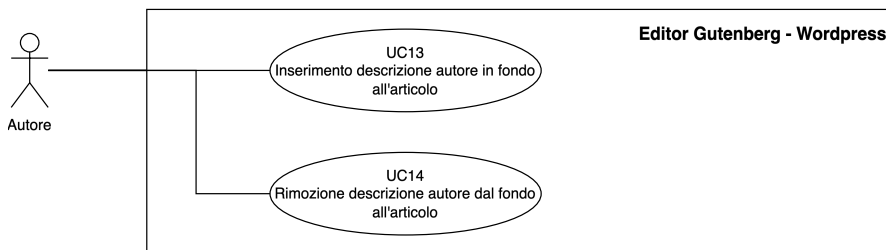


Figura 13: Diagramma dei casi d'uso UC13 - 14

- **Attori principali:** Autore;
- **Descrizione:** Un autore del blog happytobehere ha l'intenzione di inserire la sua descrizione biografica al termine di un articolo;
- **Precondizioni:** la pagina web contenente l'editor dell'articolo è stata caricata correttamente sul browser e la descrizione dell'autore non è vuota;
- **Postcondizioni:** la descrizione dell'autore viene inserita correttamente alla fine dell'articolo;

3.4. ELENCO DEI CASI D'USO

- **Scenario principale:**

1. L'autore sceglie di inserire la sua descrizione alla fine dell'articolo usando l'editor.

3.4.14. UC14 - Rimozione descrizione autore dal fondo dell'articolo

- **Attori principali:** Autore;

- **Descrizione:** Un autore del blog happytobehere ha l'intenzione di rimuovere la sua descrizione presente al termine di un articolo;

- **Precondizioni:** la pagina web contenente l'editor dell'articolo è stata caricata correttamente sul browser e al termine dell'articolo è presente la descrizione dell'autore;

- **Postcondizioni:** la descrizione dell'autore presente alla fine dell'articolo viene rimossa correttamente;

- **Scenario principale:**

1. L'autore rimuove la sua descrizione presente alla fine dell'articolo usando l'editor.

3.4.15. UC15 - Visualizzazione mappa geografica degli articoli

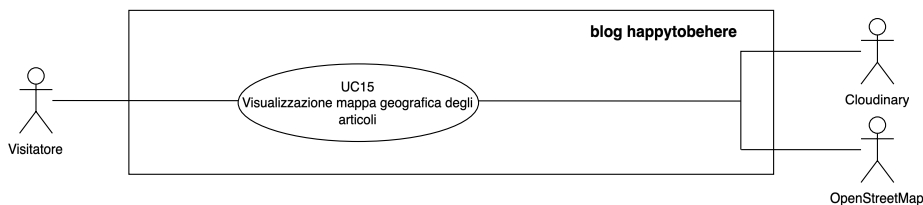


Figura 14: Diagramma del caso d'uso UC15

- **Attori principali:** Visitatore, Cloudinary, MapBox;

- **Descrizione:** un visitatore desidera visualizzare la mappa geografica che localizza gli articoli del blog che sono associati a coordinate geografiche;

- **Precondizioni:** una qualsiasi pagina del blog è stata caricata correttamente sul browser del visitatore;

- **Postcondizioni:** la mappa è stata visualizzata correttamente dal visitatore;

- **Scenario principale:**

1. Il visitatore seleziona l'icona sulla pagina per visualizzare la mappa geografica che localizza gli articoli del blog;
2. MapBox carica i vari segnaposto degli articoli sulla mappa;
3. il visitatore visualizza la mappa.

3.4.15.1. UC15.1 Visualizzazione elemento della mappa

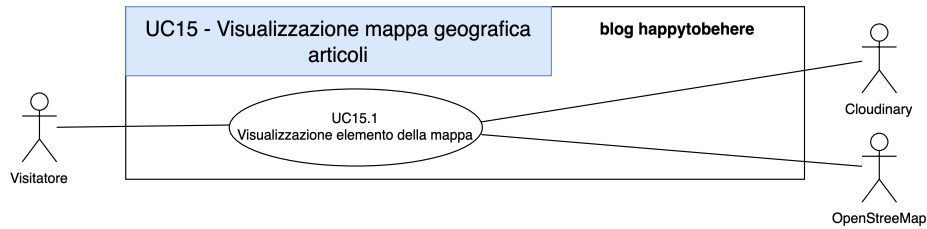


Figura 15: Diagramma dei sottocasi di UC15

- **Attori principali:** Visitatore, Cloudinary, MapBox;
- **Descrizione:** un visitatore desidera visualizzare un singolo segnaposto nella mappa geografica che localizza gli articoli del blog;
- **Precondizioni:** la mappa degli articoli è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il singolo elemento della mappa è stato visualizzato correttamente dal visitatore;
- **Scenario principale:**
 1. Il visitatore seleziona il segnaposto desiderato all'interno della mappa e ne visualizza le informazioni.

3.4.15.1.1. UC15.1.1 Visualizzazione titolo

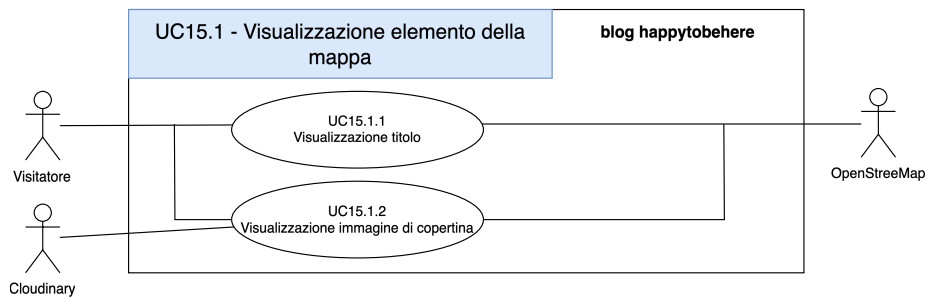


Figura 16: Diagramma dei sottocasi di UC15.1

- **Attori principali:** Visitatore, MapBox;
- **Descrizione:** un visitatore desidera visualizzare il titolo di un singolo segnaposto nella mappa geografica che localizza gli articoli del blog;
- **Precondizioni:** il segnaposto della mappa selezionato è stato caricato correttamente sul browser del visitatore;
- **Postcondizioni:** il titolo dell'elemento della mappa è stato visualizzato correttamente dal visitatore;
- **Scenario principale:**
 1. Il visitatore visualizza il titolo del segnaposto selezionato.

3.4. ELENCO DEI CASI D'USO

3.4.15.1.2. UC15.1.2 Visualizzazione immagine di copertina

- **Attori principali:** Visitatore, Cloudinary, MapBox;
- **Descrizione:** un visitatore desidera visualizzare l'immagine di copertina di un singolo segnaposto nella mappa geografica che localizza gli articoli del blog;
- **Precondizioni:** il segnaposto della mappa selezionato è stato caricato correttamente sul browser del visitatore;
- **Postcondizioni:** l'immagine di copertina dell'elemento della mappa è stato visualizzato correttamente dal visitatore;
- **Scenario principale:**
 1. L'immagine di copertina del segnaposto viene recuperata utilizzando il cloud Cloudinary;
 2. Il visitatore visualizza l'immagine di copertina del segnaposto selezionato.

3.4.16. UC16 - Visualizzazione lista articoli

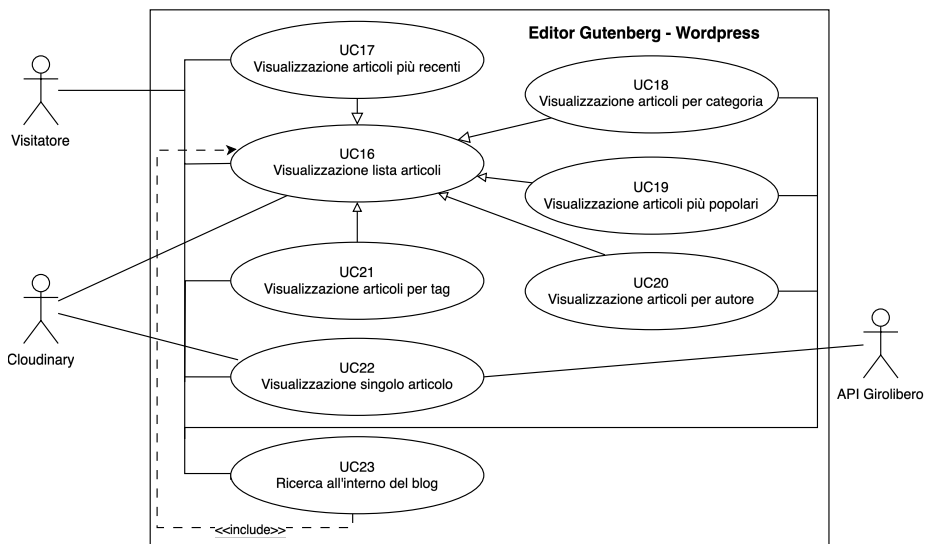


Figura 17: Diagramma dei casi d'uso UC16 - 23

- **Attori principali:** Visitatore, Cloudinary;
- **Descrizione:** un visitatore desidera visualizzare una lista di articoli presenti sul blog;
- **Precondizioni:** la pagina del blog contenente la lista è stata caricata correttamente nel browser del visitatore;
- **Postcondizioni:** il visitatore ha visualizzato correttamente la lista di articoli;
- **Scenario principale:**
 1. Il visitatore visualizza la pagina contenente una lista di articoli del blog;

- **Generalizzazioni:**
 - Visualizzazione articoli più recenti;
 - Visualizzazione articoli per categoria;
 - Visualizzazione articoli più popolari;
 - Visualizzazione articoli per autore;
 - Visualizzazione articoli per tag.

3.4.16.1. UC16.1 - Visualizzazione elemento della lista

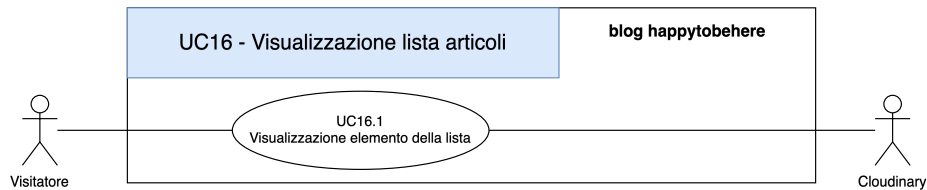


Figura 18: Diagramma dei sottocasi di UC16

- **Attori principali:** Visitatore, Cloudinary;
- **Descrizione:** un visitatore desidera visualizzare un elemento di una lista di articoli all'interno del blog. Questo elemento è un'anteprima dell'articolo e ne riporta le informazioni principali;
- **Precondizioni:** la pagina contenente la lista è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha visualizzato correttamente l'elemento di una lista di articoli all'interno del blog;
- **Scenario principale:**
 1. il visitatore visualizza correttamente un elemento di una lista di articoli del blog.

3.4.16.1.1. UC16.1.1 - Visualizzazione titolo elemento

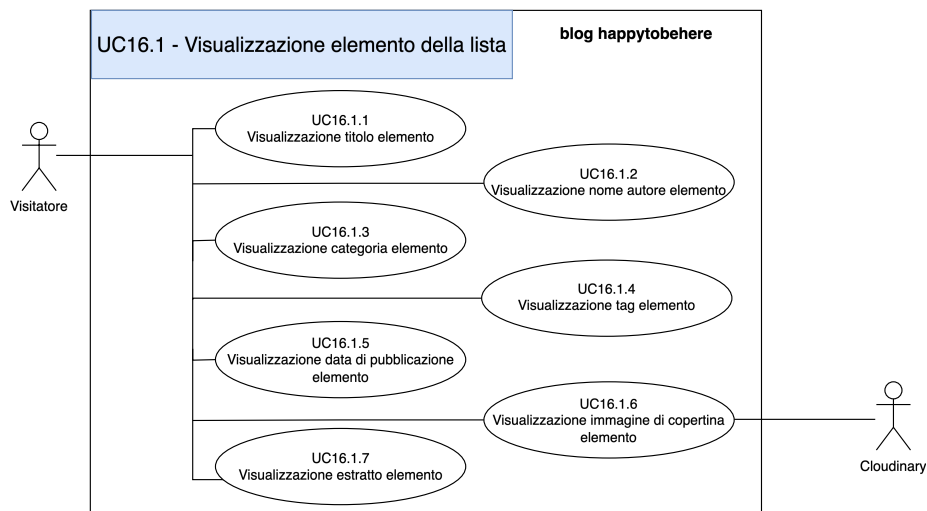


Figura 19: Diagramma dei sottocasi di UC16.1

3.4. ELENCO DEI CASI D'USO

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore desidera visualizzare il titolo di un elemento presente all'interno di una lista di articoli nel blog;
- **Precondizioni:** la pagina contenente la lista è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha visualizzato correttamente il titolo di un elemento presente all'interno di una lista di articoli;
- **Scenario principale:**
 1. il visitatore visualizza l'autore di un elemento di una lista di articoli del blog.

3.4.16.1.2. UC16.1.2 - Visualizzazione nome autore elemento

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore desidera visualizzare il nome dell'autore di un elemento presente all'interno di una lista di articoli nel blog;
- **Precondizioni:** la pagina contenente la lista è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha visualizzato correttamente il nome dell'autore di un elemento presente all'interno di una lista di articoli;
- **Scenario principale:**
 1. il visitatore visualizza correttamente il nome dell'autore di un elemento di una lista di articoli del blog.

3.4.16.1.3. UC16.1.3 - Visualizzazione categoria elemento

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore desidera visualizzare la categoria di un elemento presente all'interno di una lista di articoli nel blog;
- **Precondizioni:** la pagina contenente la lista è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha visualizzato correttamente la categoria di un elemento presente all'interno di una lista di articoli;
- **Scenario principale:**
 1. il visitatore visualizza la categoria di un elemento di una lista di articoli del blog.

3.4.16.1.4. UC16.1.4 - Visualizzazione tag elemento

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore desidera visualizzare un tag associato a un elemento presente all'interno di una lista di articoli nel blog;

3.4. ELENCO DEI CASI D'USO

- **Precondizioni:** la pagina contenente la lista è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha visualizzato correttamente un tag associato a un elemento presente all'interno di una lista di articoli;
- **Scenario principale:**
 1. il visitatore visualizza un tag associato a un elemento di una lista di articoli del blog.

3.4.16.1.5. UC16.1.5 - Visualizzazione data di pubblicazione elemento

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore desidera visualizzare la data di pubblicazione di un elemento presente all'interno di una lista di articoli nel blog;
- **Precondizioni:** la pagina contenente la lista è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha visualizzato correttamente la data di pubblicazione di un elemento presente all'interno di una lista di articoli;
- **Scenario principale:**
 1. il visitatore visualizza la data di pubblicazione di un elemento di una lista di articoli del blog.

3.4.16.1.6. UC16.1.6 - Visualizzazione immagine di copertina elemento

- **Attori principali:** Visitatore, Cloudinary;
- **Descrizione:** un visitatore desidera visualizzare l'immagine di copertina di un elemento presente all'interno di una lista di articoli nel blog;
- **Precondizioni:** la pagina contenente la lista è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha visualizzato correttamente l'immagine di copertina di un elemento presente all'interno di una lista di articoli;
- **Scenario principale:**
 1. l'immagine viene recuperata da Cloudinary e inserita all'interno della pagina;
 2. il visitatore visualizza l'immagine di copertina di un elemento di una lista di articoli del blog.

3.4.16.1.7. UC16.1.7 - Visualizzazione estratto elemento

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore desidera visualizzare l'estratto di un elemento presente all'interno di una lista di articoli nel blog;
- **Precondizioni:** la pagina contenente la lista è stata caricata correttamente sul browser del visitatore;

3.4. ELENCO DEI CASI D'USO

- **Postcondizioni:** il visitatore ha visualizzato correttamente l'estratto di un elemento presente all'interno di una lista di articoli;
- **Scenario principale:**
 1. il visitatore visualizza l'estratto di un elemento di una lista di articoli del blog.

3.4.17. UC17 - Visualizzazione articoli più recenti

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore desidera visualizzare la lista degli articoli più recenti presenti sul blog;
- **Precondizioni:** una qualsiasi pagina del blog è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha visualizzato correttamente la lista degli articoli più recenti;
- **Scenario principale:**
 1. Il visitatore seleziona il filtro relativo agli articoli più recenti;
 2. Il visitatore visualizza la pagina contenente la lista dei post più recenti del blog.

3.4.18. UC18 - Visualizzazione articoli per categoria

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore desidera visualizzare la lista degli articoli di una determinata categoria (da camminare, da pedalare...) presenti sul blog;
- **Precondizioni:** una qualsiasi pagina del blog è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha visualizzato correttamente la lista degli articoli di una determinata categoria presenti sul blog;
- **Scenario principale:**
 1. Il visitatore filtra i post selezionando una categoria tra quelle disponibili;
 2. Il visitatore visualizza la pagina contenente la lista dei post correlati alla categoria selezionata.

3.4.19. UC19 - Visualizzazione articoli più popolari

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore desidera visualizzare la lista degli articoli più popolari presenti sul blog;
- **Precondizioni:** una qualsiasi pagina del blog è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha visualizzato correttamente la lista degli articoli più popolari del blog;
- **Scenario principale:**

3.4. ELENCO DEI CASI D'USO

1. Il visitatore seleziona il filtro relativo agli articoli più popolari;
2. Il visitatore visualizza la pagina contenente la lista degli articoli più popolari.

3.4.20. UC20 - Visualizzazione articoli per autore

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore desidera visualizzare la lista degli articoli pubblicati da un determinato autore sul blog;
- **Precondizioni:** una qualsiasi pagina del blog è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha visualizzato correttamente la lista degli articoli pubblicati da un determinato autore del blog;
- **Scenario principale:**
 1. Il visitatore seleziona un autore tra quelli presenti nel blog;
 2. Il visitatore visualizza la pagina contenente la lista degli articoli pubblicati da quell'autore.

3.4.21. UC21 - Visualizzazione articoli per tag

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore desidera visualizzare la lista degli articoli relativi a un determinato tag presenti sul blog;
- **Precondizioni:** una qualsiasi pagina del blog è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha visualizzato correttamente la lista degli articoli correlati al tag;
- **Scenario principale:**
 1. Il visitatore filtra i post selezionando un tag tra quelli disponibili;
 2. Il visitatore visualizza la pagina contenente la lista dei post correlati al tag selezionato.

3.4.22. UC22 - Visualizzazione singolo articolo

- **Attori principali:** Visitatore, Cloudinary, [API](#) Girolibero;
- **Descrizione:** un visitatore desidera visualizzare interamente un articolo presente nel blog;
- **Precondizioni:** la pagina contenente la lista di post dove è contenuto l'articolo è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha visualizzato correttamente l'articolo del blog;
- **Scenario principale:**
 1. Il visitatore seleziona un articolo tra quelli presenti nella lista di post;
 2. Il visitatore legge per intero l'articolo.

3.4. ELENCO DEI CASI D'USO

3.4.22.1. UC22.1 - Visualizzazione titolo articolo

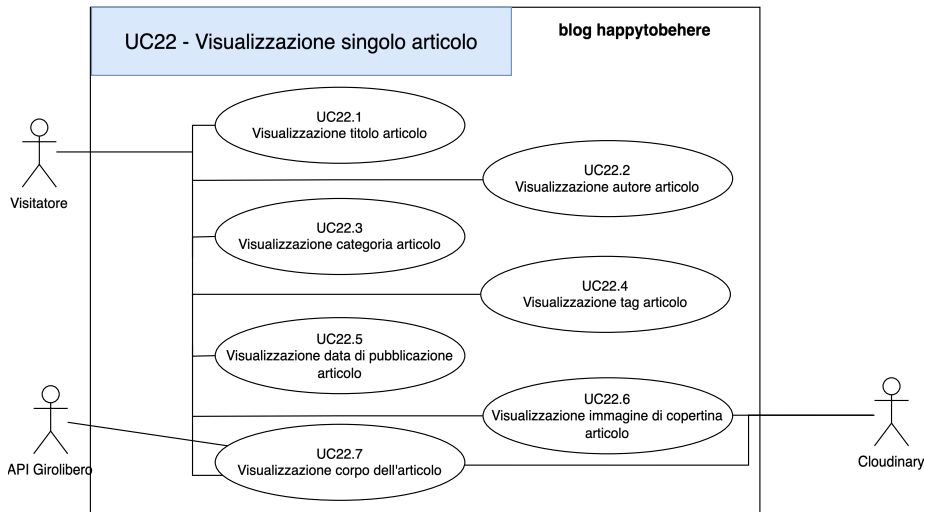


Figura 20: Diagramma dei sottocasi di UC22

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore desidera visualizzare il titolo di un articolo che sta leggendo, presente sul blog;
- **Precondizioni:** la pagina contenente l'articolo è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha visualizzato correttamente il titolo dell'articolo;
- **Scenario principale:**
 1. Il visitatore visualizza il titolo dell'articolo che sta leggendo.

3.4.22.2. UC22.2 - Visualizzazione autore articolo

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore desidera visualizzare l'autore di un articolo che sta leggendo, presente sul blog;
- **Precondizioni:** la pagina contenente l'articolo è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha visualizzato correttamente l'autore dell'articolo;
- **Scenario principale:**
 1. Il visitatore visualizza l'autore dell'articolo che sta leggendo.

3.4.22.3. UC22.3 - Visualizzazione categoria articolo

- **Attori principali:** Visitatore;

3.4. ELENCO DEI CASI D'USO

- **Descrizione:** un visitatore desidera visualizzare la categoria di un articolo che sta leggendo, presente sul blog;
- **Precondizioni:** la pagina contenente l'articolo è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha visualizzato correttamente la categoria dell'articolo;
- **Scenario principale:**
 1. Il visitatore visualizza la categoria dell'articolo che sta leggendo.

3.4.22.4. UC22.4 - Visualizzazione tag articolo

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore desidera visualizzare i tag associati all'articolo che sta leggendo, presente sul blog;
- **Precondizioni:** la pagina contenente l'articolo è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha visualizzato correttamente i tag associati all'articolo;
- **Scenario principale:**
 1. Il visitatore visualizza i tag associati all'articolo che sta leggendo.

3.4.22.5. UC22.5 - Visualizzazione data di pubblicazione articolo

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore desidera visualizzare la data di pubblicazione di un articolo che sta leggendo, presente sul blog;
- **Precondizioni:** la pagina contenente l'articolo è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha visualizzato correttamente la data di pubblicazione dell'articolo;
- **Scenario principale:**
 1. Il visitatore visualizza la data di pubblicazione dell'articolo che sta leggendo.

3.4.22.6. UC22.6 - Visualizzazione immagine di copertina articolo

- **Attori principali:** Visitatore, Cloudinary;
- **Descrizione:** un visitatore desidera visualizzare l'immagine di copertina di un articolo che sta leggendo, presente sul blog;
- **Precondizioni:** la pagina contenente l'articolo è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha visualizzato correttamente l'immagine di copertina dell'articolo;

3.4. ELENCO DEI CASI D'USO

- **Scenario principale:**
 1. L'immagine di copertina viene recuperata da Cloudinary e caricata nell'articolo;
 2. Il visitatore visualizza l'immagine di copertina dell'articolo che sta leggendo.

3.4.22.7. UC22.7 - Visualizzazione corpo dell'articolo

- **Attori principali:** Visitatore, Cloudinary, API Girolibero;
- **Descrizione:** un visitatore desidera visualizzare il corpo di un articolo che sta leggendo, presente sul blog;
- **Precondizioni:** la pagina contenente l'articolo è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha visualizzato correttamente il corpo dell'articolo;
- **Scenario principale:**
 1. Le immagini presenti vengono recuperate da Cloudinary e caricate nel corpo dell'articolo;
 2. i dati dei viaggi vengono recuperati tramite l'API Girolibero;
 3. Il visitatore visualizza il corpo dell'articolo che sta leggendo.

3.4.22.8. UC22.8 - Visualizzazione descrizione autore dell'articolo

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore desidera visualizzare la descrizione dell'autore di un articolo che sta leggendo, presente sul blog;
- **Precondizioni:** la pagina contenente l'articolo è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha visualizzato correttamente la descrizione dell'autore dell'articolo;
- **Scenario principale:**
 1. Il visitatore visualizza la descrizione dell'autore dell'articolo che sta leggendo.

3.4.23. UC23 - Ricerca all'interno del blog

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore desidera effettuare una ricerca per tag, categoria, autore... all'interno del blog;
- **Precondizioni:** una qualsiasi pagina del blog è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha effettuato una ricerca sul blog;
- **Scenario principale:**

1. Il visitatore inserisce del testo sull'apposito campo presente nel blog e conferma la ricerca.

- **Inclusioni:**
 - Visualizzazione lista post;

3.4.24. UC24 - Iscrizione alla newsletter

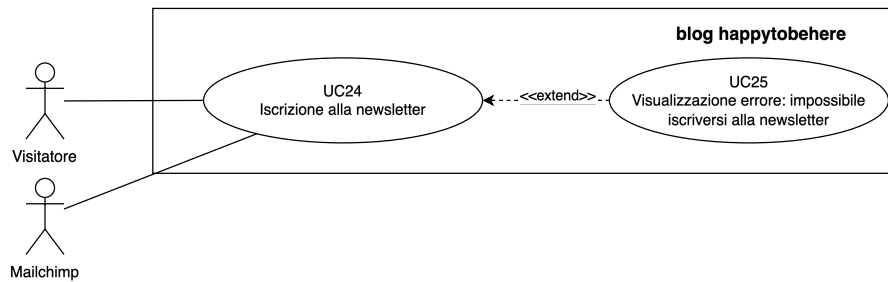


Figura 21: Diagramma dei casi d'uso UC24 - 25

- **Attori principali:** Visitatore, Mailchimp;
- **Descrizione:** un visitatore desidera iscriversi alla [newsletter](#) di Girolibero.
- **Precondizioni:** la pagina del blog contenente il form di iscrizione alla newsletter è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore si è iscritto correttamente alla newsletter di Girolibero;
- **Scenario principale:**
 1. Il visitatore compila il form per iscriversi alla newsletter presente sul blog inserendo i campi richiesti;
 2. Mailchimp completa la procedura di iscrizione, inserendo il visitatore nella lista della newsletter;
- **Estensioni:**
 - Visualizzazione errore: impossibile iscriversi alla newsletter.

3.4. ELENCO DEI CASI D'USO

3.4.24.1. UC24.1 - Inserimento nome

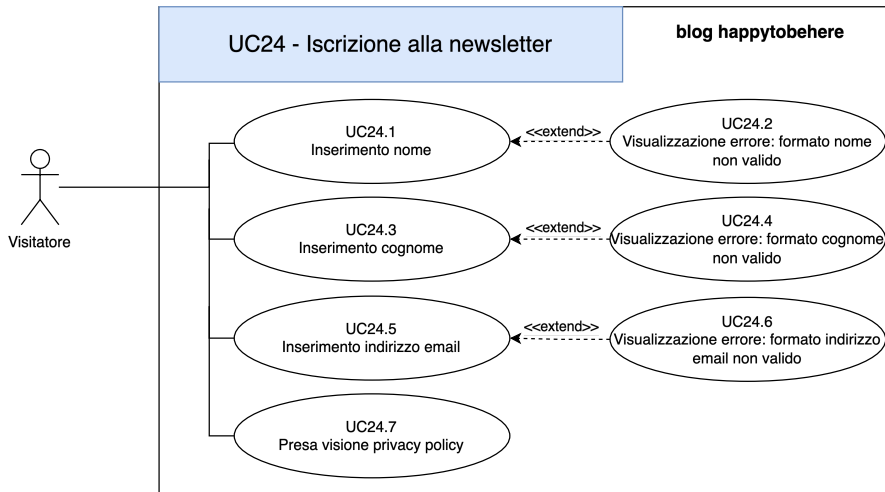


Figura 22: Diagramma dei sottocasi di UC24

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore, durante la compilazione del modulo di iscrizione alla [newsletter](#) Girolibero, desidera inserire il suo nome;
- **Precondizioni:** la pagina del blog contenente il form di iscrizione alla newsletter è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha inserito correttamente nel form di iscrizione il suo nome;
- **Scenario principale:**
 1. Il visitatore inserisce il suo nome nel modulo di iscrizione alla newsletter Girolibero;
- **Estensioni:**
 - Visualizzazione errore: formato nome non valido.

3.4.24.2. UC24.2 - Visualizzazione errore: formato nome non valido

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore, durante la compilazione del modulo di iscrizione alla [newsletter](#) Girolibero, ha inserito il proprio nome in un formato non valido e pertanto visualizza un messaggio di errore;
- **Precondizioni:** il visitatore ha compilato il form di iscrizione alla newsletter inserendo il proprio nome in un formato non valido;
- **Postcondizioni:** il visitatore ha visualizzato il messaggio di errore relativo alla compilazione del proprio nome all'interno del form;
- **Scenario principale:** il visitatore visualizza il messaggio di errore relativo alla compilazione del proprio nome all'interno del form.

3.4.24.3. UC24.3 - Inserimento cognome

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore, durante la compilazione del modulo di iscrizione alla [newsletter](#) Girolibero, desidera inserire il suo cognome;
- **Precondizioni:** la pagina del blog contenente il form di iscrizione alla newsletter è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha inserito correttamente nel form di iscrizione il suo cognome;
- **Scenario principale:**
 1. Il visitatore inserisce il suo cognome nel modulo di iscrizione alla newsletter Girolibero;
- **Estensioni:**
 - Visualizzazione errore: formato cognome non valido.

3.4.24.4. UC24.4 - Visualizzazione errore: formato cognome non valido

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore, durante la compilazione del modulo di iscrizione alla [newsletter](#) Girolibero, ha inserito il proprio cognome in un formato non valido e pertanto visualizza un messaggio di errore;
- **Precondizioni:** il visitatore ha compilato il form di iscrizione alla newsletter inserendo il proprio cognome in un formato non valido;
- **Postcondizioni:** il visitatore ha visualizzato il messaggio di errore relativo alla compilazione del proprio cognome all'interno del form;
- **Scenario principale:**
 - il visitatore visualizza il messaggio di errore relativo alla compilazione del proprio cognome all'interno del form.

3.4.24.5. UC24.5 - Inserimento indirizzo email

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore, durante la compilazione del modulo di iscrizione alla [newsletter](#) Girolibero, desidera inserire il suo indirizzo email;
- **Precondizioni:** la pagina del blog contenente il form di iscrizione alla newsletter è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha inserito correttamente nel form di iscrizione il suo indirizzo email;
- **Scenario principale:**
 - Il visitatore inserisce il suo indirizzo email nel modulo di iscrizione alla newsletter Girolibero;
- **Estensioni:**

3.4. ELENCO DEI CASI D'USO

- Visualizzazione errore: formato indirizzo email non valido.

3.4.24.5.1. UC24.6 - Visualizzazione errore: formato indirizzo email non valido

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore, durante la compilazione del modulo di iscrizione alla [newsletter](#) Girolibero, ha inserito il proprio indirizzo email in un formato non valido e pertanto visualizza un messaggio di errore;
- **Precondizioni:** il visitatore ha compilato il form di iscrizione alla newsletter inserendo il proprio indirizzo email in un formato non valido;
- **Postcondizioni:** il visitatore ha visualizzato il messaggio di errore relativo alla compilazione del proprio indirizzo email all'interno del form;
- **Scenario principale:**
 - il visitatore visualizza il messaggio di errore relativo alla compilazione del proprio indirizzo email all'interno del form.

3.4.24.5.2. UC24.7 - Presa visione privacy policy

- **Attori principali:** Visitatore;
- **Descrizione:** un visitatore, durante la compilazione del modulo di iscrizione alla [newsletter](#) Girolibero, desidera prendere visione della privacy policy di Girolibero;
- **Precondizioni:** la pagina del blog contenente il form di iscrizione alla newsletter è stata caricata correttamente sul browser del visitatore;
- **Postcondizioni:** il visitatore ha preso visione della privacy policy di Girolibero;
- **Scenario principale:**
 - Il visitatore prende visione della privacy policy visitando l'apposito link presente all'interno del form.

3.4.25. UC25 - Visualizzazione errore: impossibile iscriversi alla newsletter

- **Attori principali:** Visitatore;
- **Descrizione:** si è verificato un errore nel server durante la richiesta di iscrizione alla [newsletter](#) di Girolibero di un visitatore;
- **Precondizioni:** un visitatore ha compilato correttamente il form di iscrizione alla newsletter, ma è avvenuto un errore nel server durante l'elaborazione della richiesta;
- **Postcondizioni:** il visitatore ha visualizzato un messaggio di errore sulla pagina del blog;
- **Scenario principale:**

1. Il visitatore, che non è riuscito ad iscriversi alla newsletter, visualizza il messaggio di errore sulla pagina del blog.

3.5. Requisiti

In questa sezione sono elencati tutti i requisiti del progetto, raggruppati per tipologia.

Ogni requisito è identificato univocamente da un codice definito con il seguente formato:

R-[numero]-[tipologia]-[priorità]

dove:

- **numero**: un numero progressivo a due cifre che parte da 01;
- **tipologia** del requisito:
 - **F (funzionale)**: indica una funzione del prodotto (o di uno o più componenti), definendone il comportamento;
 - **Q (qualità)**: definisce le caratteristiche di qualità del prodotto software;
 - **V (vincolo)**: specifica i limiti e le restrizioni imposte dall'azienda sul prodotto;
- **priorità**: il grado di importanza assegnato al requisito con un numero da 1 a 3:
 1. Requisito obbligatorio che deve essere necessariamente soddisfatto;
 2. Requisito desiderabile, il cui soddisfacimento è apprezzato dall'azienda;
 3. Requisito opzionale, la cui realizzazione è a mia discrezione in base all'andamento del progetto di stage.

Per ogni requisito vengono riportate anche la descrizione e la fonte; quest'ultima può essere un caso d'uso specifico oppure una riunione interna effettuata con il team coinvolto nel progetto.

La **Tabella 3** riporta i requisiti di funzionalità.

ID Requisito	Descrizione	Fonte
R-01-F-1	L'autore deve avere la possibilità di inserire le coordinate geografiche dei luoghi correlati all'articolo utilizzando l'editor di WordPress.	UC1
R-02-F-1	L'autore che desidera inserire un luogo correlato all'articolo utilizzando l'editor deve necessariamente specificarne la latitudine.	UC1.1

3.5. REQUISITI

ID Requisito	Descrizione	Fonte
R-03-F-1	L'autore che desidera inserire un luogo correlato all'articolo utilizzando l'editor deve necessariamente specificarne la longitudine.	UC1.2
R-04-F-1	L'autore deve avere la possibilità di visualizzare le coordinate geografiche dei luoghi correlati all'articolo utilizzando l'editor di WordPress.	UC2 UC2.1
R-05-F-1	L'autore che desidera visualizzare le coordinate dei luoghi correlati all'articolo utilizzando l'editor deve essere in grado di visualizzarne la latitudine	UC2.1.1
R-06-F-1	L'autore che desidera visualizzare le coordinate dei luoghi correlati all'articolo utilizzando l'editor deve essere in grado di visualizzarne la longitudine	UC2.1.2
R-07-F-1	L'autore deve avere la possibilità di rimuovere le coordinate geografiche dei luoghi correlati all'articolo utilizzando l'editor di WordPress.	UC3
R-08-F-1	L'autore deve avere la possibilità di inserire il blocco "indice", contenente la lista di link ai relativi paragrafi dell'articolo, all'interno del corpo dell'articolo utilizzando l'editor di WordPress.	UC4
R-09-F-1	L'autore deve avere la possibilità di rimuovere il blocco "indice", contenente la lista di link ai relativi paragrafi dell'articolo, dal corpo dell'articolo utilizzando l'editor di WordPress.	UC5
R-10-F-1	L'autore deve avere la possibilità di visualizzare il blocco "indice", contenente la lista di link ai relativi paragrafi dell'articolo, all'interno del corpo dell'articolo utilizzando l'editor di WordPress.	UC6
R-11-F-1	L'autore deve avere la possibilità di inserire il blocco "carosello viaggi", contenente alcuni viaggi organizzati da Girolibero, all'interno del corpo	UC7

3.5. REQUISITI

ID Requisito	Descrizione	Fonte
	dell'articolo utilizzando l'editor di WordPress.	
R-12-F-1	L'autore deve avere la possibilità di rimuovere il blocco "carosello viaggi", contenente alcuni viaggi organizzati da Girolibero, dal corpo dell'articolo utilizzando l'editor di WordPress.	UC8
R-13-F-1	L'autore deve avere la possibilità di visualizzare il blocco "carosello viaggi", contenente alcuni viaggi organizzati da Girolibero, all'interno del corpo dell'articolo utilizzando l'editor di WordPress.	UC9 UC9.1 UC9.1.1
R-14-F-1	L'autore deve avere la possibilità di visualizzare il titolo di ogni viaggio presente nel blocco "carosello viaggi" all'interno del corpo dell'articolo, utilizzando l'editor di WordPress.	UC9.1.1.1
R-15-F-1	L'autore deve avere la possibilità di visualizzare l'immagine di copertina di ogni viaggio, fornita da Cloudinary, presente nel blocco "carosello viaggi" all'interno del corpo dell'articolo, utilizzando l'editor di WordPress.	UC9.1.1.2
R-16-F-1	L'autore deve avere la possibilità di visualizzare la tipologia di ogni viaggio presente nel blocco "carosello viaggi" all'interno del corpo dell'articolo, utilizzando l'editor di WordPress.	UC9.1.1.3
R-17-F-1	L'autore deve avere la possibilità di visualizzare la modalità di ogni viaggio presente nel blocco "carosello viaggi" all'interno del corpo dell'articolo, utilizzando l'editor di WordPress.	UC9.1.1.4
R-18-F-1	L'autore che desidera inserire un viaggio nel blocco "carosello viaggi", contenente alcuni viaggi organizzati da Girolibero, all'interno del corpo dell'articolo, deve necessariamente specificarne il codweb utilizzando l'editor di WordPress.	UC10
R-19-F-1	È necessario che un autore visualizzi un messaggio di errore se il codweb di un	UC11

3.5. REQUISITI

ID Requisito	Descrizione	Fonte
	viaggio inserito nel blocco "carosello viaggi" non fosse esistente.	
R-20-F-1	L'autore che desidera rimuovere un viaggio dal blocco "carosello viaggi", contenente alcuni viaggi organizzati da Girolibero, all'interno del corpo dell'articolo, deve necessariamente specificarne il codweb utilizzando l'editor di WordPress.	UC12
R-21-F-1	L'autore deve avere la possibilità di inserire la sua descrizione biografica al termine dell'articolo utilizzando l'editor di WordPress.	UC13
R-22-F-1	L'autore deve avere la possibilità di rimuovere la sua descrizione biografica dal fondo dell'articolo utilizzando l'editor di WordPress.	UC14
R-23-F-1	Il visitatore del blog deve avere la possibilità di visualizzare la mappa geografica contenente le localizzazioni degli articoli che sono associati a delle coordinate.	UC15 UC15.1
R-24-F-1	Il visitatore del blog deve avere la possibilità di visualizzare il titolo di ogni elemento presente nella mappa geografica degli articoli che sono associati a delle coordinate.	UC15.1.1
R-25-F-1	Il visitatore del blog deve avere la possibilità di visualizzare l'immagine di copertina di ogni elemento presente nella mappa geografica degli articoli che sono associati a delle coordinate.	UC15.1.2
R-26-F-1	Il visitatore del blog deve avere la possibilità di visualizzare la lista degli articoli più recenti presenti sul sito.	UC16 UC17
R-27-F-1	Il visitatore del blog deve avere la possibilità di visualizzare la lista degli articoli presenti sul sito suddivisi per categoria.	UC16 UC18

3.5. REQUISITI

ID Requisito	Descrizione	Fonte
R-28-F-1	Il visitatore del blog deve avere la possibilità di visualizzare la lista degli articoli più popolari presenti sul sito.	UC16 UC19
R-29-F-1	Il visitatore del blog deve avere la possibilità di visualizzare la lista degli articoli presenti sul sito suddivisi per autore.	UC16 UC20
R-30-F-1	Il visitatore del blog deve avere la possibilità di visualizzare la lista degli articoli presenti sul sito suddivisi per tag.	UC16 UC21
R-31-F-1	Il visitatore del blog deve avere la possibilità di visualizzare il titolo di ogni elemento di una lista di articoli presente sul sito.	UC16.1 UC16.1.1
R-32-F-1	Il visitatore del blog deve avere la possibilità di visualizzare il nome dell'autore di ogni elemento di una lista di articoli presente sul sito.	UC16.1 UC16.1.2
R-33-F-1	Il visitatore del blog deve avere la possibilità di visualizzare la categoria di ogni elemento di una lista di articoli presente sul sito.	UC16.1 UC16.1.3
R-34-F-1	Il visitatore del blog deve avere la possibilità di visualizzare i tag di ogni elemento di una lista di articoli presente sul sito.	UC16.1 UC16.1.4
R-35-F-1	Il visitatore del blog deve avere la possibilità di visualizzare la data di pubblicazione di ogni elemento di una lista di articoli presente sul sito.	UC16.1 UC16.1.5
R-36-F-1	Il visitatore del blog deve avere la possibilità di visualizzare l'immagine di copertina di ogni elemento di una lista di articoli presente sul sito.	UC16.1 UC16.1.6
R-37-F-1	Il visitatore del blog deve avere la possibilità di visualizzare l'estratto di ogni elemento di una lista di articoli presente sul sito.	UC16.1 UC16.1.7

3.5. REQUISITI

ID Requisito	Descrizione	Fonte
R-38-F-1	Il visitatore del blog deve avere la possibilità di leggere per interno un articolo presente in una lista sul sito.	UC22
R-39-F-1	Il visitatore del blog deve avere la possibilità di visualizzare il titolo dell'articolo che sta leggendo.	UC22.1
R-40-F-1	Il visitatore del blog deve avere la possibilità di visualizzare il nome dell'autore dell'articolo che sta leggendo.	UC22.2
R-41-F-1	Il visitatore del blog deve avere la possibilità di visualizzare la categoria dell'articolo che sta leggendo.	UC22.3
R-42-F-1	Il visitatore del blog deve avere la possibilità di visualizzare i tag associati all'articolo che sta leggendo.	UC22.4
R-43-F-1	Il visitatore del blog deve avere la possibilità di visualizzare la data di pubblicazione dell'articolo che sta leggendo.	UC22.5
R-44-F-1	Il visitatore del blog deve avere la possibilità di visualizzare l'immagine di copertina dell'articolo che sta leggendo.	UC22.6
R-45-F-1	Il visitatore del blog deve avere la possibilità di visualizzare il corpo dell'articolo che sta leggendo.	UC22.7
R-46-F-1	Il visitatore del blog deve avere la possibilità di effettuare una ricerca testuale tra gli articoli nell'apposito campo presente sul sito.	UC23
R-47-F-1	Il visitatore del blog che ha appena effettuato una ricerca all'interno del sito tramite l'apposito campo deve visualizzare la lista dei risultati.	UC16, UC23
R-48-F-2	Il visitatore del blog deve avere la possibilità di iscriversi alla newsletter di Girolibero tramite l'apposito form presente all'interno del sito.	UC24
R-49-F-2	Il visitatore del blog che desidera iscriversi alla newsletter di Girolibero	UC24.1

3.5. REQUISITI

ID Requisito	Descrizione	Fonte
	tramite l'apposito form presente all'interno del sito, deve necessariamente inserire il proprio nome.	
R-50-F-2	Il visitatore del blog che sta compilando il form di iscrizione alla newsletter sul sito, deve necessariamente visualizzare un messaggio di errore se ha inserito il proprio nome in un formato non valido.	UC24.2
R-51-F-2	Il visitatore del blog che desidera iscriversi alla newsletter di Girolibero tramite l'apposito form presente all'interno del sito, deve necessariamente inserire il proprio cognome.	UC24.3
R-52-F-2	Il visitatore del blog che sta compilando il form di iscrizione alla newsletter sul sito, deve necessariamente visualizzare un messaggio di errore se ha inserito il proprio cognome in un formato non valido.	UC24.4
R-53-F-2	Il visitatore del blog che desidera iscriversi alla newsletter di Girolibero tramite l'apposito form presente all'interno del sito, deve necessariamente inserire il proprio indirizzo email.	UC24.5
R-54-F-2	Il visitatore del blog che sta compilando il form di iscrizione alla newsletter sul sito, deve necessariamente visualizzare un messaggio di errore se ha inserito il proprio indirizzo email in un formato non valido.	UC24.6
R-55-F-2	Il visitatore del blog che desidera iscriversi alla newsletter di Girolibero tramite l'apposito form presente all'interno del sito, deve necessariamente prendere visione della privacy policy di Girolibero.	UC24.7
R-56-F-2	Il visitatore del blog che ha completato e inviato il form di iscrizione alla newsletter di Girolibero tramite l'apposito form presente sul sito, deve necessariamente visualizzare un messaggio di errore se si è verificato un	UC25

3.5. REQUISITI

ID Requisito	Descrizione	Fonte
	errore nel server durante la gestione della richiesta.	
R-57-F-3	Il visitatore del blog deve avere la possibilità di visualizzare la mappa del sito, contenente i link delle varie pagine.	Riunione
R-58-F-3	Il visitatore del blog deve avere la possibilità di visualizzare una lista ordinata di tutti i tag presenti sul blog.	Riunione

Tabella 3: Requisiti di funzionalità

La **Tabella 4** riporta i requisiti di qualità.

ID Requisito	Descrizione	Fonte
R-01-Q-1	Il blog deve essere testato utilizzando un validatore di markup e non devono essere presenti errori di questo tipo.	Riunione
R-02-Q-1	Il blog deve essere completamente funzionante sull'ultima versione disponibile di Google Chrome (v.125).	Riunione
R-03-Q-1	Il blog deve essere completamente funzionante sull'ultima versione disponibile di Safari (v.17.4 del 05-03-2024).	Riunione
R-04-Q-1	Il blog deve essere completamente funzionante sull'ultima versione disponibile di Mozilla Firefox (v.125.0.1 del 16-04-2024).	Riunione
R-05-Q-1	È necessario realizzare il codice sorgente con un sistema di versionamento.	Riunione
R-06-Q-3	Deve essere testata l'accessibilità del blog con un validatore ed eventualmente adeguata al livello <i>WCAG (Web Content Accessibility Guidelines)</i> _[9] A.	Riunione
R-07-Q-2	Deve essere predisposto il tracciamento del tempo di lettura degli articoli utilizzando Google Analytics.	Riunione
R-08-Q-2	Deve essere predisposto il tracciamento del numero di visualizzazioni degli articoli utilizzando Google Analytics.	Riunione

3.5. REQUISITI

ID Requisito	Descrizione	Fonte
R-09-Q-2	Deve essere predisposto il tracciamento dei link che reindirizzano a girolibero.it provenienti dal blog.	Riunione
R-10-Q-1	È necessario rilasciare delle linee guida per la stesura corretta degli articoli agli autori del blog.	Riunione

Tabella 4: Requisiti di qualità

La **Tabella 5** riporta i requisiti di vincolo.

ID Requisito	Descrizione	Fonte
R-01-V-1	Deve essere usato WordPress come CMS del blog.	Riunione
R-02-V-1	Il backend del blog deve essere realizzato in PHP.	Riunione
R-03-V-1	Il frontend del blog deve essere realizzato in Twig.	Riunione
R-04-V-1	Gli script per il frontend devono essere realizzati in JavaScript.	Riunione
R-05-V-2	Gli script per il frontend devono essere realizzati utilizzando la libreria JavaScript jQuery.	Riunione
R-06-V-1	Il frontend del blog deve essere realizzato in Twig.	Riunione
R-07-V-1	I fogli di stile devono essere realizzati in Sass.	Riunione
R-08-V-2	Devono essere utilizzati i fogli di stile già presenti su girolibero.it per quanto possibile.	Riunione
R-09-V-2	Per la realizzazione della mappa geografica dei post deve essere utilizzata la libreria Javascript di MapBox.	Riunione
R-10-V-1	La gestione delle immagini all'interno delle pagine deve essere realizzata mediante Cloudinary.	UC9 UC9.1 UC9.1.1 UC9.1.1.2

ID Requisito	Descrizione	Fonte
		UC15 UC15.1 UC15.1.2 UC16 UC16.1 UC16.1.6 UC22 UC22.6 UC22.7
R-11-V-1	Per le scelte di design è necessario seguire le indicazioni dell'UX designer.	Riunione

Tabella 5: Requisiti di vincolo

Infine, la **Tabella 6** riepiloga i requisiti identificati, raggruppandoli per tipologie e importanza.

Tipologia	Obbligatoriosi	Desiderabili	Opzionali
Funzionali	47	9	2
Qualità	6	3	1
Vincolo	8	3	0
Totale	61	15	3

Tabella 6: Riepilogo dei requisiti

Capitolo 4.

Tecnologie

In questo capitolo vengono presentate tutte le tecnologie impiegate nello sviluppo del progetto.

4.1. WordPress

WordPress è uno dei **CMS** più diffusi al mondo, utilizzato per la creazione di siti web e blog in modo intuitivo e veloce. È un software open-source, ovvero il cui codice sorgente è disponibile a tutti e consente agli utenti di effettuare personalizzazioni e modifiche in base alle proprie esigenze. WordPress è caratterizzato dalla presenza di temi e *plugin*_[g]:

- tema: componente che determina l'aspetto e il layout di un sito web. Caratterizza il **frontend**, per esempio, sulla disposizione delle pagine, il posizionamento degli elementi e i colori utilizzati. Esistono temi gratuiti e versioni a pagamento;
- plugin: componente che estende le funzionalità base di WordPress. Può aggiungere funzionalità di qualsiasi tipologia: sicurezza, *SEO (Search Engine Optimization)*_[g], immagini,...

Sono installabili con un semplice click e generalmente gratuiti.

Introdotta dalla versione 5.0, Gutenberg^[3] è l'editor di contenuti di WordPress basato sui blocchi. Rispetto all'editor precedente, consente di creare elementi più complessi e dinamici tramite blocchi predefiniti, come paragrafi, titoli e immagini, o personalizzati, attraverso i plugin.

Esistono due varianti di WordPress: WordPress.com e WordPress.org:

- WordPress.com: variante più semplice di WordPress, utilizzata per creare e gestire un sito web senza il carico della gestione tecnica. La personalizzazione è limitata all'utilizzo di temi e plugin disponibili pubblicamente, i quali possono non essere completamente accessibili a seconda del piano sottoscritto.
- WordPress.org^[4]: denominato anche "WordPress self-hosted", è la versione di WordPress che può essere installata e ospitata autonomamente. Per il suo utilizzo è necessario avere delle solide basi di programmazione web: è possibile, infatti, personalizzare completamente il sito, installare temi e plugin esterni e modificare direttamente il codice sorgente per adattarlo alle proprie esigenze specifiche.

4.1. WORDPRESS

Il blog happytobehere utilizzava una versione di WordPress.org obsoleta. L'azienda ha riconfermato l'uso di questa tecnologia, aggiornandola all'ultima versione disponibile (v.6.5 del 2/04/2024) con l'editor Gutenberg.



Figura 23: Logo di WordPress

Per lavorare sul **backend** di WordPress.org è necessario usare il linguaggio PHP[5]: si tratta di un linguaggio di scripting che agisce sul server ed è ampiamente utilizzato per lo sviluppo web. PHP rappresenta il core del backend di WordPress. È impiegato per tutto, dalla gestione e interazione con il database, alla creazione dinamica delle pagine.



Figura 24: Logo di PHP

WordPress si appoggia a un database MySQL[6] per archiviare e gestire tutte le informazioni necessarie al funzionamento di un sito web. Al suo interno, per esempio, sono salvati in apposite tabelle i contenuti degli articoli e i relativi **metadati**, le tassonomie e gli utenti registrati al sito.



Figura 25: Logo di MySQL

4.2. Twig e Timber

Twig[7] è un potente motore di template per le pagine HTML che separa la logica di presentazione e il markup dal codice di applicazione PHP, migliorando così la manutenibilità e la leggibilità del codice.

Uno degli aspetti distintivi di Twig è la sua capacità di gestire in modo efficiente l'inclusione di template e l'ereditarietà. Questo significa che è possibile definire un template base con blocchi comuni ed estenderlo a template figli per aggiungere o sovrascrivere parti specifiche.

Le strutture dei template in Twig offrono la possibilità di gestire il flusso logico e la presentazione dei dati utilizzando blocchi condizionali come gli if statement: è possibile, infatti, controllare dinamicamente quali parti del template vengono renderizzate in base a condizioni specifiche.

Twig, tecnologia imposta dall'azienda, è stato impiegato per la realizzazione dei template delle pagine del blog.

Timber[8] è una libreria open-source per WordPress che utilizza Twig come motore di template per gestire le viste e i template HTML. È essenziale per la creazione di temi WordPress che sfruttano Twig.



Figura 26: Logo di Twig



Figura 27: Logo di Timber

4.3. Sass

SASS[9], acronimo di “Syntactically Awesome Style Sheets”, è un'estensione di CSS che aggiunge funzionalità avanzate e miglioramenti sintattici al linguaggio, rendendo la creazione dei fogli di stile più potente e organizzata. Vengono di seguito elencate alcune delle caratteristiche principali di Sass:

- compilazione: Sass richiede la compilazione del codice sorgente Sass in CSS per renderlo utilizzabile sulle pagine web;
- variabili: gli sviluppatori possono definire e riutilizzare valori come colori e dimensioni in tutto il progetto grazie alle variabili;
- nesting: Sass supporta l'annidamento delle regole CSS, permettendo agli sviluppatori di scrivere codice più chiaro e ben strutturato;
- mixin: sono blocchi di codice riutilizzabili che contengono una serie di dichiarazioni CSS; sono facilmente integrabili all'interno dei selettori, permettendo una gestione efficiente e standardizzata degli stili;
- funzioni: Sass supporta la creazione di funzioni personalizzate, consentendo agli sviluppatori di eseguire calcoli e manipolare valori all'interno dei fogli di stile;
- controllo del flusso: tramite espressioni condizionali e cicli, gli sviluppatori possono controllare il flusso logico delle regole CSS direttamente nei fogli di stile.

Bootstrap[10] è uno dei **framework frontend** più popolari al mondo, utilizzato per la creazione di interfacce utente responsive ed esteticamente gradevoli per siti web e applicazioni. È possibile integrare Bootstrap a Sass, importando i file

4.3. SASS

sorgenti necessari allo sviluppo dei fogli di stile: questo framework, infatti, offre numerosi mixins e funzioni pronti all'uso per facilitare la personalizzazione dei fogli di stile.

Sass con Bootstrap è la tecnologia imposta dall'azienda per la realizzazione dei fogli di stile del blog.



Figura 28: Logo di Sass

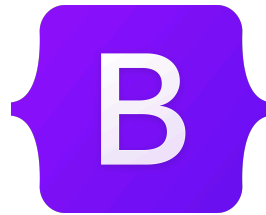


Figura 29: Logo di Bootstrap

4.4. jQuery

jQuery[11] è una libreria JavaScript molto popolare che semplifica notevolmente la scrittura di codice JavaScript per interagire con il **DOM** (*Document Object Model*), gestire eventi e comunicare con il server con chiamate **AJAX** (*Asynchronous JavaScript and XML*)^[9]. A differenza degli script PHP, uno script realizzato con JavaScript viene eseguito direttamente nel browser.

Si tratta di una tecnologia imposta dall'azienda, in quanto già presente sul sito principale di Girolibero.[11]



Figura 30: Logo di jQuery

4.5. Node.js

Node.js[12] è un ambiente di runtime open-source che consente di eseguire codice JavaScript al di fuori del browser web. Questa tecnologia risulta essenziale quando si lavora con strumenti JavaScript moderni e si sviluppa per l'editor a blocchi Gutenberg di WordPress.

Un plugin per un blocco di WordPress non è altro che un progetto Node.js: pertanto, l'utilizzo di questa tecnologia è necessario.



Figura 31: Logo di Node.js

4.6. React

React[13] è una libreria JavaScript open-source sviluppata per costruire *UI (User Interface)*_[9]. Suddivide l'UI in componenti indipendenti tra loro, che, a seconda dello stato che possiedono, possono renderizzare contenuti.

Questa libreria è impiegata principalmente per creare applicazioni web dinamiche, grazie alla sua capacità di aggiornare in modo efficiente i componenti dell'UI quando i dati cambiano.

Per creare un **blocco** personalizzato di WordPress, è necessario definire il componente React responsabile del rendering dell'interfaccia grafica dell'utente all'interno dell'editor Gutenberg. Questo permette agli utenti di interagire e visualizzare il blocco; pertanto, l'adozione di questa tecnologia è essenziale.

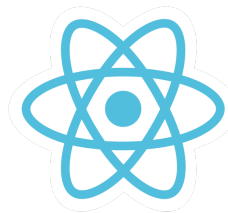


Figura 32: Logo di React

4.7. Cloudinary

Cloudinary[14] è una piattaforma cloud per la gestione e l'ottimizzazione di immagini e video. Consente di caricare e archiviare media all'interno di un cloud in modo sicuro; offre strumenti avanzati per la manipolazione delle immagini, come ridimensionamento, ritaglio, compressione e aggiunta di filtri. L'ottimizzazione automatica garantisce che le immagini siano caricate rapidamente su tutti i dispositivi, mantenendo la qualità.

Cloudinary, già utilizzato nel sito principale di Girolibero, è stato impiegato anche nel blog happytobehere per il recupero delle immagini.



Figura 33: Logo di Cloudinary

4.8. MapBox

MapBox[15] è una piattaforma di mapping e geolocalizzazione che offre potenti strumenti per la creazione di mappe personalizzate, con la possibilità di integrarle in applicazioni web e mobile. Mette a disposizione una libreria JavaScript per gli sviluppatori con cui è possibile eseguire facilmente operazioni sulle mappe, come l'inserimento di marker e popup nelle coordinate geografiche desiderate, grazie al supporto di formati come *GeoJSON*_[9].

MapBox ha la capacità di gestire grandi volumi di dati geografici in modo efficiente: utilizzando la funzionalità di clustering, è possibile aggregare automaticamente punti vicini in gruppi, migliorando così la leggibilità e la performance delle mappe con molti marker.

Grazie a MapBox Studio, uno dei servizi offerti, gli utenti possono personalizzare completamente lo stile delle mappe, modificando, per esempio, l'aspetto delle strade, degli edifici e delle città.

MapBox e la sua libreria sono state le principali tecnologie impiegate per la realizzazione della mappa degli articoli del blog.



Figura 34: Logo di MapBox

Capitolo 5.

Sviluppo

In questo capitolo vengono riportate le scelte di implementazione del progetto.

5.1. Il tema

La ristrutturazione completa del blog è stata affidata a un tema WordPress customizzato. Un tema è un insieme di file che definisce l'aspetto e il funzionamento di un sito web gestito con WordPress. Al suo interno sono riportati i layout delle pagine, con la possibilità di differenziare i modelli in base alla sezione del sito in cui ci si trova e di arricchire i template con funzionalità specifiche.

5.1.1. Struttura del tema

Viene di seguito riportato il contenuto della cartella `happytobehere-theme`, contenente il codice del tema realizzato.

```
happytobehere-theme/  
├── assets/.../  
│   ├── css/...  
│   ├── fonts/...  
│   ├── img/...  
│   ├── js/...  
│   └── scss/...  
├── views/...  
├── 404.php  
├── allTags.php  
├── archive.php  
├── author.php  
├── category.php  
├── composer.json  
├── functions.php  
├── index.php  
├── map.php  
├── package.json  
├── popular.php  
├── singular.php  
├── search.php  
└── sitemap.php
```

5.1. IL TEMA

```
├── static.php
└── tag.php
```

- **assets:** cartella contenente tutte le risorse utilizzate dal tema;
 - **css:** cartella contenente i fogli di stile compilati e utilizzati dal tema;
 - **fonts:** cartella contenente tutti i font del blog, importati dal sito principale di Girolibero;
 - **img:** cartella contenente tutte le immagini utilizzate all'interno delle pagine, come il logo del blog e le icone di navigazione;
 - **js:** cartella contenente il file di script del tema, realizzato con la libreria jQuery di JavaScript;
 - **scss:** cartella contenente i codici sorgente dello stile che verranno compilati in .css e salvati all'interno dell'omonima cartella;
- **views:** cartella contenente i template delle pagine realizzati in Twig;

```
views/
├── components/
│   ├── macros.twig
│   └── posts_list.twig
├── pages/
│   ├── allTags.twig
│   ├── 404.twig
│   ├── archive.twig
│   ├── author.twig
│   ├── category.twig
│   ├── index.twig
│   ├── map.twig
│   ├── popular.twig
│   ├── search.twig
│   ├── singular.twig
│   ├── sitemap.twig
│   ├── static.twig
│   ├── tag.twig
│   └── taxonomy.twig
└── base.twig
```

- **components/:** cartella contenente tutte le parti comuni a più modelli;
 - **macros.twig:** file contenente tutte le macro utilizzate dai template;
 - **posts_list.twig:** template della lista di articoli;
- **pages/:** cartella contenente tutti i template realizzati per le pagine del tema;
 - **allTags.twig:** template realizzato per la pagina contenente la lista di tutti i tag presenti sul blog. Eredita dal template `base.twig`;

- 404.twig: template realizzato per la pagina di errore 404 del blog. Eredita dal template base.twig;
 - archive.twig: template realizzato per le pagine contenenti delle liste di articoli del blog. Eredita dal template base.twig;
 - author.twig: template realizzato per la pagina contenente la lista degli articoli di un determinato autore. Eredita dal template archive.twig;
 - category.twig: template realizzato per la pagina contenente la lista degli articoli di una determinata categoria. Eredita dal template archive.twig;
 - index.twig: template realizzato per la pagina principale del blog, contenente la lista degli articoli più recenti. Eredita dal template archive.twig;
 - map.twig: template realizzato per la pagina contenente la mappa geografica degli articoli presenti sul blog. Eredita dal template base.twig;
 - popular.twig: template realizzato per la pagina contenente la lista degli articoli più popolari. Eredita dal template archive.twig;
 - search.twig: template realizzato per la pagina contenente i risultati di una ricerca effettuata sul blog. Eredita dal template base.twig;
 - singular.twig: template realizzato per la pagina contenente un singolo articolo del blog. Eredita dal template base.twig;
 - sitemap.twig: template realizzato per la pagina contenente la mappa del sito. Eredita dal template base.twig;
 - static.twig: template realizzato per le pagine statiche del blog, come “about us” e “contatti”. Eredita dal template base.twig;
 - tag.twig: template realizzato per la pagina contenente la lista degli articoli di un determinato tag. Eredita dal template archive.twig;
 - taxonomy.twig: template realizzato per le pagine contenenti la lista degli articoli di una determinata *tassonomia*_[g] (destinazione, regione, città). Eredita dal template archive.twig;
- ▶ base.twig: file di base che definisce la struttura generale di ogni pagina e dei suoi principali elementi, come header e footer.
 - composer.json: elenco generato in automatico di [metadati](#), configurazioni e dipendenze del tema;
 - functions.php: file utilizzato per gestire le funzionalità di WordPress e personalizzare il comportamento del sito;
 - package.json: elenco di tutte le librerie JavaScript e dei pacchetti necessari per il progetto; viene generato in automatico.

I file 404.php, allTags.php, archive.php, author.php, category.php, index.php, map.php, popular.php, singular.php, search.php, sitemap.php, static.php, tag.php, taxonomy.php renderizzano gli omonimi template Twig.

5.2. Menu

Il menu del blog presenta diversi elementi e si differenzia significativamente sia per contenuti che per layout tra mobile e desktop. La **Figura 35** mostra la versione desktop, mentre la **Figura 36** mostra la versione mobile.



Figura 35: Menu desktop del blog

5.2.1. Barra di ricerca

All'interno del blog è consentito effettuare ricerche degli articoli inserendo parole nell'apposito campo del menu. La barra mostra in tempo reale i risultati della ricerca, riportando per gli articoli individuati il titolo e un estratto del loro contenuto. Questa funzionalità è stata realizzata grazie a un apposito [plugin](#) di WordPress: [Ajax Search Lite](#)⁶.

Inviando il modulo della ricerca, l'utente verrà reindirizzato nella pagina contenente la lista dei risultati.

Nel desktop la barra è riportata per esteso in alto a sinistra; nel mobile, invece, viene visualizzata esclusivamente in seguito al click dell'apposita icona presente nella parte superiore dello schermo.

5.2.2. Pagine statiche

Alcune delle pagine statiche più importanti presenti sul blog sono riportate anche nel menu principale. Si tratta di pagine che l'azienda desidera mettere in evidenza rispetto ad altre.

Sono caratterizzate dal colore del testo tendente al nero e sono presenti a destra sul menu desktop, mentre su mobile sono raggiungibili dal menu a tendina.

Le pagine statiche del menu sono tre: about, contatti, mappa.

Per consentire il corretto funzionamento di queste voci, è stato definito un menu all'interno di WordPress con un nome univoco, nell'apposita sezione del tema e inserite al suo interno le pagine desiderate. Grazie a Timber, risulta immediato recuperare qualsiasi tipo di informazione legata a ogni voce del menu; le informazioni vengono fornite ai template con il metodo `get_menu()`.

5.2.3. Categorie

Le categorie sono contraddistinte da uno sfondo rosso della stessa tonalità del logo del blog e dai bordi arrotondati. Questi pulsanti sono presenti subito dopo le voci delle pagine statiche sia su desktop che mobile e consentono di filtrare gli articoli in base alla categoria scelta dall'utente.

Ulteriori dettagli sono presenti nella sezione **5.4.2 Tassonomie**.

Come per le pagine statiche, anche per le categorie è stato definito un menu all'interno di WordPress contenente tutte le categorie del blog.

⁶URL: <https://wordpress.org/plugins/ajax-search-lite/>

5.2.4. Menu mobile

Il menu mobile del blog si differenzia da quello desktop per i contenuti e il layout. Oltre alle voci delle pagine statiche e delle categorie, sono riportati altri elementi:

- tag: i dieci tag più utilizzati tra gli articoli del blog; è presente anche un collegamento alla pagina contenente la lista di tutti i tag del blog;
- i più popolari: i tre articoli con più visite del blog; successivamente, è possibile visualizzare la lista degli articoli più popolari tramite l'apposito link;
- social media: icone con collegamento ai profili Facebook, Instagram e YouTube dell'azienda.

Per la realizzazione di questo menu sono state importate dal sito principale di Girolibero le regole e gli script necessari ad animare e visualizzare correttamente il contenitore del menu.

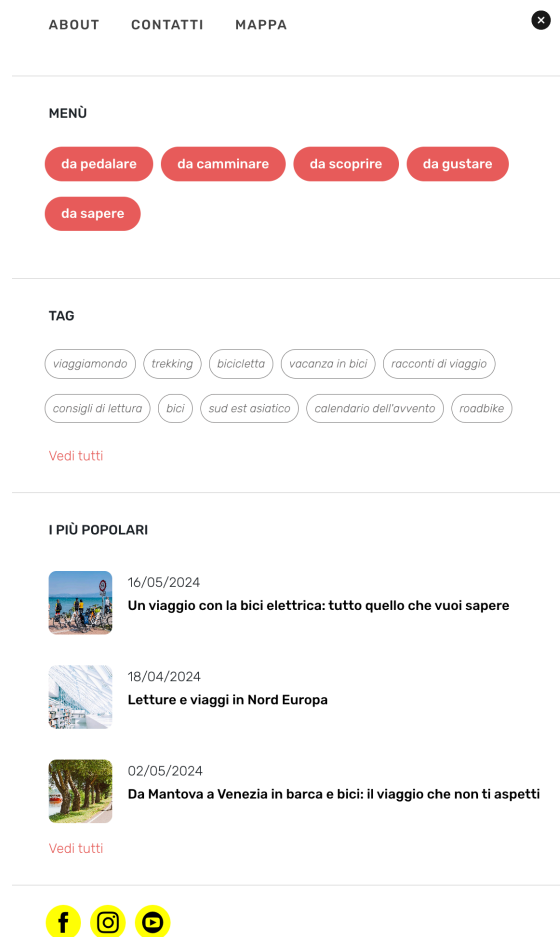


Figura 36: Menu mobile del blog

5.3. LISTA DEGLI ARTICOLI

5.3. Lista degli articoli

The screenshot shows the homepage of the blog 'HAPPY TO BE HERE'. At the top, there is a search bar with the text 'CERCA' and a navigation menu with links for 'ABOUT', 'CONTATTI', 'MAPPA', and several red buttons labeled 'da pedalare', 'da camminare', 'da scoprire', 'da gustare', and 'da sapere'. The main heading 'HAPPY TO BE HERE' is accompanied by a red heart icon. Below the heading, it says 'Il blog di Girolibero'. The page features a grid of six article cards. Each card includes a featured image, a date and author, a title, a short description, and a red button with a category label. The articles are: 1. 'Un viaggio con la bici elettrica: tutto quello che vuoi sapere' (16/05/2024, Redazione, 'da sapere'); 2. 'I muri delle Fiandre in bicicletta: cosa mangiare per affrontarli' (15/05/2024, Redazione, 'da pedalare'); 3. 'Da Mantova a Venezia in barca e bici: il viaggio che non ti aspetti' (02/05/2024, Redazione, 'da scoprire'); 4. 'Letture e viaggi in Nord Europa' (18/04/2024, Redazione, 'da scoprire'); 5. 'New York in bicicletta è ancora più bella' (11/04/2024, Redazione, 'da pedalare'); 6. 'La bellezza nascosta dell'Armenia, perché partire subito' (04/04/2024, Francesca Bedin, 'da scoprire'). A vertical yellow banner on the right side reads 'Girolibero viaggia facile e leggero'. At the bottom of the article grid, there are two buttons: 'consigli di lettura' and 'libri da leggere'.

Figura 37: Visualizzazione di una lista di articoli

Sono presenti diverse pagine che consentono di visualizzare una lista di articoli all'interno del blog, denominate pagine di archivio. Le pagine di archivio consentono di filtrare i post in base alle loro caratteristiche. Questa visualizzazione è presente:

- nella pagina iniziale, ordinando tutti i post per data di pubblicazione, come mostrato nella **Figura 37**;
- nella pagina dedicata a una categoria, ordinando i relativi post per data di pubblicazione;
- nella pagina dedicata a un tag, ordinando i relativi post per data di pubblicazione;
- nella pagina dedicata a un autore del blog, ordinando i relativi post per data di pubblicazione;

- nella pagina degli articoli più popolari, ordinando i relativi post per numero di visualizzazioni;
- nella pagina dedicata a una [tassonomia](#), ordinando i relativi post per data di pubblicazione. Questa visualizzazione verrà spesso richiamata nelle [newsletter](#) settimanali di Girolibero, a seconda del viaggio pubblicizzato. Ad esempio, se la newsletter vuole promuovere viaggi in Norvegia, sarà presente un link che rimanda a tutti gli articoli sulla Norvegia pubblicati sul blog, poiché vengono filtrati per la tassonomia “destinazione” con valore “Norvegia”.

Per ogni pagina, vengono visualizzati inizialmente i primi 10 articoli con la possibilità di caricarne altri 10 ogni qualvolta un utente lo desidera premendo l'apposito pulsante.

5.3.1. Creazione della lista

A seconda del file php in cui ci si trova, è sufficiente utilizzare il metodo di Timber `get_posts()` per recuperare le informazioni corrette relative agli articoli. Tutti i template Twig delle liste includono il file `posts_list.twig`, che fornisce la struttura e il file `macros.twig`, contenente le macro di utilità usate per costruire i singoli elementi.

Per ogni articolo della lista viene creato un contenitore che ne riporta:

- il titolo;
- l'immagine di copertina;
- la categoria;
- l'autore;
- l'estratto.

5.3.2. Caricamento di altri elementi

Il blog conta più di 100 articoli suddivisi tra le diverse categorie e tag. Per velocizzare il caricamento delle pagine è stato impostato il limite iniziale di articoli visualizzati a 10, con la possibilità di caricarne altri 10 semplicemente premendo l'apposito pulsante presente in fondo alla lista, mostrato nella [Figura 38](#).

A questo scopo è stata definita una chiamata [AJAX](#) di tipo POST all'interno del file di script.

Quando l'utente clicca sul pulsante, viene mostrata un'icona di caricamento per tutta la durata dell'operazione, mostrata nella [Figura 39](#). Lo script recupera il tipo dell'archivio visualizzato e aggiunge un valore specifico ai parametri della chiamata ajax. Un contatore tiene traccia del numero della pagina corrente, che deve essere necessariamente specificato all'interno della richiesta per il recupero corretto degli articoli.

All'interno del file `functions.php` è definita una funzione che viene chiamata durante questa operazione. A seconda del tipo della pagina di archivio che ha richiesto il caricamento di altri post, vengono recuperate le informazioni degli articoli usando il metodo Timber `get_posts()`. I post sono renderizzati utilizzando

5.3. LISTA DEGLI ARTICOLI

do il template `posts_list.twig` e inseriti all'interno della lista.

Una volta completata l'operazione, viene rimossa l'icona di caricamento e, se non sono ancora presenti articoli da caricare, viene nascosto anche il pulsante di caricamento.



Figura 38: Pulsante utilizzato nella lista di articoli



Figura 39: Icona di caricamento per la lista di articoli

5.4. Articolo singolo

La pagina che ha il compito di visualizzare interamente un articolo è chiamata `singular`. Anche in questo caso, il recupero dei contenuti è affidato a Timber, utilizzando il metodo `context()`.

Nell'omonimo template, `singular.twig`, viene definita la struttura dell'articolo, costituita come segue:

- header: mostrato nella [Figura 40](#), contiene immagine di copertina del post, data di pubblicazione, autore e categorie;
- content: riporta il contenuto del post;
- footer: contiene tutti i tag associati all'articolo;
- sezione autore: mostra l'immagine e la descrizione se l'apposito `post meta[g]` è impostato; questo aspetto viene approfondito nel paragrafo [5.4.2. Sezione autore](#).



15/05/2024 / Redazione / [da pedalare](#)

I muri delle Fiandre in bicicletta: cosa mangiare per affrontarli

Figura 40: Header di un articolo

Nella visualizzazione da desktop, il contenuto dell'articolo occupa 2/3 della schermata. A destra, all'interno di un tag <aside>, sono riportati dei contenitori con riferimenti ai tag più utilizzati, i post più visualizzati e un'immagine della mappa, similmente a come avviene sul menu mobile.

5.4.1. Tassonomie

In WordPress, una [tassonomia](#) è un meccanismo per raggruppare i post. a ogni articolo del blog sono assegnate diverse tassonomie:

- categoria: la tassonomia più importante.
Le categorie sono cinque: da pedalare, da camminare, da scoprire, da gustare, da sapere;
- tag: tassonomia che consente di associare parole chiave ai post;
- destinazione: tassonomia personalizzata che consente di associare un articolo a uno o più stati geografici. Ad esempio, articoli relativi all'Italia, Francia, o Norvegia possono essere taggati con la rispettiva destinazione;
- regione: tassonomia personalizzata che consente di associare un articolo a una o più regioni geografiche. Questo è utile per articoli che trattano aree più specifiche all'interno di un paese, come Veneto, Provenza, o Fiandre;
- città: tassonomia personalizzata che consente di associare un articolo a una o più città. Utile per articoli che offrono guide turistiche, esperienze e consigli su specifiche città come Venezia, Amsterdam, o Parigi.

Queste tassonomie rendono agile l'organizzazione, la navigazione e la ricerca all'interno del blog; per l'utente sarà più facile trovare contenuti specifici in base agli interessi personali.

5.4.2. Sezione autore

Analizzando gli articoli presenti sul vecchio blog, è emerso che diversi autori inserivano una breve biografia alla fine della pagina per firmare i propri lavori. Questi contenuti presentavano varie discrepanze visive: alcuni autori li inserivano in un'intestazione, altri in un paragrafo e altri ancora utilizzavano un collage di elementi. Per uniformare tutti gli articoli, è stato creato un box specifico posizionato in fondo alla pagina, dove l'autore può inserire la sua foto e alcune informazioni biografiche. La [Figura 41](#) ne riporta un esempio.



Figura 41: Visualizzazione descrizione dell'autore nell'articolo

5.4. ARTICOLO SINGOLO

Per raggiungere questo scopo, è stato utilizzato il [plugin](#) Advanced Custom Fields di WordPress. Questa estensione consente di aggiungere in modo immediato e intuitivo campi personalizzati ai post e alle pagine di un sito gestito con WordPress.

All'interno di un nuovo gruppo, dedicato alle informazioni aggiuntive degli articoli, è stato inserito un [post meta](#) con valore booleano per visualizzare o meno le informazioni dell'autore in fondo alla pagina. Questo campo è visualizzato all'interno dell'editor nella barra laterale, sotto forma di checkbox, come mostrato nella [Figura 42](#). In caso di selezione, verrà assegnato `true` al relativo `postmeta`, altrimenti `false`.

All'interno del template `singular.twig`, prima di procedere con l'inserimento della sezione dell'autore, viene utilizzata un'espressione condizionale per verificare il valore di questo campo.

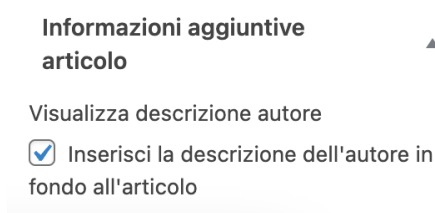


Figura 42: Inserimento descrizione dell'autore dall'editor

Ogni autore del blog può personalizzare il proprio profilo utilizzando il [backend](#) del sito, gestito da WordPress, dove può inserire l'immagine del profilo e fornire una breve biografia.

5.4.3. Il blocco indice

Il [blocco](#) di WordPress denominato Tabella dei Contenuti inserisce nel corpo di un articolo una lista di link con tutte le intestazioni presenti. Fornisce al visitatore un rapido riepilogo dei contenuti presenti e gli consente di raggiungere rapidamente qualsiasi sezione di suo interesse. Oltre ad agevolare l'utente, gli indici dei contenuti aiutano i motori di ricerca a comprendere meglio la struttura e i principali argomenti trattati nell'articolo, favorendo la classificazione e l'indicizzazione del contenuto.

Questo blocco, tuttavia, è disponibile esclusivamente su WordPress.com; per i siti di Girolibero è utilizzato invece WordPress.org; la differenza tra questi due [CMS](#) è già stata spiegata nella sezione [4.1. WordPress](#).

L'azienda ha richiesto di creare un indice che includa esclusivamente le intestazioni di secondo livello presenti nel corpo dell'articolo. L'autore inserisce un segnaposto costituito da un pattern, utilizzando l'editor Gutenberg. Un pattern è una collezione predefinita di uno o più blocchi che offre la possibilità di cambiarne il contenuto e inserirlo facilmente all'interno di articoli e pagine di un sito WordPress.

Il pattern definito per questo scopo è costituito da un blocco di codice HTML,

che inserisce un tag `<nav>` con id nella posizione desiderata dall'autore. Mediante un piccolo script realizzato con jQuery, viene creato un link per ogni intestazione di secondo livello presente nell'articolo che, successivamente, viene inserito in una lista all'interno del `<nav>` definito in precedenza.

Il risultato è riportato nella **Figura 43**:

1. New York è una città sicura per i ciclisti
2. In bici puoi raggiungere i quartieri più insoliti
3. In bicicletta sei un real americano e senti la vita vera della città
4. Pedalando ti senti parte di un film
5. Attraversare i ponti in bicicletta è incredibile

Figura 43: Indice dei contenuti di un articolo del blog

5.5. Gestione delle immagini

Il sito principale di Girolibero, che gestisce importanti quantità di immagini e video all'interno delle pagine, utilizza Cloudinary ormai da qualche anno, al contrario del blog, dove fin dai primissimi articoli le immagini venivano caricate direttamente sul server principale.

Utilizzando un servizio di gestione delle immagini esterno, si riduce significativamente il carico sul server del sito. Inoltre, uno strumento come Cloudinary consente di ottimizzare le immagini a seconda del dispositivo dell'utente, migliorando i tempi di caricamento delle pagine.

Uno degli obiettivi dello stage prevede l'integrazione di Cloudinary anche per il blog happytobehere.

Utilizzando l'apposito [plugin](#) disponibile per WordPress, è possibile trasferire automaticamente tutti i media del blog nell'archivio cloud, senza la necessità di effettuare modifiche manuali alle risorse o cambiare l'origine dei file; tuttavia prima di intraprendere questa strada sono state effettuate delle considerazioni. Molte delle immagini incluse negli articoli sono già presenti nell'archivio cloud di Cloudinary, essendo state scaricate dal sito principale di Girolibero. La configurazione automatica comporterebbe un uso eccessivo di spazio per i duplicati: l'azienda infatti, per gestire tutti i media il cui peso totale sfiora i 200GB, ha sottoscritto il piano premium di Cloudinary; nonostante lo spazio di archiviazione disponibile sia importante, una volta superata la soglia massima vengono addebitati dei costi aggiuntivi.

Pertanto, è stato deciso di iniziare con l'utilizzo di Cloudinary per gli articoli più recenti, tralasciando quelli già presenti sul blog. Le modifiche, quindi, non sono retroattive.

Se disponibile, il contenuto dell'immagine verrà fornito dal cloud o in alternativa dal relativo file presente all'interno del server.

5.5. GESTIONE DELLE IMMAGINI

Dopo aver installato e configurato il [plugin](#) è possibile procedere con l'inserimento dei media da Cloudinary.

È possibile percorrere due strade:

- selezionare un media presente sul Cloud, navigando tra le cartelle;
- caricare un media dal proprio dispositivo.

Nel primo caso, quindi, verranno utilizzati i media già presenti sul proprio account Cloudinary, nel secondo, invece, verranno prima caricati nell'apposita cartella destinata alle risorse del blog, come mostrato a [Figura 44](#).

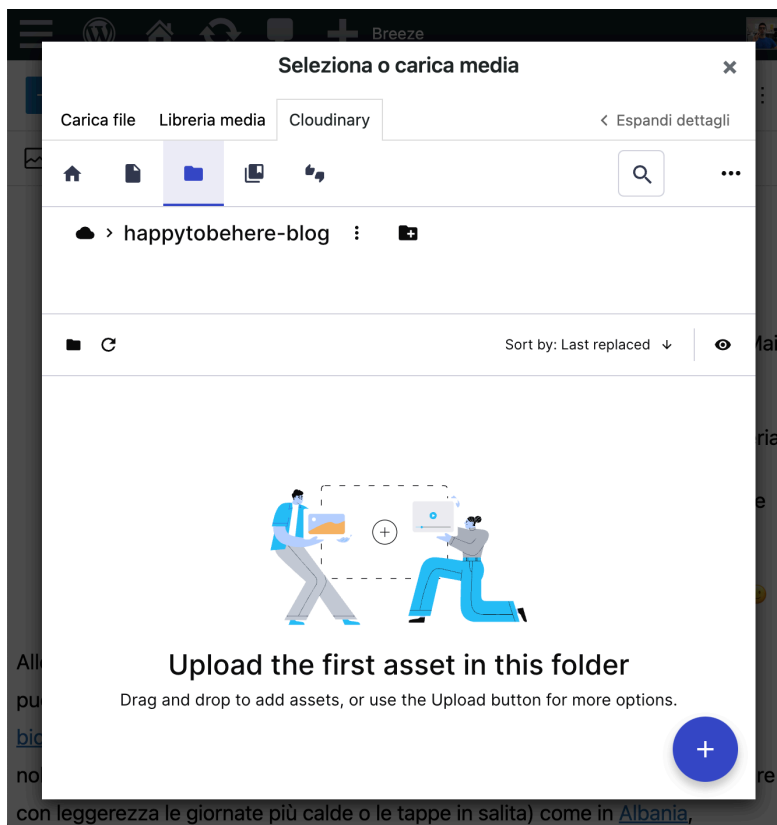


Figura 44: Caricamento di un media su Cloudinary dall'editor Gutenberg

5.6. Pagine statiche

Il blog include anche diverse pagine secondarie, ciascuna con scopi specifici legati ai contenuti o all'azienda. Come descritto precedentemente, è stato assegnato un template specifico per ognuna di queste pagine utilizzando l'editor Gutenberg:

- about us e contatti: due pagine già presenti sul vecchio blog, sono state adattate allo stile del nuovo tema. La prima descrive la filosofia e la missione dell'azienda, mentre nella seconda vengono riportati i recapiti per i clienti;

- pagina dei tag: lista alfabetica di tutti i tag utilizzati nel blog. Contribuisce all'indicizzazione del sito e consente agli utenti di accedere direttamente ai contenuti correlati a un particolare tag, facilitando la scoperta di articoli pertinenti e di approfondimenti tematici;
- sitemap: mappa completa degli articoli pubblicati sul blog, organizzata in liste che raggruppano i post per categoria. Gli elementi sono ordinati per data di pubblicazione e per ognuno viene riportato il link alla pagina e il titolo. La sitemap consente ai motori di ricerca di capire meglio il contenuto delle pagine e di indicizzare il sito più efficientemente;
- mappa degli articoli: descritta nella sezione **5.8. Mappa degli articoli**.

5.7. Newsletter

Tutti gli utenti iscritti alla [newsletter](#) di Girolibero sono salvati in un'apposita lista su Mailchimp, servizio email utilizzato dal team di marketing per comunicare periodicamente con i suoi clienti. Nella versione precedente del blog, gli utenti che si iscrivevano alla relativa newsletter, venivano salvati in una lista separata da quella del sito principale. L'azienda ha voluto rimuovere questa netta separazione tra il blog e il sito principale, incorporando le due liste.

Il form di iscrizione deve essere inserito all'interno del blog in modo armonico con gli altri contenuti e deve riprendere lo stile di quello già presente sul sito principale.

5.7.1. Funzionamento

Utilizzando il [plugin](#) Mailchimp for WordPress⁷, è stato definito un form HTML collegato alla lista principale di Girolibero. Al suo interno sono presenti i campi di input per nome, cognome, indirizzo email e accettazione dell'informativa sulla privacy.

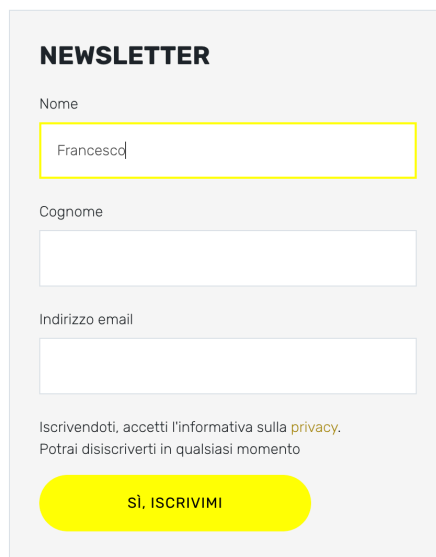
All'interno dello script principale del tema sono stati definiti dei controlli sui valori inseriti nei campi del nome, cognome ed email, al fine di prevenire l'invio del modulo in caso di compilazione errata.

5.7.2. Visualizzazione finale

È possibile visualizzare il modulo di iscrizione alla [newsletter](#) nelle pagine che riportano una lista di articoli, come la pagina principale e le pagine di archivio (autore, categoria, tag..). Il form, mostrato nella **Figura 45**, è racchiuso in un contenitore della stessa dimensione dei post della lista, è facilmente distinguibile dagli altri elementi della pagina e richiama il sito principale dell'azienda grazie al pulsante di colore giallo acceso.

⁷URL: wordpress.org/plugins/mailchimp-for-wp/

5.7. NEWSLETTER



NEWSLETTER

Nome

Cognome

Indirizzo email

Iscrivendoti, accetti l'informativa sulla [privacy](#).
Potrai disiscriverti in qualsiasi momento

SÌ, ISCRIVIMI

Figura 45: Il form di iscrizione alla newsletter del blog

5.8. Mappa degli articoli

Per rendere il blog più fruibile e coinvolgente, è stata realizzata una mappa contenente gli articoli collocati nelle aree geografiche di riferimento. Alcuni utenti possono trovare più intuitivo visualizzare gli articoli in questo modo anziché consultare una semplice lista; inoltre i visitatori che intendono pianificare un viaggio con Girolibero possono trovare facilmente gli articoli relativi alle destinazioni di loro interesse.

5.8.1. Coordinate

Per consentire la realizzazione della mappa, è necessario salvare all'interno del database le coordinate geografiche dei luoghi a cui gli articoli fanno riferimento. A questo scopo è stato creato un apposito [post meta](#) per gli articoli, consentendo agli autori di inserire e modificare le coordinate direttamente all'interno dell'editor Gutenberg, come mostra nella [Figura 46](#). Non è stato possibile usare il [plugin](#) Advanced Custom Fields, in quanto per la realizzazione di questo elemento era necessaria la versione a pagamento.

Coordinate Geografiche ▲

[Aggiungi coordinate](#)
 Inserisci le coordinate in questo formato: 12.34567890

PUNTO 1 [Rimuovi](#)

Latitudine

Longitudine

Figura 46: Inserimento coordinate di un articolo nell'editor Gutenberg

Questo postmeta viene memorizzato all'interno del database di WordPress come un array di array: il primo è il contenitore di tutte le coordinate dell'articolo, mentre il secondo è il contenitore di una singola coppia, latitudine e longitudine. La logica di questo processo è affidata a una funzione presente in `functions.php`, file del tema utilizzato per personalizzare e gestire le funzionalità di WordPress. Per consentire l'inserimento e la rimozione dei campi relativi a latitudine e longitudine, è stata utilizzata la libreria jQuery di JavaScript.

5.8.2. Preparazione dei dati

Prima di inserire gli articoli all'interno della mappa è necessario preparare correttamente i dati: MapBox accetta solo determinati tipi di file, come il formato [GeoJSON](#), per la costruzione degli strati della mappa contenenti i punti.

All'interno dell'apposito template Twig viene creato un oggetto geojson che associa le coordinate al relativo articolo. Oltre alla latitudine e alla longitudine, sono specificati per ogni articolo il titolo, l'URL dell'immagine di copertina, la categoria e l'indirizzo della pagina in cui è pubblicato.

Tutte queste informazioni sono raccolte e utilizzate per creare ogni singolo elemento della mappa.

5.8.3. Inserimento dei punti

Dopo aver iniziato la procedura di creazione della mappa, all'interno dello script viene inserito uno strato utilizzando l'oggetto [GeoJSON](#) contenente i dati geografici e identificativi degli articoli. Gli strati consentono di visualizzare i dati geografici in modo personalizzato: per ogni coppia di coordinate viene collocato sulla mappa un punto con lo stile personalizzato di Girolibero, fornito dall'[UX designer](#) e mostrato nella [Figura 47](#).



Figura 47: Punto geografico personalizzato

5.8. MAPPA DEGLI ARTICOLI

Il blog conta più di 100 articoli collocabili geograficamente all'interno della mappa. Per migliorarne la fruizione, è stato inserito uno strato aggiuntivo contenente i cluster dei punti: al variare dello zoom, i segnaposto presenti nella mappa vengono raggruppati in base alla loro posizione in contenitori che riportano il numero totale di elementi contenuti. Nella **Figura 48** è riportato un esempio.



Figura 48: Cluster di punti geografici della mappa

5.8.4. Selezione di un elemento

I dati identificativi degli articoli, presenti all'interno del file **GeoJSON** utilizzato per la costruzione dello strato principale della mappa, sono impiegati per costruire un popup quando il segnaposto associato viene selezionato. L'elemento scelto compare centrato all'interno della schermata e ospita un elemento cliccabile che riporta alla relativa pagina del blog, come mostrato a **Figura 49**. Si può quindi visualizzare il titolo, l'immagine di copertina e la categoria.



Figura 49: Popup di un elemento selezionato nella mappa

5.8.5. Visualizzazione della mappa

La mappa realizzata per il blog è mostrata nella **Figura 50**; risulta molto simile alla mappa dei viaggi organizzati presente sul sito principale di Girolibero.



Figura 50: Visualizzazione della mappa degli articoli

5.9. Il carosello viaggi

Il [carosello](#) viaggi è un contenitore presente nel sito principale di Girolibero che riporta per ogni viaggio selezionato il titolo, l'immagine di copertina, la [tipologia](#) e la [modalità](#) che lo caratterizzano.

La necessità di questo [blocco](#) personalizzato per il blog nasce dal desiderio dell'azienda di inserire all'interno del corpo degli articoli un carosello contenente i viaggi correlati. All'interno di un articolo dedicato al Veneto, per esempio, l'azienda può pubblicizzare i viaggi che fanno tappa a Padova, Verona, Venezia o Vicenza.

L'editor utilizzato per la scrittura degli articoli è Gutenberg, del [CMS](#) WordPress, descritto nella sezione [4.1. WordPress](#) di questo documento. Per inserire il carosello all'interno degli articoli, selezionando i viaggi tra quelli disponibili, è necessario creare un [plugin](#) contenente il blocco personalizzato.

5.9. IL CAROSELLO VIAGGI

5.9.1. Struttura del plugin

Viene di seguito riportata la struttura della cartella `carosello-block`, contenente il [plugin](#) del [blocco](#).

```
carosello-block/
├── src/
│   ├── block.json
│   ├── edit.js
│   ├── editor.scss
│   ├── index.js
│   ├── render.php
│   └── style.scss
├── node_modules/
│   └── ...
├── carosello-block.php
└── package.json
```

- `src/`: cartella contenente tutti i file di codice sorgente del blocco;
 - `block.json`: file contenente i [metadati](#) del blocco. Al suo interno sono custoditi dettagli identificativi del blocco come nome, descrizione e autore, ma anche informazioni sul funzionamento come le posizioni dei file di script e stile;
 - `edit.js`: file che utilizza React per renderizzare e consentire la modifica del blocco all'interno dell'editor a blocchi Gutenberg;
 - `editor.scss`: foglio di stile aggiuntivo contenente tutte le regole scss da applicare esclusivamente all'interno dell'editor a blocchi Gutenberg;
 - `index.js`: file responsabile della registrazione del blocco all'interno del client. Al suo interno è stata definita l'icona personalizzata del [carosello](#) viaggi;
 - `render.php`: file che si occupa del render del blocco che verrà inserito nel [frontend](#)
 - `style.scss`: foglio di stile contenente tutte le regole scss da applicare sia all'editor a blocchi Gutenberg che al frontend;
- `node_modules/`: cartella contenente tutte le librerie esterne utilizzate dal blocco;
- `carosello-block.php`: file utilizzato per registrare il blocco all'interno del server WordPress;
- `package.json`: file di configurazione di progetti che utilizzano Node.js, come i blocchi Gutenberg. Al suo interno vengono definite le informazioni relative alle dipendenze del blocco.

5.9.2. Collegamento all'API viaggi di Girolibero

Prima di procedere con la creazione del **blocco**, è necessario recuperare le informazioni legate ai viaggi organizzati di Girolibero, fornite da una delle **API** presenti del sito principale. I testi dei viaggi sono disponibili in lingua italiana, inglese e tedesca: pertanto, è necessario fornire la lingua desiderata all'interno del parametro GET della richiesta.

Se la richiesta viene effettuata correttamente, l'API restituisce tutti i dati dei viaggi strutturati in un unico file in formato **JSON (JavaScript Object Notation)** [9]: queste informazioni saranno utilizzate per renderizzare in tempo reale il **carosello** all'interno dell'editor e sul sito.

5.9.3. Creazione del blocco wordpress

Seguendo la guida [16] ufficiale di WordPress, è stato creato il **plugin** del **blocco**. Nel file `edit.js` è presente la logica di funzionamento e visualizzazione del blocco all'interno dell'editor, realizzata utilizzando la libreria React di JavaScript. È divisa in due parti: la prima è dedicata alla configurazione del pannello di controllo, menu visualizzato nell'apposita sezione presente sulla destra dell'editor Gutenberg; la seconda riporta i contenuti da visualizzare all'interno del blocco in base ai valori inseriti nel pannello di controllo.

L'elemento principale presente all'interno del pannello di controllo del blocco è un `FormTokenField`: si tratta di un componente Gutenberg che consente l'inserimento di elementi testuali definiti dall'utente all'interno di una lista. a ogni inserimento vengono presentati tra le opzioni suggerite i valori **codweb** dei viaggi di Girolibero; nel caso in cui un valore immesso dall'utente non dovesse corrispondere a un codweb esistente, questo non verrà inserito nella lista.

Quando un codweb viene aggiunto correttamente, verrà renderizzato il viaggio corrispondente all'interno del blocco. Questo processo è affidato alla seconda parte dello script realizzato con React, che recupera le informazioni correlate al codweb e le riporta nella struttura HTML che viene visualizzata a schermo. La **Figura 51** ne riporta un esempio.

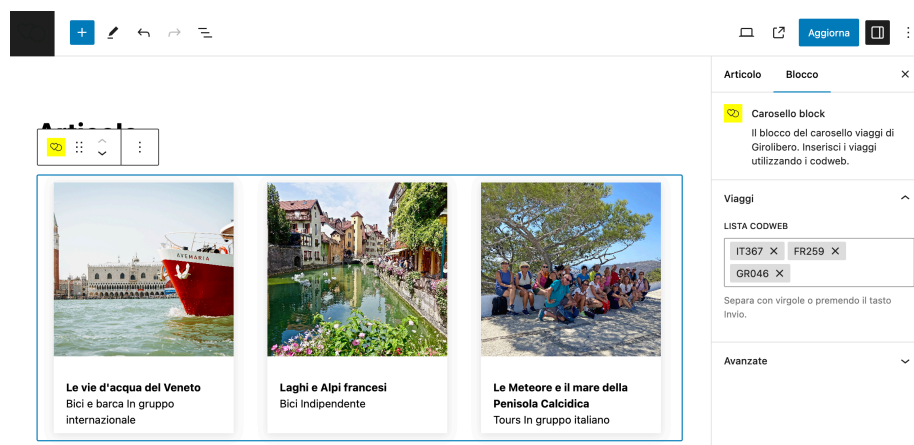


Figura 51: Il blocco del carosello viaggi nell'editor Gutenberg

5.9. IL CAROSELLO VIAGGI

5.9.4. Visualizzazione nell'articolo

Per visualizzare il **blocco** nel corpo dell'articolo all'interno del blog è necessario renderizzare i valori inseriti nell'editor. Questo processo è affidato al file `render.php` del **plugin**, ed è molto simile alla seconda parte del file `edit.js`: per ogni **codweb** inserito all'interno dell'editor, viene creato un elemento HTML contenente le informazioni principali del viaggio, che vengono riportate all'interno di un **carosello**, mostrato a **Figura 52**.

Lo script utilizzato per il funzionamento dinamico del carosello è lo stesso presente nel sito principale di Girolibero.

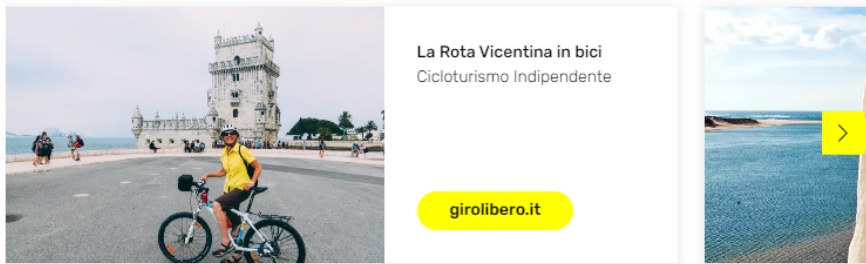


Figura 52: Il carosello viaggi nel corpo di un articolo

Capitolo 6.

Revisione

Questo capitolo riassume gli interventi di revisione effettuati sugli articoli già pubblicati nel vecchio blog e le linee guida rilasciate per gli autori. Segue un'analisi dell'accessibilità del sito.

Dopo aver creato il nuovo tema, è apparso evidente come gli articoli già presenti all'interno del blog non seguissero una linea comune per struttura e impaginazione. Prima del rilascio del tema quindi, è stato necessario adattare le nuove modifiche ai contenuti del vecchio blog e correggere manualmente i post utilizzando l'editor Gutenberg.

6.1. Interventi

Le modifiche e gli interventi apportati sugli articoli del blog sono avvenuti in un ambiente di staging, ovvero un server interno all'azienda dove vengono testate le nuove funzionalità prima di essere rilasciate al pubblico.

6.1.1. Errori di formattazione

Durante la revisione degli articoli sono stati riscontrati diversi errori di formattazione del testo, in parte dovuti all'utilizzo del vecchio editor di WordPress, che lasciava troppa libertà agli scrittori poco esperti di linguaggio di markup HTML.

Alcuni autori strutturavano gli articoli seguendo esclusivamente lo stile degli elementi inseriti; nonostante fossero graficamente piacevoli, le scelte effettuate erano semanticamente errate. Sono di seguito elencati i principali errori riscontrati:

- mancato rispetto delle gerarchie di titoli (H1, H2, H3): per scelta stilistica, alcuni articoli non seguivano linearmente la struttura imposta dalla gerarchia delle intestazioni, ma presentavano salti di un livello, come, per esempio, H1 seguito da H3. Questa pratica non è corretta perché rompe la coerenza strutturale del documento. Inoltre crea confusione nella sua interpretazione sia agli utenti che agli strumenti di accessibilità come gli *screen reader*^[9] quest'ultimi possono avere difficoltà a navigare e interpretare correttamente la struttura del testo quando le gerarchie non sono rispettate. Impatta anche sui motori di ricerca, influenzando negativamente sulla [SEO](#) del sito;
- impiego di un titolo con funzioni di paragrafo: una frase, considerata rilevante dall'autore veniva impostata come intestazione, esclusivamente per poterla visualizzare in grassetto e con una grandezza maggiore. Utilizzare i titoli per enfatizzare singole frasi non è consigliabile perché le intestazioni sono

6.1. INTERVENTI

destinate a suddividere il contenuto del documento in sezioni logiche; questa prassi errata compromette la coerenza strutturale come spiegato sopra, impattando sia sugli strumenti di accessibilità che sul ranking SEO;

- impiego di un titolo con funzioni di immagine: un'immagine veniva inserita all'interno di un'intestazione al fine di ridurre la sua larghezza (es. dentro un H2). Le intestazioni devono essere usate esclusivamente per organizzare e strutturare il contenuto; inserire immagini nei titoli compromette la semantica del documento, con il rischio di causare problemi di visualizzazione su diversi dispositivi. Le immagini devono essere ridimensionate usando l'apposito comando nell'editor.

6.2. Linee guida per la formattazione

Per garantire una coerenza futura nei contenuti pubblicati, sono state comunicate delle linee guida di formattazione agli autori del blog di Girolibero.

La struttura è stata definita con Elena Riatti, responsabile del team marketing e tutor, e Caterina Romio, UX designer.

Di seguito vengono riportate le linee guida.

6.2.1. Introduzione

Ogni articolo deve includere una breve introduzione che ne riassume il contenuto e attiri l'interesse del lettore. Questo paragrafo sarà inserito appena dopo l'header e, a discrezione dell'autore, potrà essere seguito dall'indice dei contenuti dell'articolo, entrambi già descritti nella sezione [5.4 Articolo singolo](#);

6.2.2. Contenuto

Nel corpo degli articoli è necessario prestare attenzione all'utilizzo corretto di intestazioni, paragrafi ed immagini.

Intestazioni:

- l'utilizzo delle gerarchie di titoli (H1, H2, H3) deve essere appropriato. L'intestazione di primo livello sarà presente esclusivamente nel titolo dell'articolo, all'interno dell'header; sarà prediletto l'uso del H2, intestazione indicizzata dalla lista dei contenuti, e sarà evitato il più possibile l'utilizzo degli H3, poichè difficilmente gli articoli del blog necessitano di tale profondità di contenuti;
- i titoli non devono essere preceduti da pallini rotondi (es. • Titolo 1) tipici degli elenchi puntati; tuttavia è consentito organizzarli in una lista ordinata inserendo antecedentemente un numero (es. 1. Titolo);
- non deve essere inserita un'intestazione subito dopo un'intestazione priva di contenuti, come nel seguente esempio:
 - H1 (titolo): Scopri i castelli della Loira con Girolibero
 - P: *sommario...*
 - H2: Scopri i castelli della Loira
 - H3: Castello n.1

6.2. LINEE GUIDA PER LA FORMATTAZIONE

- ▶ P: *contenuto...*
- ▶ H3: Castello n.2
- ▶ P: *contenuto...*

L'utilizzo del titolo H2, in questo caso, è superfluo: può essere rimosso e, di conseguenza, le intestazioni di terzo livello diventano H2.

Avere contenuto tra le intestazioni aiuta a mantenere un layout visivamente piacevole e coerente; titoli che introducono altri titoli sono vietati.

Paragrafi:

- le decorazioni testuali come il sottolineato sono vietate, ad eccezione di grassetto e italic;
- l'uso del grassetto va moderato per non rischiare di perdere l'effetto di enfaticizzazione e di rendere il testo meno chiaro e più difficile da leggere. Un testo con un uso equilibrato di grassetto appare più professionale e curato; un uso eccessivo può far sembrare il testo disorganizzato e amatoriale;
- l'italic va limitato alle citazioni;
- i link devono essere riportati con un apposito collegamento ipertestuale, senza inserire l'URL per esteso;
- i link che rimandano ai viaggi specifici organizzati da Girolibero devono essere inseriti all'interno di un [carosello](#) tramite l'apposito [blocco](#) offerto dall'editor Gutenberg.

Immagini:

- devono essere inserite necessariamente dalla libreria di Cloudinary;
- devono essere dotate di un attributo alt valido: si tratta di un'alternativa testuale che non viene mostrata a schermo, ma viene letta dagli [screen reader](#) e altri strumenti assistivi per le pagine web. Non penalizza gli utenti che hanno problemi di vista e rappresenta un requisito necessario per il livello di [WCAG A](#) (livello accessibilità base); inoltre migliora l'indicizzazione del blog;
- non vanno inserite immediatamente dopo un'intestazione (H1, H2, H3), ma vanno posizionate sotto a un paragrafo;
- devono essere allineate a sinistra se presentano una larghezza inferiore al corpo dell'articolo.

6.2.3. Conclusione

La conclusione di un articolo deve essere realizzata mediante un paragrafo non breve, seguito se possibile da una o due immagini.

6.2. LINEE GUIDA PER LA FORMATTAZIONE

6.2.4. Altre direttive

È bene specificare il contenuto del tag meta description che, pur non influenzando direttamente sulla [SEO](#), è un elemento importante nei motori di ricerca. Fornisce, infatti, un'anteprima del contenuto della pagina tra i risultati di una ricerca web: una descrizione concisa e accurata può attirare l'attenzione degli utenti e aumentare il tasso di click.

6.3. Accessibilità

L'accessibilità nell'ambito web è la capacità di un sito di essere utilizzabile da un ampio pubblico, incluse persone con disabilità fisiche, visive, uditive e cognitive. Gli utenti, indipendentemente dalle loro capacità, hanno il diritto di navigare in modo efficace e comprendere i contenuti di un sito senza impedimenti.

Durante lo sviluppo di un sito, è importante implementare fin da subito le principali nozioni di accessibilità, ormai consolidate nelle best practices e che non richiedono particolare lavoro aggiuntivo.

L'obiettivo facoltativo dello stage, FA1, prevedeva che venisse svolta l'analisi dell'accessibilità del blog appena realizzato e, se necessario, che venissero applicate eventuali correzioni per adeguarla al livello [WCAG\[17\]](#) A (requisito R-06-Q-3).

L'obiettivo non è stato realizzato nella sua interezza: sebbene sia stata eseguita l'analisi dell'accessibilità del blog utilizzando Total Validator come validatore, diversi criteri del primo livello WCAG non sono stati soddisfatti. Questo è dovuto principalmente alle scelte stilistiche dell'[UX](#) designer e, in secondo luogo, ai contenuti degli articoli già presenti sul blog.

Viene di seguito riportata l'analisi dell'accessibilità del blog.

6.3.1. Navigazione

Un sito web accessibile deve fornire funzionalità di supporto all'utente durante la sua navigazione.

Il blog è completamente navigabile da tastiera, utilizzando il tasto tab e le frecce di navigazione e non è presente alcun impedimento per il suo utilizzo. Anche i link sono selezionabili usando l'apposito comando da tastiera; questo soddisfa il criterio di successo 2.1.1 relativo all'uso della tastiera di [WCAG2.1](#):

Tutte le funzionalità del contenuto sono utilizzabili tramite un'interfaccia di tastiera senza richiedere tempi specifici per la pressione dei singoli tasti...

Un altro criterio di successo rispettato è il 2.4.1 - Salto di blocchi:

È disponibile un meccanismo per saltare i blocchi di contenuto che si ripetono su più pagine Web.

È stato messo in pratica usando un link, offrendo la possibilità di saltare tutte

le voci presenti nel menu per raggiungere con facilità il contenuto principale della pagina.

Grazie all'uso dell'attributo `aria-label`, è possibile assegnare agli elementi interattivi privi di testo, come i pulsanti torna-su o apri-menu, un nome che ne descriva le finalità. Questo accorgimento consente anche agli strumenti assistivi di comprendere il loro scopo, come affermato dal Criterio di successo 1.1.1 - Contenuti non testuali e ribadito dal Criterio di successo 4.1.2 - Nome, ruolo, valore:

...se il contenuto non testuale è un controllo o accetta l'input degli utenti, allora ha un nome che ne descrive la finalità. (Fare riferimento al criterio di successo 4.1.2 per requisiti aggiuntivi per controlli e contenuti che accettano l'input dell'utente). ...

6.3.2. Immagini

Uno dei punti più critici del blog è legato alla gestione delle immagini. Il criterio di successo 1.1.1 di [WCAG2.1](#) stabilisce:

Tutti i contenuti non testuali presentati all'utente hanno un'alternativa testuale equivalente che serve allo stesso scopo, ...

Per conformarsi a questa regola, le immagini devono riportare un attributo alt valido: si tratta di un testo di breve lunghezza che viene utilizzato dagli [screen reader](#) per descrivere i media, come le immagini, agli utenti non vedenti o ipovedenti. Inoltre è utilizzato dai motori di ricerca per comprendere il contenuto delle immagini e classificare i siti.

La mancanza di questo attributo all'interno della quasi totalità delle immagini del blog, rappresenta una significativa barriera all'accessibilità, compromettendo l'esperienza degli utenti che utilizzano tecnologie assistive. Influisce negativamente sulla [SEO](#), poiché i motori di ricerca non possono interpretare correttamente il contenuto delle immagini.

Come riportato nella sezione [6.2.2. Contenuto](#), l'inserimento dell'attributo alt è di fondamentale importanza e non è trascurabile negli articoli già pubblicati.

6.3.3. Link

I collegamenti inseriti all'interno delle pagine devono essere chiaramente distinguibili dagli altri elementi presenti.

È importante ricordare il Criterio di successo 1.4.1 delle linee guida [WCAG2.1](#) livello A relativo all'uso del colore:

Il colore non deve essere utilizzato come unica modalità visiva per rappresentare informazioni, indicare azioni, richiedere risposte o come elemento di distinzione visiva.

Sono diversi gli elementi del blog che non rispettano questo requisito:

6.3. ACCESSIBILITÀ

- le voci del menu superiore: nonostante si distinguano dal testo principale perchè in maiuscolo, presentano lo stesso stile e colore delle voci presenti nella schermata laterale di un singolo articolo, ovvero elementi non cliccabili;
- i link alle pagine autore: presentano lo stesso stile e colore del testo comune, pertanto non sono graficamente distinguibili da nessun utente;
- i link “vedi tutti”: collegamenti a un’altra pagina del blog presenti nella schermata laterale di un singolo articolo, che non si distinguono stilisticamente dai termini in grassetto utilizzati nel corpo di un post.

In genere per soddisfare questo requisito del primo livello basta aggiungere un indicatore visivo che consenta di distinguere i collegamenti dagli altri elementi delle pagine, come una sottolineatura oppure un’icona. È fortemente consigliato, inoltre, adottare i seguenti accorgimenti per migliorare l’esperienza utente:

- focus visibile: in presenza di tanti link aiuta il visitatore a riconoscere il link che ha selezionato dalla tastiera;
- colore diverso per link visitati: si ricorda all’utente quali pagine ha già visitato.

6.3.4. Form

Un form, o modulo, è un contenitore HTML che racchiude dei campi che possono essere compilati dagli utenti e inviati a un server. All’interno del blog ne sono presenti due: uno per la ricerca degli articoli e uno per l’iscrizione alla [newsletter](#).

Due componenti di markup, spesso utilizzati nei moduli, sono `input` e `label`:

- `input`: elemento utilizzato per consentire l’inserimento di informazioni da parte degli utenti;
- `label`: elemento che fornisce una descrizione o un’etichetta a un campo `input`.

In questi moduli è fondamentale associare ogni `input` a una `label` per garantire anche agli utenti che utilizzano tecnologie assistive di comprendere chiaramente l’uso e la funzione; gli [screen reader](#), infatti, annunciano il contenuto della `label` quando l’utente interagisce con l’`input` associato.

Inoltre, l’adozione di questa pratica dà la possibilità ai visitatori di cliccare sulla `label` per attivare il relativo campo di `input`, migliorando l’usabilità del modulo. Questa associazione è necessaria per soddisfare uno dei requisiti del [WCAG A](#), Criterio di successo 1.3.1:

Le informazioni, la struttura e le correlazioni trasmesse dalla presentazione possono essere determinate programmaticamente oppure sono disponibili tramite testo.

6.3.5. Colori e contrasti

Il primo livello delle [WCAG](#), il livello A, non specifica alcun requisito di contrasto dei colori per il testo; queste indicazioni sono presenti a partire dal livello

AA. Nonostante il livello desiderato dall'azienda fosse il primo, è stata comunque eseguita l'analisi dei colori principali utilizzati all'interno del blog. Durante la progettazione, l'UX designer ha preferito favorire il design, senza effettuare considerazioni sull'accessibilità. Questa scelta non è inclusiva e non segue le best practices della programmazione web.

Vengono di seguito riportate le linee guida WCAG2.1 livello AA e AAA relative all'uso dei colori nel testo.











Livello AA, Criterio di successo 1.4.3:

- il rapporto di contrasto tra il testo e il suo sfondo deve essere almeno di 4.5:1 per il testo normale;
- per il testo grande (definito come testo di almeno 18 punti di dimensione o 14 punti se grassetto), il rapporto di contrasto deve essere almeno di 3:1.

Livello AAA, Criterio di successo 1.4.6:

- il rapporto di contrasto tra il testo e il suo sfondo deve essere almeno di 7:1 per il testo normale;
- per il testo grande, il rapporto di contrasto deve essere almeno di 4.5:1.

Tramite uno strumento di analisi dei contrasti, sono stati verificati i colori utilizzati per l'interfaccia grafica del blog e riportati nella **Tabella 7**.

Descrizione	Testo	Sfondo	Con- trasto	WCAG
Colore principale del testo, utilizzato in: titoli, estratti, corpo dell'articolo, menu superiore.	 #000000	 #FFFFFF	21:1	AAA
Colori che contraddistinguono le categorie: testo in grassetto su sfondo con bordi arrotondati.	 #FFFFFF	 #E75E59	3.41:1	Nessuno
Colori che contraddistinguono i tag: testo in corsivo su sfondo con bordi arrotondati.	 #686767	 #FFFFFF	5.63:1	AA
Colore dei link "altro" e "vedi tutti", presenti in: menu mobile, aside degli articoli, estratti.	 #E75E59	 #FFFFFF	3.41:1	Nessuno
Colore dei link presenti nel corpo degli articoli.	 #AB9D70	 #FFFFFF	2.69:1	Nessuno

6.3. ACCESSIBILITÀ









Descrizione	Testo	Sfondo	Con- trasto	WCAG
Colori del banner contenente il collegamento al sito principale di Girolibero.	 #000000	 #FFFF00	19.55:1	AAA

Tabella 7: Contrasti dei colori dell'interfaccia grafica del blog

Tre elementi della lista non raggiungono il livello AA. Senza impattare troppo sullo stile, è possibile adattare questi colori al secondo livello:

- per lo sfondo delle categorie: passaggio dal colore #E75E59  a #E02F29  per un contrasto pari a 4.55:1;
- per il testo dei link “altro” e “vedi tutti”: passaggio dal colore #E75E59  a #E02F29  per un contrasto pari a 4.55:1;
- per il testo dei link presenti nel corpo degli articoli: passaggio dal colore: #AB9D70  a #7C774B  per un contrasto pari a 4.56:1.

Capitolo 7.

Conclusioni

Il capitolo conclusivo riassume lo stato degli obiettivi di stage e dei requisiti concordati con l'azienda. Seguono delle considerazioni personali sull'esperienza di stage.

7.1. Consuntivo

Lo scopo principale dello stage era quello di creare un nuovo tema di WordPress per il blog aziendale, collegando i dati dei viaggi presenti sul sito principale e utilizzando Cloudinary per il recupero delle immagini.

Terminato lo sviluppo del tema, il lavoro è proseguito con la correzione degli articoli già pubblicati sul blog e il loro adattamento alle linee guida per la formattazione delineate con il team. Come spiegato nel capitolo precedente, questa fase di revisione non è stata completata interamente a causa dell'elevato numero di articoli già presenti.

Lo stage è iniziato in data 29/04/2024 e si è concluso il 24/06/2024, per un totale di 320 ore di lavoro.

7.2. Raggiungimento degli obiettivi

La **Tabella 8** riassume gli obiettivi di stage e i relativi esiti.

Codice	Descrizione	Esito
OB1	Realizzazione backend e frontend	Raggiunto
OB2	Collegamento dei dati provenienti dal sito principale	Raggiunto
OB3	Integrazione del blog con Cloudinary	Raggiunto
DE1	Realizzazione form di iscrizione alla newsletter	Raggiunto
DE2	Tracciamenti Google Analytics	Non raggiunto
FA1	Analisi accessibilità del blog ed eventuale adeguamento	Non raggiunto

Tabella 8: Raggiungimento obiettivi fissati per lo stage.

7.2. RAGGIUNGIMENTO DEGLI OBIETTIVI

Come si può notare, tutti gli obiettivi sono stati raggiunti tranne DE2 e FA1 per le ragioni di seguito esplicitate.

- DE2: non raggiunto in quanto il team **IT** ha richiesto che il tracciamento di Google Analytics fosse aggiunto solamente una volta pubblicato online il nuovo tema del blog. Questo non è avvenuto entro il termine finale della mia esperienza di stage, poichè l'azienda ha preferito procedere con la sistemazione di tutti gli articoli già pubblicati secondo le nuove linee guida per la formattazione.
- FA1: raggiunto parzialmente. Sebbene sia stata eseguita un'analisi generale dell'accessibilità del blog, non è stato possibile adeguare il sito al livello **WCAG A** a causa di alcune scelte effettuate dall'**UX** designer, già discusse nel capitolo precedente.

7.3. Stato dei requisiti

Le tabelle **9** e **10** mostrano l'esito dei requisiti funzionali, di qualità e di vincolo, fissati durante l'analisi dei requisiti.

Codice	Stato
R-1-F-1	Soddisfatto
R-2-F-1	Soddisfatto
R-3-F-1	Soddisfatto
R-4-F-1	Soddisfatto
R-5-F-1	Soddisfatto
R-6-F-1	Soddisfatto
R-7-F-1	Soddisfatto
R-8-F-1	Soddisfatto
R-9-F-1	Soddisfatto
R-10-F-1	Soddisfatto
R-11-F-1	Soddisfatto
R-12-F-1	Soddisfatto
R-13-F-1	Soddisfatto
R-14-F-1	Soddisfatto
R-15-F-1	Soddisfatto
R-16-F-1	Soddisfatto
R-17-F-1	Soddisfatto
R-18-F-1	Soddisfatto
R-19-F-1	Soddisfatto
R-20-F-1	Soddisfatto
R-21-F-1	Soddisfatto

7.3. STATO DEI REQUISITI

Codice	Stato
R-22-F-1	Soddisfatto
R-23-F-1	Soddisfatto
R-24-F-1	Soddisfatto
R-25-F-1	Soddisfatto
R-26-F-1	Soddisfatto
R-27-F-1	Soddisfatto
R-28-F-1	Soddisfatto
R-29-F-1	Soddisfatto
R-30-F-1	Soddisfatto
R-31-F-1	Soddisfatto
R-32-F-1	Soddisfatto
R-33-F-1	Soddisfatto
R-34-F-1	Soddisfatto
R-35-F-1	Soddisfatto
R-36-F-1	Soddisfatto
R-37-F-1	Soddisfatto
R-38-F-1	Soddisfatto
R-39-F-1	Soddisfatto
R-40-F-1	Soddisfatto
R-41-F-1	Soddisfatto
R-42-F-1	Soddisfatto
R-43-F-1	Soddisfatto
R-44-F-1	Soddisfatto
R-45-F-1	Soddisfatto
R-46-F-1	Soddisfatto
R-47-F-1	Soddisfatto
R-48-F-2	Soddisfatto
R-49-F-2	Soddisfatto
R-50-F-2	Soddisfatto
R-51-F-2	Soddisfatto
R-52-F-2	Soddisfatto
R-53-F-2	Soddisfatto
R-54-F-2	Soddisfatto
R-55-F-2	Soddisfatto
R-56-F-2	Soddisfatto

7.3. STATO DEI REQUISITI

Codice	Stato
R-57-F-3	Soddisfatto
R-58-F-3	Soddisfatto
R-01-Q-1	Soddisfatto
R-02-Q-1	Soddisfatto
R-03-Q-1	Soddisfatto
R-04-Q-1	Soddisfatto
R-05-Q-1	Soddisfatto
R-06-Q-3	Non soddisfatto
R-07-Q-2	Non soddisfatto
R-08-Q-2	Non soddisfatto
R-09-Q-2	Non soddisfatto
R-10-Q-1	Soddisfatto
R-1-V-1	Soddisfatto
R-2-V-1	Soddisfatto
R-3-V-1	Soddisfatto
R-4-V-1	Soddisfatto
R-5-V-2	Soddisfatto
R-6-V-1	Soddisfatto
R-7-V-1	Soddisfatto
R-8-V-2	Soddisfatto
R-9-V-2	Soddisfatto
R-10-V-1	Soddisfatto
R-11-V-1	Soddisfatto

Tabella 9: Stato finale dei requisiti

Tipologia	Obbligatori		Desiderabili		Opzionali	
	SÌ	NO	SÌ	NO	SÌ	NO
Funzionali	47	0	9	0	2	0
Qualità	6	0	0	3	0	1
Vincolo	8	0	3	0	0	0
Totale	61	0	12	3	2	1

Tabella 10: Riepilogo finale dei requisiti raggiunti

7.4. Conoscenze acquisite

Le conoscenze maturate prima di questa esperienza sono state un buon punto di partenza per acquisirne di nuove.

Questo progetto di stage mi ha permesso di conoscere WordPress, uno dei [CMS](#) più utilizzati al mondo. È stato particolarmente interessante scoprire come vengono gestiti i contenuti di numerosi siti web presenti online. La creazione di un [plugin](#) personalizzato per l'editor Gutenberg è stato un valore aggiunto e mi ha fatto apprezzare le potenzialità di questo strumento, che consente un'ampia personalizzazione dei contenuti.

Twig è un potente motore di template per le pagine HTML che separa la logica di presentazione e il markup HTML dal codice di applicazione PHP, migliorando così la manutenzione e la leggibilità del codice. Grazie alle sue funzionalità avanzate, come l'eredità tra template, Twig consente di creare layout puliti, riutilizzabili ed efficienti. Questo strumento rappresenterà una valida opzione per i miei progetti futuri legati al mondo web.

JQuery mi ha permesso di capire come rendere interattive e reattive le pagine web con poche linee di codice. Ho avuto l'opportunità di conoscere le chiamate [AJAX](#), strumento utilizzato per comunicare con il server senza dover ricaricare l'intera pagina; argomento non affrontato nel corso del terzo anno Tecnologie Web. Questa libreria è ampiamente utilizzata nel mondo web e averla sperimentata mi ha dato soddisfazione.

Anche Sass è stata una scoperta interessante. Estensione del linguaggio CSS, introduce una serie di funzionalità avanzate che rendono la scrittura dei fogli di stile più potente e flessibile, migliorando l'organizzazione e la leggibilità del codice. Sebbene sia stata sfruttata solamente una piccola parte delle potenzialità che offre, impiegherò sicuramente Sass in progetti personali futuri.

7.5. Valutazione personale

Questo stage ha rappresentato la mia primissima esperienza in azienda nel settore [IT](#). Le conoscenze pregresse sono state un ottimo punto di partenza per questa attività formativa: le solide basi fornite da corsi come Tecnologie Web, mi hanno consentito di svolgere lo stage senza particolari difficoltà e di approfondire uno dei miei settori preferiti del vasto mondo dell'informatica, il web.

Questa esperienza mi ha confermato quanto sia importante saper padroneggiare gli strumenti di versionamento del codice, ampiamente utilizzati nel mondo reale. Se potessi rivere il mio piano di studio, seguirei molto volentieri il corso opzionale del secondo anno Metodi e Tecnologie per lo Sviluppo Software, che tratta queste tematiche approfonditamente. Le conoscenze possedute di questo strumento sono comunque risultate sufficienti per lo svolgimento del progetto

7.5. VALUTAZIONE PERSONALE

di stage, tuttavia, considerata la loro importanza, desidero consolidarle nell'immediato futuro.

Nicola Cecchetto, il tutor che mi ha seguito di più, è sempre stato disponibile ad aiutarmi qualora ne avessi avuto bisogno, in particolare nella fase introduttiva alle tecnologie e durante lo sviluppo. Ammiro, in particolare, la sua abilità di problem solving, di cui farò tesoro e porterò una piccola parte nel mio bagaglio personale.

L'ambiente accogliente e collaborativo in cui sono stato inserito mi ha consentito di svolgere questa primissima esperienza in azienda sempre con grande entusiasmo e interesse. Il personale svolge il proprio lavoro con passione e dedizione e mi ha fornito tutto il supporto necessario.

Sono consapevole che passerà del tempo prima di vedere pubblicato, poiché la revisione integrale degli articoli è piuttosto lunga; sapere, però, di aver realizzato un prodotto che verrà effettivamente utilizzato mi gratifica e mi rende soddisfatto.

A conclusione di queste riflessioni, valuto la mia esperienza presso l'azienda Girolibero costruttiva e molto positiva.

Glossario

Backend: l'insieme delle applicazioni e dei programmi essenziali al funzionamento del sistema con cui l'utente non interagisce direttamente. [2](#), [3](#), [4](#), [45](#), [48](#), [62](#), [81](#)

Blocco: singola unità di contenuto nell'editor Gutenberg di WordPress. Consente l'inserimento di componenti base come testo e immagini, ma anche di componenti più complesse come pulsanti di pagamento e calendari. Ogni blocco è autonomo e può essere aggiunto, modificato e configurato indipendentemente dagli altri blocchi. [2](#), [15](#), [16](#), [17](#), [20](#), [21](#), [38](#), [39](#), [40](#), [50](#), [51](#), [62](#), [69](#), [70](#), [71](#), [72](#), [75](#)

Carosello: elemento di UI comunemente utilizzato all'interno di siti web e applicazioni, che consente agli utenti di scorrere una lista di elementi (immagini, testi...) in uno spazio limitato. [16](#), [17](#), [20](#), [21](#), [38](#), [39](#), [40](#), [69](#), [70](#), [72](#), [75](#)

Codweb: codice alfanumerico univoco che identifica un viaggio organizzato di Girolibero. [20](#), [21](#), [39](#), [40](#), [71](#), [72](#)

Framework: architettura logica di supporto sulla quale un prodotto software può essere realizzato. Nell'ambito della programmazione web supporta lo sviluppo di siti dinamici, applicazioni e servizi web. [4](#), [49](#)

Frontend: tutte le componenti di un sistema con cui l'utente può interagire direttamente. [2](#), [3](#), [4](#), [45](#), [47](#), [70](#), [81](#)

Geojson: basato sul formato json, è uno standard aperto progettato per rappresentare semplici elementi geografici, come punti e linee, insieme ai loro attributi non spaziali, come nome e descrizione. [52](#), [67](#), [68](#)

Metadati: informazioni descrittive che forniscono contesto e dettagli su un determinato dato o risorsa. [11](#), [12](#), [14](#), [48](#), [55](#), [70](#)

Modalità: organizzazione di un viaggio Girolibero: in gruppo, indipendente, per famiglie. [20](#), [39](#), [69](#)

Newsletter: strumento di marketing utilizzato dalle aziende per mantenere un rapporto periodico e costante con i propri clienti grazie all'invio di un'email. Tramite un apposito form di iscrizione, un utente può dare il consenso alla sua registrazione in una lista newsletter. [3](#), [10](#), [33](#), [34](#), [35](#), [36](#), [42](#), [43](#), [59](#), [65](#), [78](#), [81](#)

ACRONIMI

Plugin: componente software progettato per estendere delle funzionalità di un prodotto software già esistente. [47](#), [56](#), [62](#), [63](#), [64](#), [65](#), [66](#), [69](#), [70](#), [71](#), [72](#), [85](#)

Post Meta: campo personalizzato in cui sono custoditi dati aggiuntivi associati a un post o una pagina di WordPress. Questi dati sono memorizzati nel database e possono includere informazioni di qualsiasi tipo. [60](#), [62](#), [66](#)

Screen Reader: tecnologia ad hoc realizzata per rendere accessibili e usabili prodotti informatici alle persone con disabilità. Identifica e interpreta il testo mostrato sullo schermo di un dispositivo e lo riporta come output di un assistente vocale o uno schermo braille. [73](#), [75](#), [77](#), [78](#)

Tassonomia: modalità di raggruppare i post e altri tipi di contenuti all'interno di WordPress. Le tassonomie sono utilizzate per classificare e organizzare i contenuti in categorie significative, permettendo una migliore navigazione e ricerca sul sito. WordPress offre due tassonomie predefinite: categorie e tag; tuttavia è possibile creare tassonomie personalizzate a seconda delle proprie esigenze. [55](#), [59](#), [61](#)

Tipologia: attività svolta durante un viaggio di Girolibero: in bici, in barca e trekking, tour. [19](#), [39](#), [69](#)

Acronimi

AJAX – Asynchronous JavaScript and XML: tecniche di sviluppo software utilizzate per la realizzazione di siti web interattivi. I dati vengono scambiati in background tra browser e server, consentendo l'aggiornamento dinamico di una pagina web senza necessariamente ricaricarla. [50](#), [59](#), [85](#)

API – Application Programming Interface: insieme di regole e protocolli che consentono a diversi software o componenti di comunicare tra loro. Un API definisce i metodi e le strutture dati che possono essere utilizzati per interagire con un'applicazione o un servizio, consentendo a diverse applicazioni di scambiare informazioni e funzionalità in modo standardizzato. [10](#), [17](#), [20](#), [21](#), [29](#), [71](#)

CMS – Content Management System: strumento installato su un server web utilizzato dagli amministratori di un sito per gestirne il contenuto tramite un'apposita UI. [2](#), [10](#), [45](#), [47](#), [62](#), [69](#), [85](#)

DAM – Digital Asset Management: gestore di media digitali, come immagini e video, che consente di distribuire i suoi contenuti su diversi canali di comunicazione, come siti web e applicazioni. [4](#)

DOM – Document Object Model: rappresentazione strutturata di un documento HTML che consente ad alcuni linguaggi di programmazione, come JavaScript, di accedere e manipolare il contenuto e la struttura di una pagina web. [50](#)

IT – Information Technology: termine generico che comprende tutte le tecnologie utilizzate per l'elaborazione, la memorizzazione e la trasmissione delle informazioni. Alcuni esempi di tecnologie IT sono: computer, software e reti di comunicazione. [1](#), [2](#), [7](#), [8](#), [82](#), [85](#)

JSON – JavaScript Object Notation: semplice formato di testo utilizzato per lo scambio di dati. Questa notazione utilizza un insieme di coppie nome/valore oppure un elenco ordinato di valori. [71](#)

SEO – Search Engine Optimization: insieme delle pratiche e delle tecniche utilizzate per migliorare la visibilità e il posizionamento di un sito web nei risultati dei motori di ricerca. [47](#), [73](#), [76](#), [77](#)

UI – User Interface: punto di interazione e comunicazione tra l'utente e l'applicazione. Comprende tutti gli elementi visivi e interattivi che permettono agli utenti di navigare e interagire con un sito web o un software. [51](#)

UX – User Experience: come un utente interagisce con un prodotto o un servizio. Un'esperienza utente positiva incoraggia gli utenti a utilizzare il prodotto o servizio in modo frequente e con soddisfazione. [2](#), [7](#), [8](#), [46](#), [67](#), [74](#), [76](#), [79](#), [82](#)

WCAG – Web Content Accessibility Guidelines: linee guida delineate dal Web Accessibility Initiative per rendere i contenuti web accessibili a persone con disabilità, come non vedenti, ipovedenti o persone con disabilità motorie. [44](#), [75](#), [76](#), [77](#), [78](#), [82](#)

Bibliografia

- [1] «Sito web di Girolibero». Consultato: 30 giugno 2024. [Online]. Disponibile su: <https://www.girolibero.it/>
- [2] «Sito web del blog di Girolibero happytobehere». Consultato: 30 giugno 2024. [Online]. Disponibile su: <https://happytobehere.it/>
- [3] «Documentazione dell'editor a blocchi Gutenberg di WordPress.org». Consultato: 30 giugno 2024. [Online]. Disponibile su: <https://developer.wordpress.org/block-editor/>
- [4] «Documentazione di WordPress.org». Consultato: 30 giugno 2024. [Online]. Disponibile su: <https://wordpress.org/documentation/>
- [5] «Documentazione di PHP». Consultato: 30 giugno 2024. [Online]. Disponibile su: <https://www.php.net/manual/it/>
- [6] «Documentazione di MySQL». Consultato: 30 giugno 2024. [Online]. Disponibile su: <https://dev.mysql.com/doc/refman/8.4/en/>
- [7] «Documentazione di Twig». Consultato: 30 giugno 2024. [Online]. Disponibile su: <https://twig.symfony.com/doc/>
- [8] «Documentazione di Timber». Consultato: 30 giugno 2024. [Online]. Disponibile su: <https://timber.github.io/docs/v2/>
- [9] «Documentazione di SaaS». Consultato: 30 giugno 2024. [Online]. Disponibile su: <https://sass-lang.com/documentation/>
- [10] «Documentazione della libreria Bootstrapp». Consultato: 30 giugno 2024. [Online]. Disponibile su: <https://getbootstrap.com/docs/5.3>
- [11] «Documentazione di jQuery». Consultato: 30 giugno 2024. [Online]. Disponibile su: <https://api.jquery.com/>
- [12] «Documentazione di Node.js». Consultato: 30 giugno 2024. [Online]. Disponibile su: <https://nodejs.org/docs/latest/api/>
- [13] «Documentazione di React». Consultato: 30 giugno 2024. [Online]. Disponibile su: <https://legacy.reactjs.org/docs/getting-started.html>
- [14] «Sito web di Cloudinary». Consultato: 30 giugno 2024. [Online]. Disponibile su: <https://cloudinary.com/>
- [15] «Documentazione di MapBox». Consultato: 30 giugno 2024. [Online]. Disponibile su: <https://docs.mapbox.com/mapbox-gl-js/guides>

- [16] «Tutorial: Build your first block». Consultato: 30 giugno 2024. [Online]. Disponibile su: <https://developer.wordpress.org/block-editor/getting-started/tutorial/>
- [17] «Linee guida per l'accessibilità dei contenuti Web». Consultato: 30 giugno 2024. [Online]. Disponibile su: <https://www.w3.org/Translations/WCAG21-it/>