

Claudio De Martino

Matricola 560934

RELAZIONE

Basi di dati e laboratorio web (12 CFU)

Docenti:

Professore Fabrizio Baiardi

Professoressa Barbara Guidi

CdS:

Informatica Umanistica



UNIVERSITÀ DI PISA

SOMMARIO

Specifiche del sistema	1
Progettazione concettuale	2
Schema E-R	2
Dizionario delle entità.....	3
Dizionario delle relazioni	4
Tabella delle business rules.....	5
Progettazione logica.....	6
Schema logico	6
Modello relazionale	6
Vincoli di integrità.....	7
Codice DDL	8
Analisi del database	13
Analisi delle prestazioni	13
Analisi delle ridondanze.....	13
Wireframe	16
Mockup.....	19

Specifiche del sistema

Il sito web, chiamato P-Link (da *Personal Link*), nasce con l'obiettivo di offrire agli utenti registrati la possibilità di creare e gestire blog. Gli utenti possono essere sia semplici **visitatori** che **utenti registrati**.

Un **visitatore** può navigare all'interno del sito, leggere blog ed effettuare ricerche. Un visitatore si può registrare al sito, accedendo alla pagina di registrazione tramite pulsante nell'header e fornendo nickname (univoco), password (max 16 caratteri), email (univoca), numero di telefono/cellulare e gli estremi di un documento.

Il **login**, anch'esso accessibile dal pulsante nell'header, avviene tramite nickname e password; è possibile anche mantenere in memoria i dati di accesso per 30 giorni (utilizzo dei *cookie*). È possibile effettuare in qualsiasi momento il **logout**, sempre tramite pulsante nell'header.

Un **utente registrato** può modificare i propri blog, aggiungere dei post, commentare i post di tutti i blog e mettere Mi piace e Non mi piace. Può inoltre concedere ad un altro utente registrato di essere **coautore** di uno dei suoi blog e successivamente revocare questa possibilità. Viceversa, un coautore può smettere di collaborare ad un blog in qualsiasi momento e ha gli stessi privilegi dell'autore, eccetto l'eliminazione del blog stesso, che rimane un'esclusiva dell'autore. Infine è possibile personalizzare i blog tramite dei set predefiniti messi a disposizione dal sito (oltre a quello di default).

Gli utenti registrati possono **abbonarsi** gratuitamente per utilizzare la funzione di creazione blog. La durata è di 30 giorni dalla data di attivazione.

Nella **userpage** il sito mostra i blog a cui si sta collaborando, i blog creati e il pulsante per l'eliminazione del profilo. Se l'utente risulta abbonato mostra anche il pulsante per la creazione di blog.

Per creare un **blog** è necessario inserire titolo e tema.

Per creare un **post** è necessario inserire almeno titolo e testo, ma è possibile anche allegare un'**immagine** (max 1 MB, le dimensioni vengono riadattate a 640x480). È possibile rimuovere i post (solo dall'autore del blog o da un eventuale coautore).

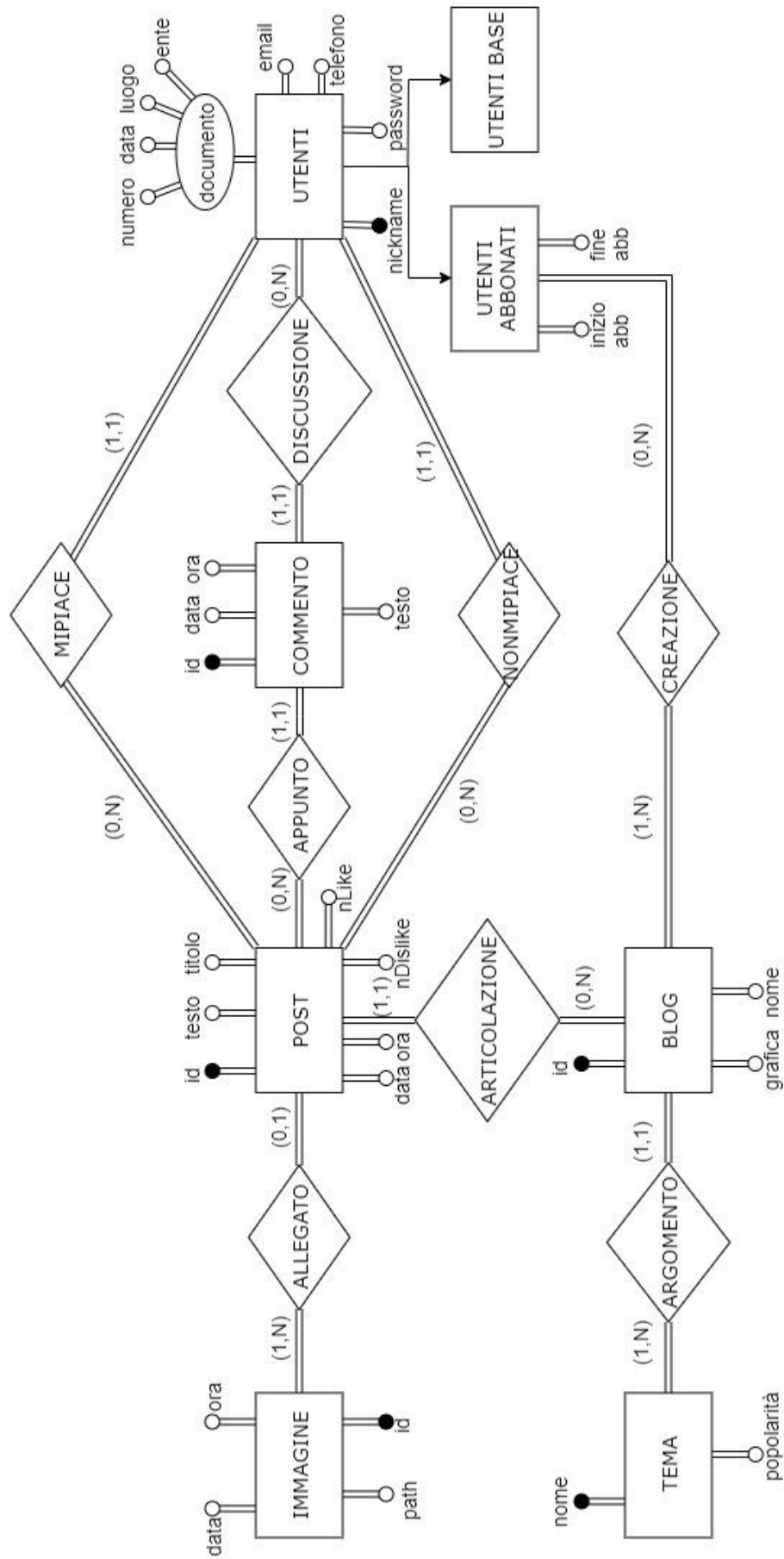
I **commenti** possono essere rimossi solo dai rispettivi utenti. I commenti relativi ad un post vengono comunque rimossi se il post o il blog viene cancellato. Per ogni post, il sito rileva il numero di commenti e decide se mostrarli (max 2 commenti) o creare un pulsante per mostrarli (più di 2 commenti).

Sotto ad ogni post vengono mostrati, solo agli utenti registrati, anche il numero di **Mi piace** e **Non mi piace**, con relativi pulsanti per aggiungere o togliere il giudizio. Viene mostrato un pollice in su se risulta già inserito un Mi piace o un pollice in giù in caso di Non mi piace.

La **barra di ricerca**, situata nell'header, consente, scrivendo una o più parole, di trovare i blog tramite titolo, i post tramite titolo o testo, un utente tramite nickname. È possibile anche cercare un blog o una lista di blog scrivendo il tema.

Progettazione concettuale

Schema E-R



Dizionario delle entità

Entità	Descrizione	Attributi	Identificatore
Utenti	Utente generico della piattaforma	Nickname, password, email, telefono, numero documento, data documento, luogo documento, ente documento	Nickname
Utenti base	Utente non abbonato che non può creare blog		Nickname
Utenti abbonato	Utente abbonato che può creare blog	Inizio abb, fine abb	Nickname
Commento	Messaggio lasciato da un utente ad un post	Id, data, ora, testo	Id
Blog	Blog creato da un utente abbonato	Id, nome, grafica	Id
Post	Articolo contenuto in un blog	Id, titolo, testo, data, ora, nLike, nDislike	Id
Immagine	Immagine allegata ad un post	Id, data, ora, path	Id
Tema	Argomento di un blog	Nome, popolarità	Nome

Dizionario delle relazioni

Relazione	Descrizione	Componenti	Attributi	Tipo
Discussione	Associa un commento al relativo autore	Utenti, commento		1-a-molti
Mi piace	Associa l'utente che ha messo mi piace, ad un post	Utenti, post		1-a-molti
Non mi piace	Associa l'utente che ha messo non mi piace, ad un post	Utenti, post		1-a-molti
Appunto	Associa un commento ad un post	Commento, post		1-a-molti
Articolazione	Associa un post ad un blog	Post, blog		1-a-molti
Allegato	Associa un'immagine ad un post	Immagine, post		1-a-molti
Creazione	Associa un blog al suo autore	Utenti abbonati, blog		1-a-molti
Argomento	Specifica il tema di un blog	Tema, blog		1-a-molti

Tabella delle business rules

Regole di vincolo
(1) Gli utenti base non possono creare blog
(2) Un post può avere come allegato una sola immagine
(3) Un'immagine deve avere massimo 1 MB di peso
(4) Un post può essere creato solo dall'autore o dal coautore del blog
(5) Un commento può essere creato da un solo utente e afferire ad un solo post

Progettazione logica

Schema logico

Modello relazionale

BLOG (blog_id, nomeBlog, graficaBlog)

UTENTI (nickname, password, email, telefono, nDocumento, dataDocumento, luogoDocumento, enteDocumento, inizioAbb, fineAbb)

CREAZIONE (blog_idC, autoreC)

POST (post_id, dataPost, oraPost, titoloPost, nLike, nDislike, testoPost)

ARTICOLAZIONE (blog_idAr, post_idAr)

IMMAGINE (img_id, dataImg, oraImg, pathImg)

ALLEGATO (post_idAl, img_idAl)

COMMENTO (commento_id, dataCommento, oraCommento, testoCommento)

APPUNTO (commento_idAp, post_idAp)

DISCUSSIONE (commento_idD, autoreD)

TEMA (nome, popolarità)

ARGOMENTO (nomeTema, blog_idArg)

MIPIACE (nicknameL, post_idL)

NONMIPIACE (nicknameDL, post_idDL)

Vincoli di integrità

BLOG (blog_id, nomeBlog, graficaBlog)

UTENTI (nickname, password, email, telefono, nDocumento, dataDocumento, luogoDocumento, enteDocumento, inizioAbb, fineAbb)

CREAZIONE (blog_idC, autoreC)

POST (post_id, dataPost, oraPost, titoloPost, nLike, nDislike, testoPost)

ARTICOLAZIONE (blog_idAr, post_idAr)

IMMAGINE (img_id, dataImg, oralmg, pathImg)

ALLEGATO (post_idAl, img_idAl)

COMMENTO (commento_id, dataCommento, oraCommento, testoCommento)

APPUNTO (commento_idAp, post_idAp)

DISCUSSIONE (commento_idD, autoreD)

TEMA (nome, popolarità)

ARGOMENTO (nomeTema, blog_idArg)

MIPIACE (nicknameL, post_idL)

NONMIPIACE (nicknameDL, post_idDL)

[**CREAZIONE** (blog_idC), **ARTICOLAZIONE** (blog_idAr), **ARGOMENTO** (blog_idArg)] → **BLOG** (blog_id)

[**CREAZIONE** (autoreC), **DISCUSSIONE** (autoreD), **MIPIACE** (nicknameL), **NONMIPIACE** (nicknameDL)] → **UTENTI** (nickname)

[**ARTICOLAZIONE** (post_idAr), **ALLEGATO** (post_idAl), **APPUNTO** (post_idAp), **MIPIACE** (post_idL), **NONMIPIACE** (post_idDL)] → **POST** (post_id)

[**ALLEGATO** (img_idAl)] → **IMMAGINE** (img_id)

[**APPUNTO** (commento_idAp), **DISCUSSIONE** (commento_idD)] → **COMMENTO** (commento_id)

[**ARGOMENTO** (nomeTema)] → **TEMA** (nome)

Codice DDL

```
CREATE DATABASE gestore_blog CHARACTER SET utf8 COLLATE utf8_general_ci;
```

```
CREATE TABLE blog (  
    blog_id SMALLINT AUTO_INCREMENT PRIMARY KEY,  
    nomeBlog VARCHAR(20) UNIQUE NOT NULL,  
    graficaBlog VARCHAR(20) NOT NULL DEFAULT "Default.css"  
)ENGINE=INNODB;
```

```
CREATE TABLE utenti (  
    nickname VARCHAR(20) PRIMARY KEY,  
    password VARCHAR(255) NOT NULL,  
    email VARCHAR(30) NOT NULL UNIQUE,  
    telefono VARCHAR(20) NOT NULL,  
    nDocumento VARCHAR(20) NOT NULL,  
    dataDocumento DATE NOT NULL,  
    luogoDocumento VARCHAR(20) NOT NULL,  
    enteDocumento VARCHAR(20) NOT NULL,  
    inizioAbb DATE DEFAULT NULL,  
    fineAbb DATE DEFAULT NULL  
)ENGINE=INNODB;
```

```
CREATE TABLE creazione (  
    blog_idC SMALLINT,  
    autoreC VARCHAR(20),  
    coautoreC VARCHAR(20) DEFAULT NULL,  
    PRIMARY KEY (blog_idC, autoreC),  
    FOREIGN KEY(blog_idC) REFERENCES blog(blog_id) ON DELETE CASCADE,
```

```
FOREIGN KEY(autoreC) REFERENCES utenti(nickname) ON DELETE CASCADE  
)ENGINE=INNODB;
```

```
CREATE TABLE post (  
    post_id INTEGER AUTO_INCREMENT PRIMARY KEY,  
    dataPost date,  
    oraPost time,  
    titoloPost VARCHAR(40) NOT NULL,  
    nLike SMALLINT DEFAULT 0,  
    nDislike SMALLINT DEFAULT 0,  
    testoPost VARCHAR(400)  
)ENGINE=INNODB;
```

```
CREATE TABLE articolazione (  
    blog_idAr SMALLINT,  
    post_idAr INTEGER,  
    PRIMARY KEY (blog_idAr, post_idAr),  
    FOREIGN KEY(blog_idAr) REFERENCES blog(blog_id) ON DELETE CASCADE,  
    FOREIGN KEY(post_idAr) REFERENCES post(post_id) ON DELETE CASCADE  
)ENGINE=INNODB;
```

```
CREATE TABLE immagine (  
    img_id INTEGER AUTO_INCREMENT PRIMARY KEY,  
    dataImg date,  
    oraImg time,  
    pathImg VARCHAR(150)  
)ENGINE=INNODB;
```

```

CREATE TABLE miplace (
    nicknameL VARCHAR(20),
    post_idL INTEGER,
    PRIMARY KEY (nicknameL, post_idL),
    FOREIGN KEY(nicknameL) REFERENCES utenti(nickname) ON DELETE
CASCADE,
    FOREIGN KEY(post_idL) REFERENCES post(post_id) ON DELETE CASCADE
)ENGINE=INNODB;

```

```

CREATE TABLE nonmiplace (
    nicknameDL VARCHAR(20),
    post_idDL INTEGER,
    PRIMARY KEY (nicknameDL, post_idDL),
    FOREIGN KEY(nicknameDL) REFERENCES utenti(nickname) ON DELETE
CASCADE,
    FOREIGN KEY(post_idDL) REFERENCES post(post_id) ON DELETE CASCADE
)ENGINE=INNODB;

```

```

CREATE TABLE allegato (
    post_idAl INTEGER,
    img_idAl INTEGER,
    PRIMARY KEY(post_idAl, img_idAl),
    FOREIGN KEY (post_idAl) REFERENCES post(post_id) ON DELETE CASCADE,
    FOREIGN KEY(img_idAl) REFERENCES immagine(img_id) ON DELETE
CASCADE
)ENGINE=INNODB;

```

```
CREATE TABLE commento (
    commento_id SMALLINT AUTO_INCREMENT PRIMARY KEY,
    dataCommento date NOT NULL,
    oraCommento time NOT NULL,
    testoCommento VARCHAR(140) NOT NULL
)ENGINE=INNODB;
```

```
CREATE TABLE appunto (
    commento_idAp SMALLINT,
    post_idAp INTEGER,
    PRIMARY KEY(commento_idAp, post_idAp),
    FOREIGN KEY(commento_idAp) REFERENCES commento(commento_id) ON
DELETE CASCADE,
    FOREIGN KEY(post_idAp) REFERENCES post(post_id) ON DELETE CASCADE
)ENGINE=INNODB;
```

```
CREATE TABLE discussione (
    commento_idD SMALLINT,
    autoreD VARCHAR(20),
    PRIMARY KEY (commento_idD, autoreD),
    FOREIGN KEY(commento_idD) REFERENCES commento(commento_id)ON
DELETE CASCADE,
    FOREIGN KEY(autoreD) REFERENCES utenti(nickname)ON DELETE CASCADE
)ENGINE=INNODB;
```

```
CREATE TABLE tema (
    nome VARCHAR(20) PRIMARY KEY,
    popolarita SMALLINT NOT NULL DEFAULT 0
)ENGINE=INNODB;
```

```
CREATE TABLE argomento (  
    nomeTema VARCHAR(20),  
    blog_idArg SMALLINT,  
    PRIMARY KEY (nomeTema, blog_idArg),  
    FOREIGN KEY(nomeTema) REFERENCES tema(nome)ON DELETE CASCADE,  
    FOREIGN KEY(blog_idArg) REFERENCES blog(blog_id)ON DELETE CASCADE  
)ENGINE=INNODB;
```


Analisi del database

Il motore usato per la memorizzazione dei dati è InnoDB. La codifica dei caratteri utilizzata è UTF-8, case insensitive. Il database è in terza forma normale (TFN), poiché tutti gli attributi non chiave dipendono dalla chiave soltanto.

Analisi delle prestazioni

Costo registrazione utente

Si assume una media di 30 nuove iscrizioni al giorno.

$$C(O_T) = 30 * 1 * (2 * 1) = 60$$

Costo creazione post (senza immagine)

Si assume una media di 40 nuovi post al giorno.

$$C(O_T) = 40 * 1 * (2 * 2) = 160$$

Costo creazione post (con immagine)

Si assume una media di 40 nuovi post al giorno.

$$C(O_T) = 40 * 1 * (2 * 4) = 320$$

Costo creazione commento

Si assume una media di 40 nuovi post al giorno.

$$C(O_T) = 40 * 1 * (2 * 3) = 240$$

Analisi delle ridondanze

Costo creazione blog

Si assume una media di 15 nuovi blog al giorno. Schema con la ridondanza "popolarità".

$$C(O_{T1}) = 15 * 1 * (2 * 4) = 120$$

Si assume una media di 15 nuovi blog al giorno. Schema senza ridondanza.

$$C(O_{T1}) = 15 * 1 * (2 * 3) = 90$$

Costo visualizzazione tendenza temi

Si assume una media di 50 volte al giorno e 50 blog con lo stesso tema. Schema con la ridondanza "popolarità".

$$C(O_{T2}) = 50 * 1 * (2 * 0 + 1) = 50$$

Si assume una media di 50 volte al giorno e 50 blog con lo stesso tema. Schema senza ridondanza.

$$C(O_{T2}) = 50 * 1 * (2 * 0 + 50) = 2500$$

$$C(O_{TRID}) = 120 + 50 = 170$$

$$C(O_T) = 90 + 2500 = 2590$$

Memoria ridondanza “popolarità”

Si assumono circa 25 temi diversi. **2 Bytes * 25 = 50 Bytes.**

Con un aumento di memoria trascurabile (considerando che il peso complessivo è nell'ordine delle centinaia di migliaia di Bytes) e un aumento prestazionale notevole ($2590 / 170 \approx 15$), conviene mantenere la ridondanza.

Costo mi piace

Si assume una media di 120 mi piace al giorno. Schema con la ridondanza “nLike”.

$$C(O_{T1}) = 120 * 1 * (2 * 2) = 480$$

Si assume una media di 120 mi piace al giorno. Schema senza ridondanza.

$$C(O_{T1}) = 120 * 1 * (2 * 1) = 240$$

Costo visualizzazione numero mi piace

Si assumono una media di 50 volte al giorno e, per ogni blog, 5 post. Inoltre si assumono, in media, 10 mi piace a post. Schema con la ridondanza “nLike”.

$$C(O_{T2}) = 250 * 1 * (2 * 0 + 1) = 250$$

Si assumono una media di 50 volte al giorno e, per ogni blog, 5 post. Inoltre si assumono, in media, 10 mi piace a post. Schema senza ridondanza.

$$C(O_{T2}) = 250 * 1 * (2 * 0 + 10) = 2500$$

$$C(O_{TRID}) = 480 + 250 = 730$$

$$C(O_T) = 240 + 2500 = 2740$$

Memoria ridondanza “nLike”

Si assumono circa 500 mi piace complessivi. **2 Bytes * 500 = 1000 Bytes.**

Con un aumento di memoria trascurabile (considerando che il peso complessivo è nell'ordine delle centinaia di migliaia di Bytes) e un aumento prestazionale contenuto ($2740 / 730 \approx 4$), conviene comunque mantenere la ridondanza.

Costo non mi piace

Si assume una media di 30 non mi piace al giorno. Schema con la ridondanza “nDislike”.

$$C(O_{T1}) = 30 * 1 * (2 * 2) = 120$$

Si assume una media di 30 non mi piace al giorno. Schema senza ridondanza.

$$C(O_{T1}) = 30 * 1 * (2 * 1) = 60$$

Costo visualizzazione numero non mi piace

Si assumono una media di 50 volte al giorno e, per ogni blog, 5 post. Inoltre si assumono, in media, 3 non mi piace a post. Schema con la ridondanza “nDislike”.

$$C(O_{T2}) = 250 * 1 * (2 * 0 + 1) = 250$$

Si assumono una media di 50 volte al giorno e, per ogni blog, 5 post. Inoltre si assumono, in media, 3 non mi piace a post. Schema senza ridondanza.

$$C(O_{T2}) = 250 * 1 * (2 * 0 + 3) = 750$$

$$C(O_{TRID}) = 120 + 250 = 370$$

$$C(O_T) = 60 + 750 = 810$$

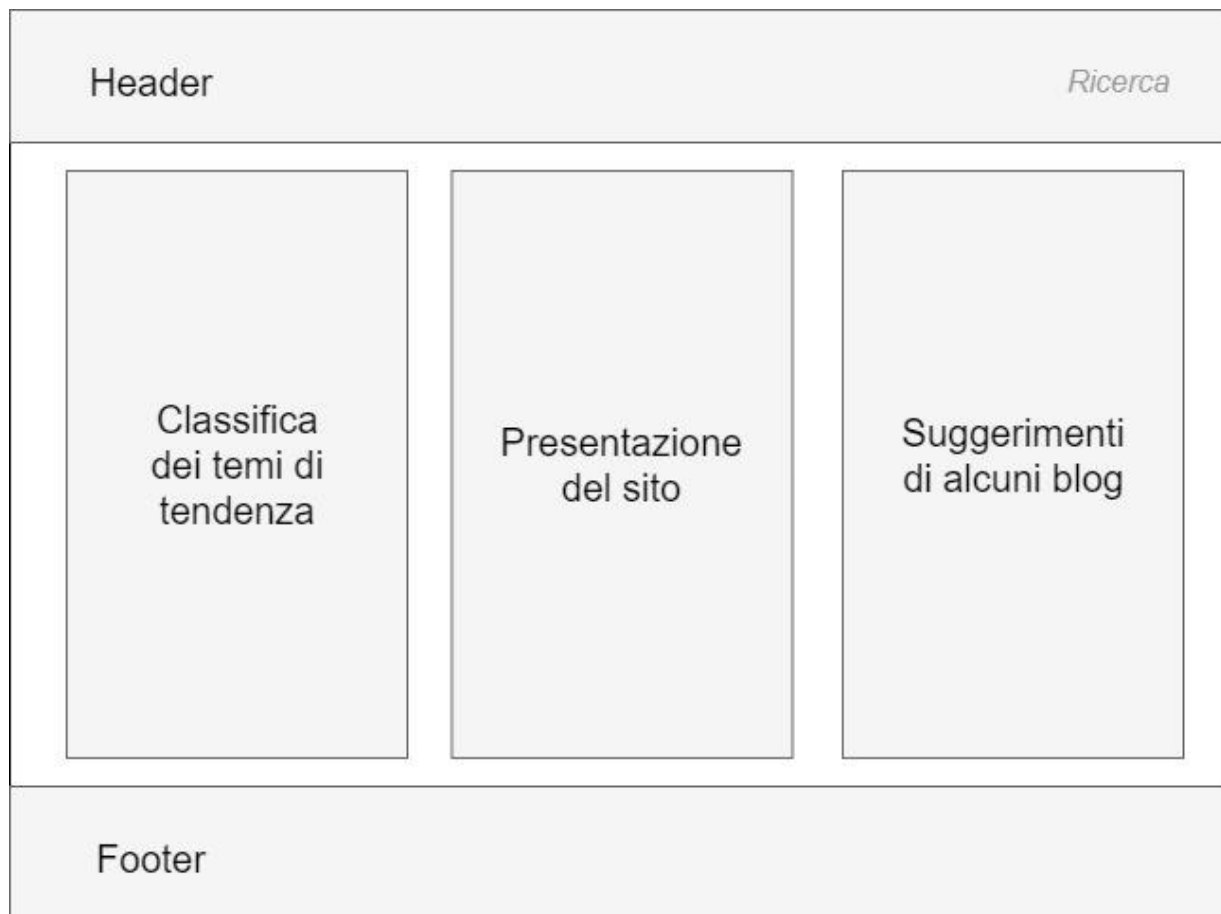
Memoria ridondanza “nDislike”

Si assumono circa 150 non mi piace complessivi. **2 Bytes * 150 = 300 Bytes.**

Con un aumento di memoria trascurabile (considerando che il peso complessivo è nell'ordine delle centinaia di migliaia di Bytes) e un aumento prestazionale minimo ($810 / 370 \approx 2$), conviene comunque mantenere la ridondanza, considerando che si sono assunti numeri abbastanza bassi e che devono abbinarsi ai mi piace.

Wireframe

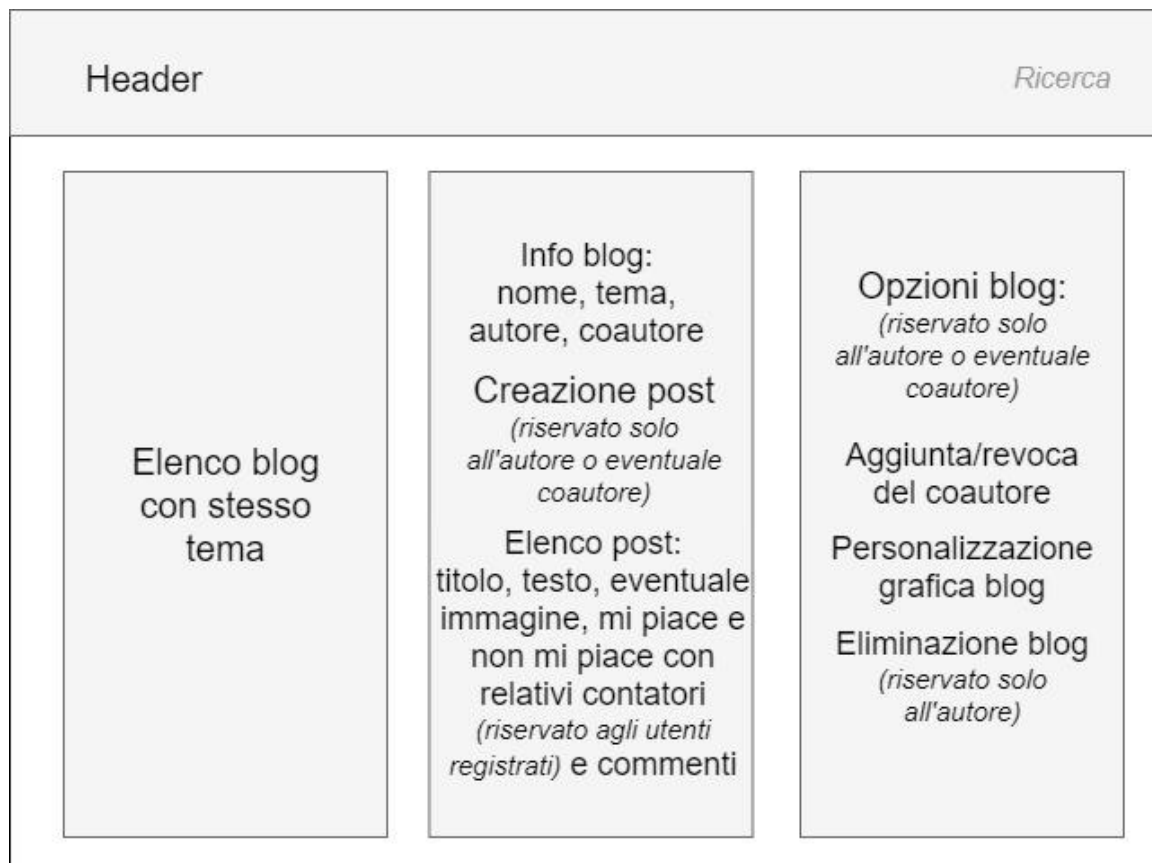
Index:



Userpage (riservata agli utenti registrati):



Blog:



Login:

Header		Ricerca
<div>Logo</div> <div>Campi per accedere al sito</div>		
Footer		

Registrazione:

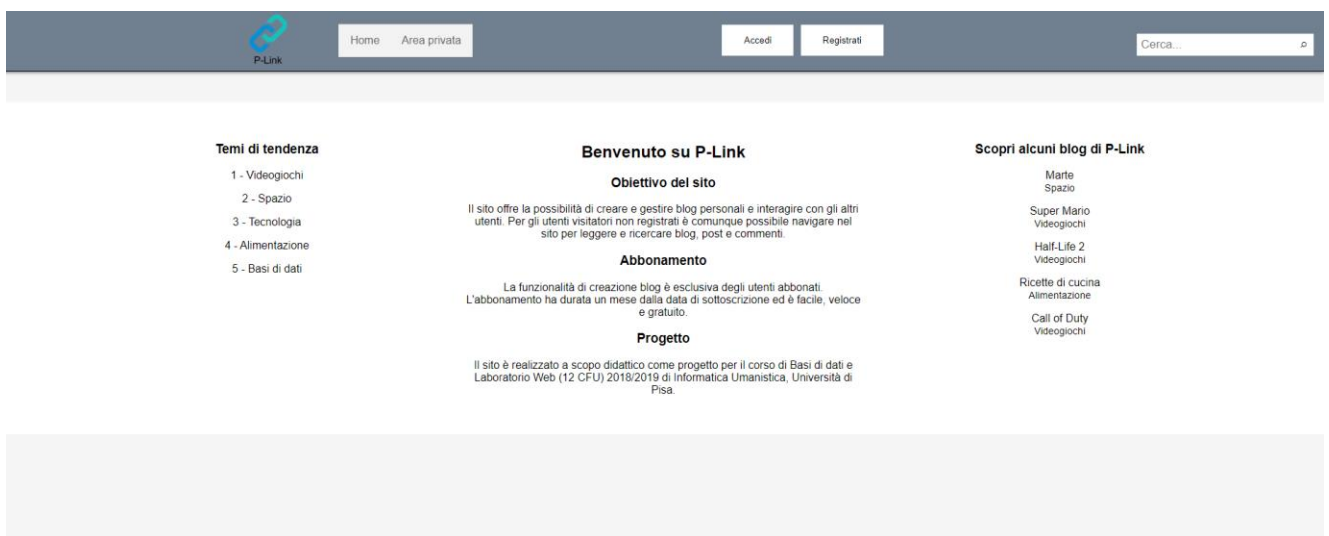
Header		Ricerca
<div>Istruzioni per la registrazione</div> <div>Campi per registrarsi al sito</div>		
Footer		

Search:



Mockup


Index:



Ideato e progettato da:

Claudio De Martino
claudio.demartino11@gmail.com

Userpage (riservata solo agli utenti registrati):


[Home](#) [Area privata](#)

Ciao Profilo11 [Esci](#)

Cerca...

Blog di cui sei coautore
Non sei ancora coautore di nessun blog.**Area privata**
Status login = OK
Status abbonamento = Abbonato
[Crea nuovo blog](#)
I tuoi blog
Progetto**Profilo**
[Cancella profilo](#)Ideato e progettato da: **Claudio De Martino**
claudio.demartino11@gmail.com

Blog:


[Home](#) [Area privata](#)


[Accedi](#) [Registrati](#)

Cerca...

Altri blog con lo stesso tema
[Super Mario](#)
[Call of Duty](#)
[Age of Empires](#)**Half-Life 2**
Genere: Videogiochi
Autore: Profilo7
Coautore: nessuno
Post
Annunciato Half-Life Alyx! - 16/12/2019 - 17:28:36
Sarà un gioco in VR, prima grande produzione nel suo genere. I fatti narrati saranno antecedenti alla storia di Half-Life 2. Uscirà a marzo 2020.
Commenti
Non ci sono commenti.

Login:

[Home](#) [Area privata](#) [Registrati](#)




P-Link

Login

☐ Mantieni l'accesso

Registrazione:

[Home](#) [Area privata](#) [Accedi](#)

Registrazione

Attraverso questo modulo di registrazione sarà possibile iscriversi alla community del sito. Potrete avere un vostro profilo, commentare i post e mettere Mi piace o Non mi piace. Inoltre, se in seguito si sceglie di sottoscrivere l'abbonamento premium, sarà possibile anche creare, modificare e gestire blog e pubblicare post.

Il **nickname** non può contenere spazi, caratteri accentati e caratteri speciali eccetto * - _ . La **password** deve avere almeno 4 caratteri ed un massimo di 16, inoltre non può contenere spazi, caratteri accentati e caratteri speciali eccetto * - _ .

N.B.: i campi sono tutti obbligatori.

Generali

Documento d'identità

gg/mm/aaaa

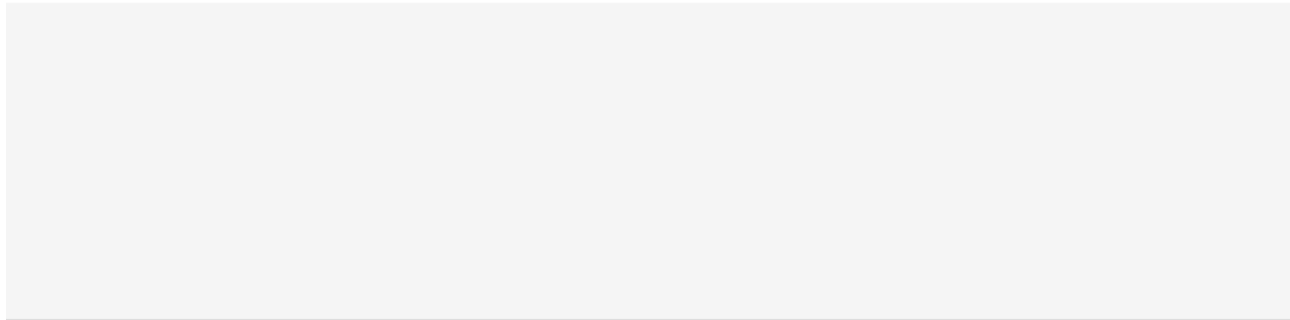
Ideato e progettato da:

Claudio De Martino
claudio.demartino11@gmail.com

Search:

Blog trovati o che potrebbero interessarti

Tech news



Ideato e progettato da:

Claudio De Martino
claudio.demartino11@gmail.com