

ALMA MATER STUDIORUM – UNIVERSITA' DI BOLOGNA  
CAMPUS DI CESENA  
SCUOLA DI SCIENZE

---

CORSO DI LAUREA MAGISTRALE IN SCIENZE E TECNOLOGIE  
INFORMATICHE

METADATI E INTEGRAZIONI SOCIAL  
NEL WEB ODIERNO

Tesi di laurea in  
Web Semantico

Relatore  
Prof.ssa Antonella Carbonaro

Presentata da  
Marco Succi

---

Sessione III  
Anno Accademico 2013-2014



*A Veronica e Giulio*





---

# Indice

<b>Indice .....</b>	<b>3</b>
<b>Indice delle Figure.....</b>	<b>7</b>
<b>Introduzione.....</b>	<b>11</b>
<b>1. Social Network.....</b>	<b>15</b>
<i>1.1. La nascita di un nuovo mondo .....</i>	<i>15</i>
<i>1.2. Giant Global Graph e Web 3.0 .....</i>	<i>19</i>
<i>1.3 Social Graph .....</i>	<i>21</i>
<b>2. Applicazioni e integrazioni Social.....</b>	<b>23</b>
<i>2.1 Social Login .....</i>	<i>23</i>
2.1.1 Permessi legati al Facebook Login.....	25
<i>2.2 Facebook App.....</i>	<i>27</i>
2.2.1 Pubblicazione di un'applicazione.....	28
<i>2.3 Facebook Social Plugin .....</i>	<i>29</i>
2.3.1 Like Button e Like Box .....	30
2.3.2 Share Button e Share Dialog .....	30
2.3.3 Comments Plugin .....	31
2.3.4 Activity e Recommendations Feed .....	32
<i>2.4 Facebook Graph Search .....</i>	<i>32</i>
<i>2.5 Google+, MyBusiness e Schema.org .....</i>	<i>34</i>
<b>3. Open Graph Protocol.....</b>	<b>37</b>
<i>3.1 Dal "Mi Piace" all'Open Graph .....</i>	<i>37</i>
<i>3.2 L'Open Graph in pillole.....</i>	<i>38</i>
<i>3.3 Lo schema dei meta-dati .....</i>	<i>39</i>
3.3.1 I meta-dati di base .....	39
3.3.2 I meta-dati opzionali.....	40
3.3.3 I meta-dati strutturati.....	41

3.4 L'evoluzione (Video, libri, musica...)	42
3.4.1 Video	42
3.4.2 Music	43
3.4.3 Book	44
3.5 Custom Stories	44
<b>4. Graph API</b>	<b>47</b>
4.1 REST API	47
4.2 JSON	50
4.3 Autenticazione OAuth	50
4.4 Graph API nel dettaglio	53
4.4.1 Access Token	55
4.4.2 Permessi	56
4.4.3 Elaborazione dei risultati	57
<b>5. Il Progetto "Volontieri"</b>	<b>59</b>
5.1 Introduzione al progetto	59
5.2 Architettura del progetto	60
5.2.1 WordPress	60
5.2.2 Facebook PHP SDK	62
5.2.2 Alberatura del progetto	62
5.3 Integrazioni social	65
5.3.1 Creazione dell'App Volontieri	65
5.3.2 Login	65
5.3.3 Metadati Open Graph	72
5.3.4 Custom story e Facebook feed	74
5.3.5 Plugin sociali	83
5.3.6 Inserimento Schema.org	85
5.4 Estensioni future	86
<b>Conclusioni</b>	<b>87</b>
<b>Bibliografia</b>	<b>93</b>





---

# Indice delle Figure

Figura 1.1 - Friendster (Wired) .....	16
Figura 1.2 - The Facebook (Wikipedia).....	18
Figura 1.3 - FOAF (Swad) .....	20
Figura 2.1 - Esempio di Social Login (Intenthq) .....	24
Figura 2.2 - Accettazione App Facebook Login .....	26
Figura 2.3 - Creazione Facebook App. (Facebook) .....	28
Figura 2.4 - Developer App dashboard (Facebook).....	29
Figura 2.5 - Like Box (Facebook).....	30
Figura 2.6 - Share Dialog (Facebook).....	31
Figura 2.7 - Comments plugin (Facebook) .....	32
Figura 2.8 - Facebook Graph Search.....	33
Figura 2.9 - Interfaccia home page(Google+).....	34
Figura 2.10 - Rich Snippet (Google).....	35
Figura 3.1 - The Social Graph (Business Insider).....	38
Figura 3.2 - Esempi di Custom story Facebook.....	44
Figura 3.3 - Tripla di una Facebook story.....	45
Figura 4.1 - Flusso di autenticazione OAuth .....	52
Figura 4.2 - JSON informazioni pagina FacebookItalia .....	54
Figura 5.1 - Volontieri - Home Page.....	64
Figura 5.2 - Facebook Login - Impostazione parametri e permessi.....	66
Figura 5.3 - Facebook Login - Inizializzazione classi .....	67
Figura 5.4 - Facebook Login - Verifica sessioni.....	68
Figura 5.5 - Facebook Login - Logout url.....	68
Figura 5.6 - Facebook Login - Richiesta utente.....	69
Figura 5.7 - Wordpress - Utente backend .....	70
Figura 5.8 - Creazione utente Wordpress.....	70
Figura 5.9 - Facebook Login - Logica pulsante .....	71
Figura 5.10 - Volontieri - Flusso Login .....	72
Figura 5.11 - Open Graph - Integrazione metadati .....	73

---

Figura 5.12 - Facebook dashboard - Definizione Custom Story .....	74
Figura 5.13 - Facebook dashboard - Codice esempio oggetto Custom Story .....	75
Figura 5.14 - Facebook dashboard - Markup Open Graph oggetto Custom Story	76
Figura 5.15 - Volontieri - Recensione, utente non riconosciuto.....	77
Figura 5.16 - Volontieri - Recensione, utente riconosciuto.....	78
Figura 5.17 - WordPress - Custom field rating .....	79
Figura 5.18 - Volontieri - Recensione utente .....	79
Figura 5.19 - Wordpress - Creazione Custom Story .....	80
Figura 5.20 - Facebook - Custom Story all'interno delle attività recenti Utente ..	80
Figura 5.21 - Facebook - Custom Story (News Feed).....	81
Figura 5.22 - WordPress - Pubblicazione post sulla bacheca utente .....	81
Figura 5.23 - Facebook - Custom Story (Profilo utente).....	82
Figura 5.24 - Volontieri - Plugin LikeBox .....	83
Figura 5.25 - Volontieri - Plugin LikeBox .....	83
Figura 5.26 - Volontieri – Follow Button.....	84





# Introduzione

Sono trascorsi più di venti anni dal giorno in cui, nell'agosto del 1991, sir Tim Berners-Lee e il suo collega Robert Cailliau, rivoluzionarono il mondo tecnologico pubblicando la prima pagina web.

Da allora sono stati sviluppati e rilasciati, e tuttora continuano ad esserlo, numerosi linguaggi di programmazione per la pubblicazione di pagine web, lato client e server, molti dei quali principalmente finalizzati alla presentazione, gestione e organizzazione di documenti e contenuti di vario genere.

Ciò che soprattutto è aumentata però è la quantità d'informazioni presenti nel web, una risposta alla crescita altrettanto inarrestabile del bacino di utenti connessi ad internet, cresciuto rapidamente ed esponenzialmente, grazie inevitabilmente alla diffusione di dispositivi sempre più mobile e sempre più connessi.

Per questo, e in previsione delle naturali necessità di ricerca e di fruizione di queste informazioni da parte degli utenti, da diversi anni si parla di Web Semantico e Web 3.0, una versione e visione del web costituita da documenti non solo consultabili ma anche interpretabili, arricchendolo nei suoi contenuti, espandendone di fatto le capacità, grazie ad una trasformazione da *machine-readable* a *machine-understandable*.

In quest'ottica sono stati rilasciati dal W3C alcuni standard importanti come *RDF* (Resource Description Framework) e *OWL* (Web Ontology Language) per poter esprimere concetti complessi e articolati attraverso associando ai contenuti significanti attributi semantici.

Altra importante W3C Recommendation è stato *RDFa* (Resource Description Framework in Attributes), strumento proposto per associare etichette ai contenuti, al fine di descriverli rendendoli riconoscibili come uno specifico tipo di informazioni, come la recensione di un ristorante, un evento, una persona o la scheda di un prodotto.

Rispetto ai primi due citati, quest'ultimo strumento sta vivendo una favorevole diffusione, favorita dalla semplicità d'integrazione dei suoi attributi nei tag XHTML (come `<span>` o `<div>`) per assegnare nomi brevi e descrittivi alle entità e alle loro proprietà.

Fra i principali attori protagonisti a farne uso e ad averlo scelto come base per lo sviluppo delle proprie API, troviamo i più importanti Social Network al momento in cui viene proposto questo lavoro.

Facebook prima, LinkedIn e Twitter poi, hanno infatti adottato come proprio standard per l'interpretazione e l'estensione dei contenuti all'interno delle pagine web il protocollo *Open Graph*, strumento basato proprio su RDFa, ispirato nel suo utilizzo e nella semplicità di integrazione da *FOAF* e *Dublin Core*.

In questo contesto viene sviluppato il progetto di tesi, per presentare un esempio concreto di come sia possibile sfruttare le potenzialità messe a disposizione dall'integrazione di questi protocolli di markup, per la definizione, tramite l'utilizzo di metadati, di concetti semanticamente ricchi e interpretabili dai principali Social Network.

Per farlo è stato realizzato un sito web dal nome *Volontieri*, dedicato alla promozione dei progetti di Associazioni senza scopo di lucro, tese all'aiuto del prossimo, le quali potranno sfruttare le potenzialità messe a disposizione da questi strumenti, per aumentare il bacino di potenziali volontari ed interessati.

In questo sito gli utenti potranno, tramite le principali integrazioni in questo momento fornite dalle API distribuite dai Social Network, pubblicare recensioni e condividerle nei propri profili promuovendo in modo semplice e immediato le storie vissute dai volontari del progetto.

Il lavoro svolto, può essere riassunto nella seguente struttura di capitoli:

- Il *primo capitolo* vuole introdurre al mondo dei Social Network, a come questi si siano a mano a mano integrati nella visione crescente di un Web Semantico e quali di questi stiano dando in questo senso la spinta maggiore;
- Il *secondo capitolo* fornisce una ricca panoramica degli strumenti distribuiti dai Social network ai developer, per l'integrazione di

plugin sociali all'interno delle proprie pagine web e per la produzione d'implementazioni con il Grafo Sociale;

- Il *terzo capitolo* affronta in maniera esaustiva il protocollo Open Graph, sviluppato per rendere ogni pagina web, un oggetto interpretabile all'interno del concetto di Grafo Sociale;
- Nel *quarto capitolo* vengono affrontate in maniera esaustiva le Graph API, librerie prodotte da Facebook per l'interrogazione del Grafo Sociale;
- Il *quinto capitolo* offre una descrizione del progetto realizzato, mostrando una panoramica delle tecnologie utilizzate e delle integrazioni con gli strumenti sopra citati, analizzandone i frammenti di codice principali.



# 1. Social Network

## 1.1. La nascita di un nuovo mondo

È generalmente definito con il termine *Social Network*, un servizio web-based capace di consentire agli utenti di:

1. costruire un profilo pubblico o semi-pubblico all'interno di un sistema delimitato;
2. articolare una lista di altri utenti con i quali viene condivisa una connessione;
3. visitare e attraversare la propria lista di connessioni e quelle costruite da altri all'interno del sistema.

La natura e la nomenclatura di queste connessioni possono variare da sito a sito. In accordo con la definizione sopra data, il primo Social Network ad esser stato lanciato, fu *SixDegrees.com* nel 1997 [BEL07].

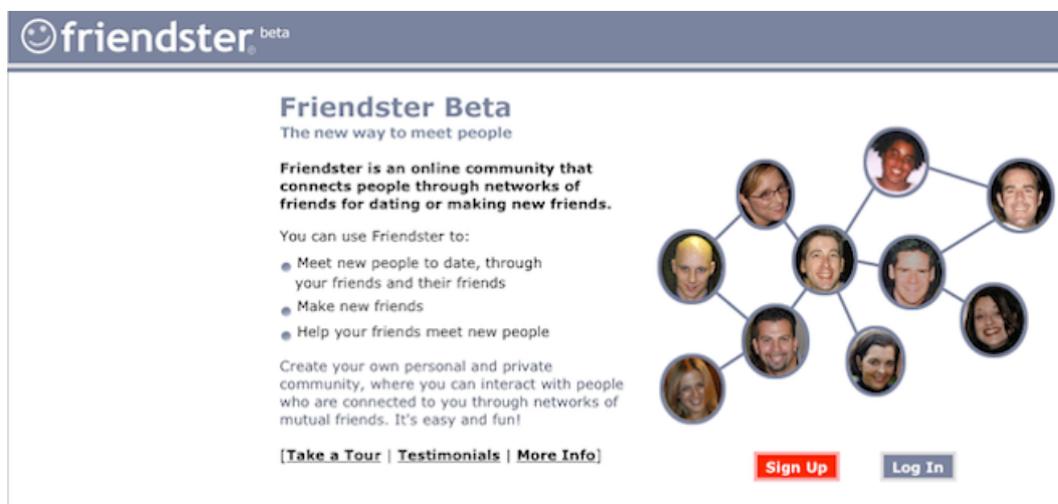
Questo portale permetteva agli utenti di creare un proprio profilo contenente le relative informazioni personali, visualizzare un elenco dei propri amici e, dall'anno successivo al suo rilascio, anche l'elenco degli amici dei propri amici.

Alcune di queste feature erano già note all'epoca, poiché presenti in alcuni servizi web come le "buddy list" di *AIM* e *ICQ* e nei collegamenti che potevano essere stretti tramite *Classmates.com*, ma *SixDegrees* è stato senza dubbio un servizio innovativo poiché si è prodigato per primo per combinarle tutte insieme.

Forse proprio per questo riuscì ad attrarre milioni di utenti in poco tempo. Utenti che però purtroppo nel 2000, solo pochi anni dopo la sua nascita, ne videro arrivare il suo tramonto, a causa dell'insostenibilità economica del progetto. Nonostante l'avanguardia delle sue funzionalità, era comunque un sistema ancora troppo acerbo e incapace di soddisfare le esigenze degli utenti, per via delle limitate operazioni messe a disposizione.

Nel 2002 fu il momento di *Friendster*, progettato e realizzato per competere con *Match.com*, un proficuo sito di dating online. Ma mentre la maggior parte dei siti d'incontri era focalizzata sul cercare di introdurre e favorire la conoscenza di persone ad estranei sulla base dell'affinità dei propri interessi, Friendster è stato ideato per aiutare gli amici-di-amici ad incontrarsi, basandosi sul presupposto che gli amici-di-amici possono avere una compatibilità maggior rispetto a quelli di due estranei.

La popolarità di Friendster crebbe così velocemente da raggiungere l'incredibile quota di tre milioni di utenti in soli tre mesi [CUR13].



*Figura 1.1 - Friendster (Wired)*

Una crescita così insapettata, considerata la sua principale diffusione allora unicamente basata sul passaparola, da causare svariate difficoltà tecniche dovute ad un sotto dimensionamento dei server e dei database dedicati.

Difficoltà che portarono spesso ad una frustrazione dei suoi utenti, soprattutto coloro che avevano visto in questo servizio una possibile alternativa all'utilizzo delle e-mail.

L'assalto di nuovi utenti, venuti a conoscenza del portale tramite campagne di promozione mediatica, comportò inoltre un tracollo dei contesti sociali: gli utenti iniziarono infatti ad incontrare i propri datori di lavoro ed ex compagni di classe insieme ai loro amici più stretti.

Il progetto iniziale di Friendster limitava gli utenti nella visualizzazione dei soli profili di persone che erano al più di quattro gradi di distanza (amici-di-amici-di-amici-di-amici). Di conseguenza, per visualizzare altri profili, gli utenti iniziarono così ad aggiungere conoscenti e sconosciuti, cercando di espandere la portata dei profili visualizzabili. Tutto questo portò alla generazione di profili che iniziarono a raccogliere in maniera massiccia Amici, attività implicitamente incoraggiata attraverso una funzione del portale chiamata "Most Popular".

Questi collezionisti detti "Fakesters" erano spesso falsi profili che rappresentati collegati a personaggi di fantasia, celebrità e concetti in genere.

La combinazione di continue difficoltà tecniche e collisioni sociali causarono una rottura fra gli utenti e il sito che portò a un crollo totale di utilizzi negli Stati Uniti. A gioire delle sue difficoltà e degli abbandoni dei suoi utenti, fu *MySpace*, Social Network avviato nel 2003 a Santa Monica, California.

Anche se nel suo intento primario non era stato pensato per la promozione di gruppi musicali, questi vennero da subito ben accolti e svariate band iniziarono a creare i loro profili.

Ma la fonte di crescita che ha portato MySpace a espandersi ben oltre gli ex utenti Friendster, non è stato il numero di band registrate ma, principalmente, la dinamica band-fan, reciprocamente vantaggiosa: le band volevano essere in grado di raggiungere i propri fan, mentre questi desideravano ottenere l'attenzione dei loro gruppi preferiti, utilizzando l'uso della "connessione di amicizia" per segnalare la propria identità e l'appartenenza.

Nel 2004 MySpace conobbe un aumento sensibile dei propri utenti, grazie anche alla modifica delle proprie restrizioni riguardo alla registrazione dei minori, permettendo l'affacciarsi al sito a milioni di adolescenti.

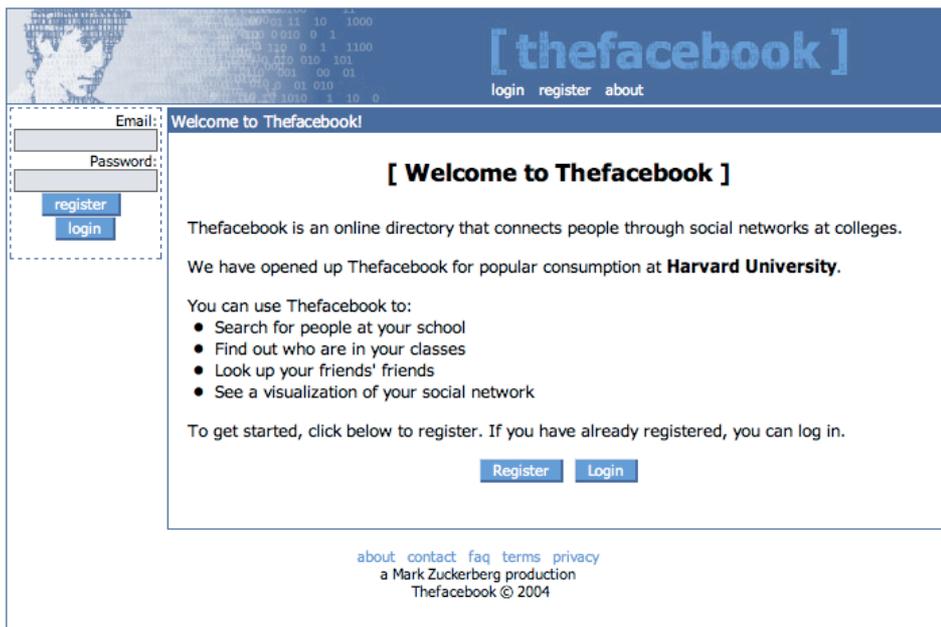
Nello stesso anno, due studenti universitari compagni di stanza, Mark Zuckerberg ed Eduardo Saverin, diedero vita al più importante Social Network finora prodotto, *TheFacebook* poi divenuto solo *Facebook*.

Inizialmente riservato agli studenti dell'Università di Harvard, dove entrambi studiavano, fu poi aperto anche agli studenti dei college di Boston e della Ivy League, per poi arrivare a coprire in breve tempo la maggior parte delle facoltà di Canada e Stati Uniti.

---

L'ultimo passo per arrivare alla crescita globale avvenne nel 2006 quando Facebook, aprì le sue porte al mondo intero, applicando due uniche restrizioni: la richiesta di un indirizzo email valido ed un'età maggiore di 13 anni.

Da allora questo Social Network ha registrato una crescita esponenziale senza precedenti che lo ha visto arrivare nel recente settembre 2014 alla quota di 1,35 miliardi di utenti [COI – 2014].



**Figura 1.2 - The Facebook (Wikipedia)**

Numeri e crescita simili, ma comunque inferiori, raggiunti soltanto da un altro grande Social Network, nato nel 2006 in California come servizio per consentire agli utenti di scrivere e leggere messaggi di 140 caratteri, chiamato *Twitter*.

Ad oggi è l'unico grande rivale in termini numerici e in fatto di monetizzazione per quel che riguarda campagne marketing e strategie social oriented.

A caratterizzare la popolarità di altri Social Network nati in questi anni e a favorirne l'utilizzo da parte di milioni di utenti sono state indubbiamente la capacità di innovare e non replicare i servizi già presenti, introducendo nuovi modi di comunicare con gli altri, grazie a servizi sempre più multi canale e multi Platform e spesso categorizzati, come ad esempio LinkedIn nell'ambito della condivisione della propria esperienza professionale, Instagram e Flickr dedicati al mondo della fotografia, Pinterest in generale dedicato al mondo della creatività.

Unico vero neo in questo senso è rappresentato da Google+, sistema di social networking introdotto dal colosso Google nel 2011, il quale ha in parte preso spunto dai Social Network preesistenti, integrandone però l'utenza con quelle delle piattaforme di sua proprietà come Gmail, Youtube, Hangouts, etc.

## **1.2. Giant Global Graph e Web 3.0**

Era il 2007, quando in un articolo del suo blog, Tim Berners-Lee, uno dei principali artefici del World Wide Web, utilizzò per la prima volta il termine *Giant Global Graph*, definendo quest'ultimo come il terzo gradino di un sistema a livelli.

Questo "sistema" è costituito, in primo luogo, da *Internet*, la rete fisica, regolata da protocolli di comunicazione, capace di collegare fra loro computer a migliaia di chilometri di distanza. Internet ha per la prima volta concentrato il focus sui computer, sui dispositivi capaci di comunicare tra di loro, evidenziando che non sono i cavi che trasmettono i segnali definiti un'informazione ad essere interessanti, ma i dispositivi stessi che li scambiano e li interpretano.

Il secondo protagonista è il *World Wide Web*, risultato dell'insieme di codice e di contenuti tramite i quali sono descritti miliardi di documenti e di relazioni che collegano quest'ultimi. Grazie al WWW è diventato possibile navigare un'infinita quantità di documenti senza doversi più interessare in quale computer questi siano contenuti, spostando l'importanza dai computer ai documenti stessi che essi contengono.

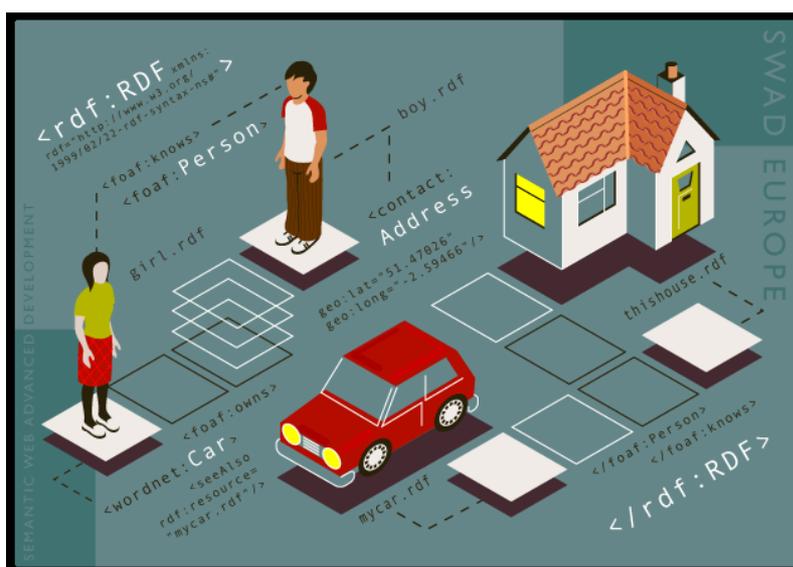
Il Web nei suoi quasi vent'anni di vita si è evoluto notevolmente passando da un Web composto e concentrato principalmente sui documenti al Web 2.0, nel quale si è posta la persona al centro e non più la tecnologia, grazie soprattutto agli sviluppi che han portato a minori curve di apprendimento per la creazione e la diffusione di contenuti da parte degli utenti.

In questo passaggio concettuale e questa dinamica sociale, oggi siamo ad un altro punto di svolta, non meno importante del passaggio conosciuto come Web 2.0.

Si pensa al Web come un grande grafo che collega tutto e tutti, che collega persone e cose del mondo, che collega le persone alle persone, che collega il mondo e le nostre rappresentazioni della sua controparte digitale [NOV - 2010].

Ragionando in termini matematici, sia la rete Internet che il WWW modellano quindi qualcosa che può essere connotabile come un grafo, ma con differenti strumenti e fini. Come da definizione, infatti, Internet è stato sviluppato per connettere computer mentre il Web per collegare documenti.

E allora ecco il Giant Global Graph, una nuova rete, un sistema in cui le informazioni disegnate vanno oltre le pagine web discrete, grazie alla definizione e all'enfatizzazione di contenuti semantici tramite l'utilizzo di metadati che descrivono oggetti informativi in modo più completo attribuendo relazioni che concettualmente collegano gli oggetti informativi fra loro.



**Figura 1.3 - FOAF (Swad)**

I servizi di social networking sono uno degli esempi evidenti di questa distinzione fra informazioni e relazioni. Ciò che, infatti, sono maggiormente valorizzate sono le informazioni descrittive le relazioni tra le persone e i differenti tipi di oggetti che gli utenti condividono.

Il problema principale, come sottolineato anche da Berners-Lee, è che ad oggi questi nuovi tipi di dati sono strutturati e gestiti principalmente dalle aziende protagoniste in ambito social.

Nell'ipotesi stessa però di un Giant Graph davvero globale, una necessità primaria è la decentralizzazione delle informazioni, strutturate e distribuite in modo da poter essere leggibili da qualsiasi piattaforma o sistema, organizzate dinamicamente in una pluralità differente di formati leggibili dall'utente.

Un esempio concreto proposto da Berners-Lee, è dato da *FOAF*, acronimo di *Friend of a friend* (Amico di un amico), un'ontologia atta a descrivere persone, con le loro attività e le relazioni con altre persone e oggetti. Chiunque può usare FOAF per descriversi, permettendo a gruppi di persone di descrivere quel fenomeno noto come social network senza la necessità di un database centralizzato [WIK - FOA].

In sostanza quindi, mentre il Web 3.0 si può riassumere come una suite di tecnologie o ad una particolare fase dello sviluppo del web e del suo cammino, il termine Giant Global Graph è destinato a riferirsi, in modo più alto, all'ambiente totale delle informazioni che verrà generato e sostenuto attraverso l'implementazione di queste tecnologie.

Questo ambiente è teso a diventare qualitativamente migliore e maggiormente completo rispetto a quello conosciuto prima dello sviluppo di queste tecnologie [WIK-GIA].

## **1.3 Social Graph**

Il *Social Graph* è la rete di legami (connessioni) che esistono nel modo in cui ogni soggetto (nodo) comunica e scambia informazioni con gli altri nodi [FUR07].

Tramite il grafo vengono quindi rappresentati globalmente l'insieme ed il grado delle relazioni personali e i soggetti che ne fanno parte, mappando una vera e propria rete sociale.

Il grafo è stato ed è tuttora la linfa vitale dei Social Network, la loro essenza, tramite la quale quotidianamente noi stessi definiamo con un crescente dettaglio chi siamo, le nostre relazioni, i nostri interessi e le nostre passioni.

Come definito da M. Zuckerberg, creatore di Facebook, il grafo sociale è il motivo stesso per cui Facebook esiste. Maggiore è il numero di utenti e della

definizione di connessioni fra questi, maggiore diventa l'utilità e la capacità funzionale stessa di Facebook [FAR07].

Per estendere il potere descrittivo di questa rete, non solo ai soggetti, intesi come persone o aziende, all'interno del grafo e alle loro relazioni, ma anche a tutti gli attori esterni che li circondano e fanno parte del Web, Facebook ha introdotto, nel 2008, l'*Open Graph Protocol*.

Il protocollo OG è un insieme di metodi meta descrittivi finalizzati ad attribuire un contenuto semantico a ogni pagina web, trasformando il suo contenuto in un insieme di informazioni interpretabili e interrogabili.

Così facendo la si rende parte integrante del Social Graph, ovvero un oggetto capace di interagire con i social network.

Allo stato dell'arte, non è ancora stato prodotto un protocollo per la definizione dell'identità degli utenti universalmente utilizzato ed unificato, aperto e disponibile, fra i vari servizi di social networking presenti, generando spesso un'esperienza di presenza sociale imperfetta.

Nel gennaio del 2008, per cercare di contrastare il dominio informativo di Facebook, Google ha cercato di offrire una soluzione introducendo le *Social Graph API* permettendo ai siti web di elaborare informazioni disponibili pubblicamente, permettendo agli utenti di creare una forma d'identità portatile.

Questo progetto non ha però riscontrato il successo sperato, causando il ritiro del progetto nel 2012.

Dal 2010 Facebook, grazie alla distribuzione di Open Graph Protocol, all'introduzione delle *Graph API*, per permettere l'interrogazione del Social Graph e l'integrazione con applicazioni di terze parti e alla sua diffusione come social network, è diventato il più grande dataset sociale al mondo, contenente il maggior numero di relazioni definite tra il più grande numero di persone nell'ambito social.

## 2. Applicazioni e integrazioni Social

Per aumentare sempre di più il livello di coinvolgimento degli utenti con il mondo dei Social Network, sono sempre di più le integrazioni e le estensioni sviluppate e rese disponibili dai principali competitors della scena per essere partecipi e protagonisti di questo mondo, anche se non vi stiamo teoricamente interagendo.

Vediamo ora nel dettaglio alcune delle principali funzionalità messe a disposizione dai Social Network.

### 2.1 Social Login

Il *Social login*, anche conosciuto come social sign-in, è un servizio di single sign-on tramite il quale, previa richiesta di autorizzazione, vengono lette e memorizzate le informazioni inserite all'interno di un servizio di social networking come Facebook, Twitter o Google+ per consentire la registrazione all'interno di un sito web esterno, permettendo così una creazione più rapida di credenziali d'accesso utente all'interno del proprio sito [WIK - SOC].

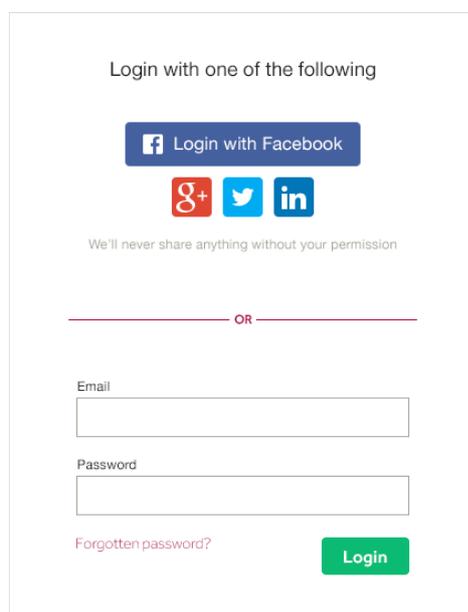
I vantaggi intrinseci che rendono questa tipologia di login sempre più diffusa sono diversi:

- *Rapidità nella registrazione*: il processo di registrazione è notevolmente accelerato, sgravando l'utente dal ripetitivo e noioso compito di dover compilare un form per la registrazione dei dati per ogni nuovo sito web. E' ormai noto che un numero inferiore di click che possono portare allo stesso obiettivo, è un importante incentivo per garantire un numero sensibilmente maggiore di conversioni.

- *Accesso unificato a molteplici siti:* permette all'utente di non dover necessariamente ricordare le credenziali di accesso a un numero svariato di siti.
- *Bontà dei dati:* spesso per velocizzare la compilazione dei moduli di registrazione gli utenti finali inseriscono, spesso, informazioni incorrette o incomplete. Recuperare le informazioni da profili social garantisce una possibilità maggiore di reperire informazioni complete e aggiornate.

Per garantire il massimo livello di sicurezza fornibile, la maggior parte dei login sociali han scelto di adottare il protocollo di autorizzazione sicura *OAuth 2.0*.

A fronte di questo insieme di vantaggi abbiamo però anche il possibile svantaggio di fornire un insieme di informazioni personali per il login sul sito al quale abbiamo fornito l'autorizzazione finalizzate principalmente a operazioni di marketing o affini. Ad esempio, facendo login su un sito di vendita di elettrodomestici con un profilo Facebook nelle cui attività si evince una passione per la cucina, non bisogna stupirsi se fra gli articoli proposti in promozione, vi saranno robot da cucina.



The image shows a login interface with the heading "Login with one of the following". Below this, there are four social media login buttons: "Login with Facebook" (blue), and buttons for Google+ (red), Twitter (blue), and LinkedIn (blue). Below these buttons is the text "We'll never share anything without your permission". A horizontal line with "OR" in the center separates the social login options from a traditional login form. The form consists of two input fields: "Email" and "Password". Below the "Email" field is a link for "Forgotten password?". At the bottom right of the form is a green "Login" button.

**Figura 2.1 - Esempio di Social Login (Intenthq)**

Solitamente, la scelta delle informazioni richieste viene formulata in fase di creazione dei pulsanti e modulata in base alle esigenze. Allo stesso modo l'utente, come detto inizialmente, viene informato su quali informazioni riceverà il sito

integrante questa tipologia di login, richiedendo un consenso formale al loro utilizzo. In un secondo momento, l'utente può comunque decidere di modificare la propria scelta e ritirare il consenso all'applicazione di ricevere queste informazioni, senza però garantire la rimozione delle precedenti ormai fornite.

Pro e contro di un mondo digitale in cui gli utenti hanno bisogno di accedere sempre più velocemente ai propri servizi e sono sempre più disinteressati a fornire le proprie informazioni personali.

### 2.1.1 Permessi legati al Facebook Login

Quando un utente sceglie di accedere ad una applicazione attraverso il Facebook Login, è possibile, tramite una richiesta esplicita, accedere a un sottoinsieme di dati memorizzati dall'utente stesso all'interno del social network. Per accedervi l'utente, deve però fornire un'autorizzazione che ne confermi la consultazione e l'utilizzo.

In base alle impostazioni di privacy definite dall'utente e ai permessi richiesti durante il login, sarà possibile accedere ad un insieme più o meno ampio di informazioni.

I permessi di autorizzazione sono richiesti sotto forma di stringhe che sono passate durante la richiesta di accesso o l'esecuzione di una chiamata API.

Le informazioni contenute all'interno del profilo pubblico sono richieste implicitamente e sono composte dal seguente elenco:

- *id*: identificativo dell'utente, composto da sole cifre
- *name*: nome completo dell'utente
- *first\_name*: il nome dell'utente
- *last\_name*: il cognome dell'utente
- *link*: l'url al profilo Facebook dell'utente
- *gender*: il sesso dell'utente
- *locale*: la lingua impostata dall'utente
- *timezone*: il fuso orario dell'utente, basato sulla sua nazionalità
- *updated\_time*: un timestamp relativo all'ultimo aggiornamento delle informazioni dell'utente
- *verified*: definisce se un utente ha verificato o meno la sua identità

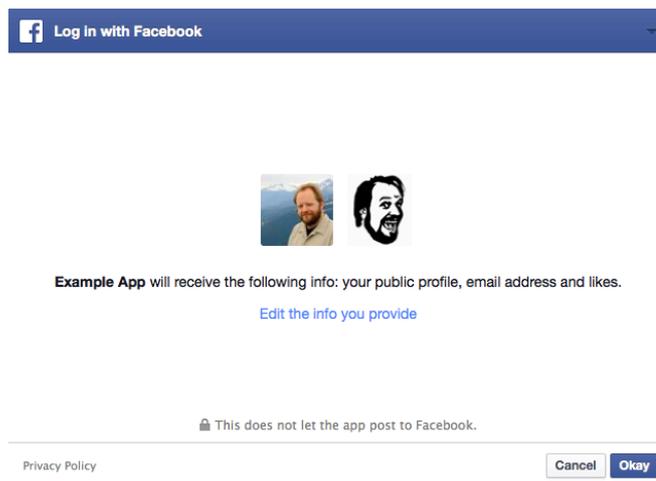
Oltre a questo insieme di dati è possibile richiedere, ad esempio, l'indirizzo *email*, l'insieme degli amici (*user\_friends*), i like espressi (*user\_likes*), le azioni

compiute (*user\_actions*) con gli oggetti standard come libri, film o musica, e non come le *custom stories*, o gli eventi a cui l'utente ha confermato la sua partecipazione (*user\_events*).

Queste autorizzazioni non possono essere definite come opzionali durante l'operazione di login. Pertanto è consigliabile definire eventuali richieste non strettamente necessarie durante questa fase, ma riconducibili ad altre operazioni, nel momento opportuno.

Inoltre, l'utilizzo pubblico di applicazioni richiedenti permessi ulteriori alle informazioni base dell'utente, se non per scopo di test dei servizi, richiede una fase di accettazione da parte di Facebook, durante la quale vengono richieste allo sviluppatore informazioni dettagliate sui motivi di utilizzo e sulla procedura di richiesta, accompagnando questi dati con la produzione di screenshot correlati alle azioni.

Una volta approvata e pubblicata l'applicazione ed inserita in produzione, la richiesta di consenso all'utente, avviene attraverso una finestra di dialogo all'interno della quale vengono elencati tutti i permessi richiesti e dove viene fornito un collegamento alla pagina di modifica dati, consentendoci di aggiornarli prima della consegna di questi all'app.



**Figura 2.2 - Accettazione App Facebook Login**

Unico permesso importante al punto da richiedere una finestra di dialogo dedicata, è la richiesta di *publish\_actions* tramite la quale l'applicazione può pubblicare, durante la propria esecuzione, delle notizie sulla timeline dell'utente.

L'insieme dei permessi richiesti da un'applicazione ad un utente possono cambiare nel tempo, e l'aggiunta di nuovi, prevede la richiesta di una nuova approvazione e la possibilità da parte dell'utente di rimuovere il consenso all'applicazione e la cancellazione di tutte le attività compiute con questa.

## **2.2 Facebook App**

Nel maggio del 2007 Facebook ha introdotto un'importante innovazione, la Facebook Platform, un framework dedicato agli sviluppatori per creare applicazioni capaci di interagire con le funzionalità alla base del core di Facebook. Per dimostrarne l'utilizzo e le potenzialità, Facebook stesso rilasciò e promosse le sue prime applicazioni tutt'ora utilizzabili, come *Events* grazie alla quale è possibile condividere informazioni sugli eventi a cui gli utenti sono stati invitati a partecipare, oppure *Video*, dedicato alla condivisione di filmati.

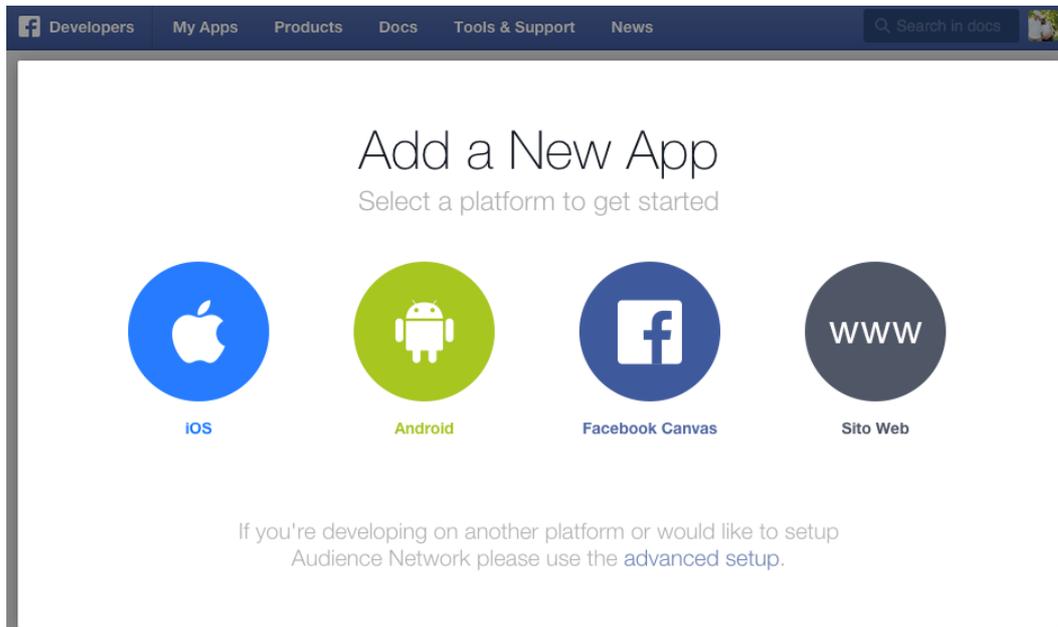
Per la produzione delle applicazioni, venne inizialmente prodotto e distribuito un particolare linguaggio di markup, chiamato *FBML (Facebook Markup Language)*. La svolta importante è stata poi conseguita con la pubblicazione delle Graph API e del protocollo Open Graph, grazie ai quali è possibile consultare i dati contenuti all'interno di Facebook come un grafo composto di nodi e collegamenti, descriventi il mondo delle nostre informazioni e delle nostre caratteristiche personali, come gusti, abitudini e passioni.

Per la consultazione di queste nuove API, Facebook ha rilasciato diverse SDK, fra le quali le librerie per lo sviluppo web con i linguaggi Javascript e PHP, e per gli ambienti mobile come iOS e Android, decidendo di deprecare il linguaggio FBML.

L'utilizzo e l'integrazione delle librerie sopra citate all'interno di un progetto, necessitano l'identificazione di quest'ultimo con i dati associati ad una applicazione Facebook, passaggio approfondito nel prossimo capitolo.

## 2.2.1 Pubblicazione di un'applicazione

Per la pubblicazione di un'applicazione Facebook, è sufficiente collegarsi alla pagina dedicata agli sviluppatori <http://developers.facebook.com> e seguire gli step suggeriti per la creazione guidata.



**Figura 2.3 - Creazione Facebook App. (Facebook)**

Una volta definita la piattaforma che dovrà ospitare l'applicazione, come iOS, Android, Canvas o Web page, è sufficiente definire il *Display Name*, ovvero il nome dell'app o del sito internet che dovrà contenerla, il *Namespace* identificativo univoco dell'app, e la *categoria* dell'applicazione, come cibo, giochi o istruzione. Terminata questa prima fase, vengono resi disponibili l'*App ID* e l'*App Secret* collegati all'applicazione, informazioni necessarie per l'inizializzazione delle librerie SDK.

Completata questa parte di definizione, tramite la dashboard dedicata è possibile:

- gestire l'applicazione aggiungendo o modificando gli utenti collegati all'applicazione con i rispettivi ruoli (*administrators*, *developers*, *testers*, *insight users*);
- modificare le impostazioni generiche d'utilizzo ed aggiungere eventuali piattaforme di utilizzo;
- creare *Custom Stories* tramite la voce di menu *Open Graph*;
- controllare eventuali sistemi di pagamento collegati all'applicazione;

- includere le pubblicità di Facebook all'interno delle proprie app tramite la piattaforma le inserzioni
- includere la pubblicità di Facebook direttamente nelle proprie app grazie alla piattaforma *Audience Network*;
- verificare le analitiche di utilizzo dell'app grazie alla sezione *Insights*.

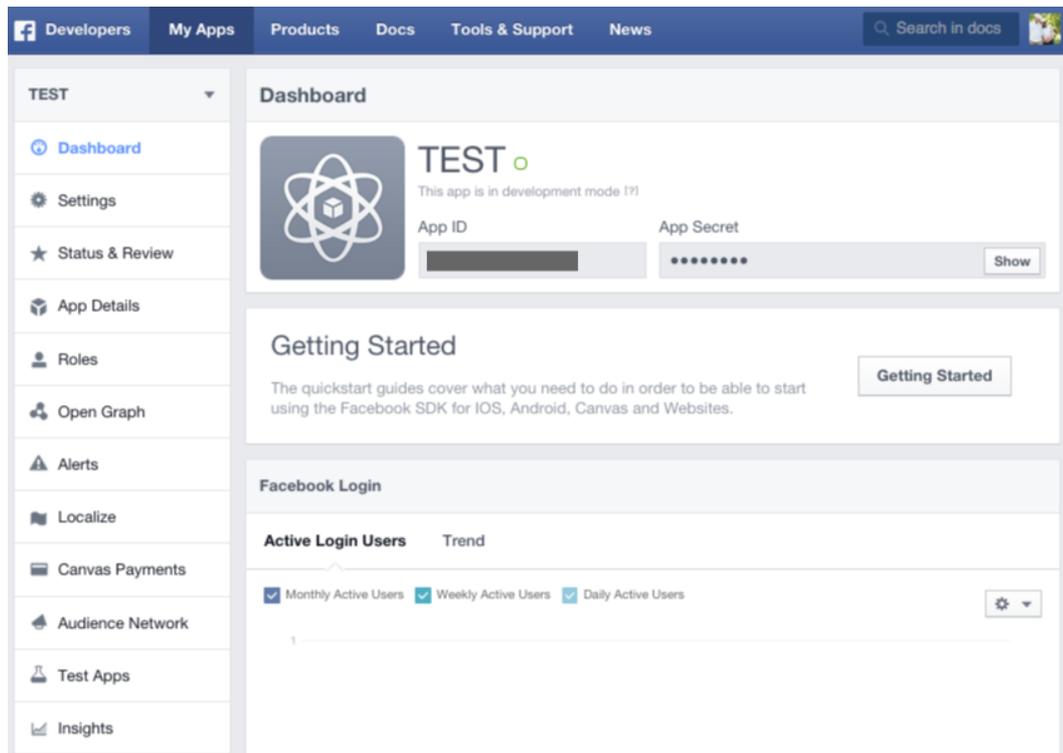


Figura 2.4 - Developer App dashboard (Facebook)

## 2.3 Facebook Social Plugin

I Social plugin, sviluppati da Facebook come proprie estensioni, forniscono agli sviluppatori dei set di codice configurabili e reperibili in modo intuitivo ed automatizzato grazie ad un'interfaccia contenuta nell'*Area Developer* di Facebook, delle integrazioni per spostare una parte dell'esperienza social all'interno del proprio sito web.

Fra questi vi sono, ad esempio, pulsanti come *Like*, *Share* e *Follow* e integrazioni più complesse come quella riguardante i *Commenti* o le aree di rassegna dei *Feed*. Di seguito saranno approfonditi i più utilizzati e d'interesse per il progetto di tesi.

### 2.3.1 Like Button e Like Box

Il *Like Button* è il modo più rapido per consentire alle persone di condividere un contenuto con i propri amici. Tramite un click, si può semplicemente esprimere un'intenzione rispetto a una porzione di contenuto web, condividendola in modo automatico su Facebook.

A differenza del button invece, il *Like Box*, è esclusivamente dedicato alle Pagine Facebook, ovvero dedicato a mostrare e permettere l'azione Like su di una pagina presente sul Social Network.

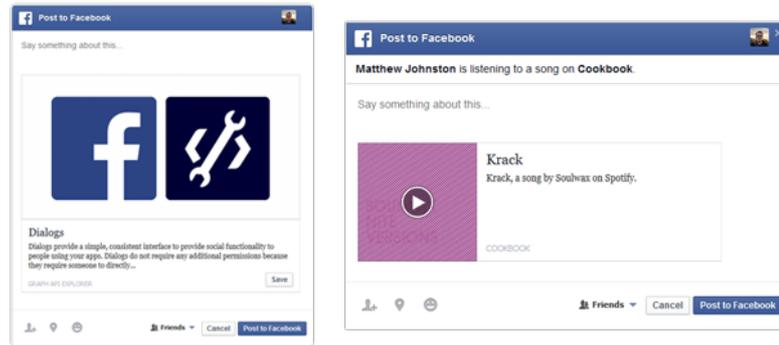
Grazie a questo plugin, è quindi permesso agli amministratori delle pagine, di promuoverle semplicemente includendo, attraverso la creazione di snippet automatica fornita da Facebook, una porzione di codice all'interno del proprio sito web.



*Figura 2.5 - Like Box (Facebook)*

### 2.3.2 Share Button e Share Dialog

A differenza del Like Button, tramite il quale, data la sua semplicità, è possibile eseguire soltanto l'azione di interesse per un contenuto, senza però poter mostrare questa operazione nella propria *timeline*, grazie al pulsante di *Share* non solo potremmo pubblicare questo contenuto all'interno del proprio sito, ma sarà possibile farlo anche sulla bacheca di un amico o inviarlo come messaggio privato.



**Figura 2.6 - Share Dialog (Facebook)**

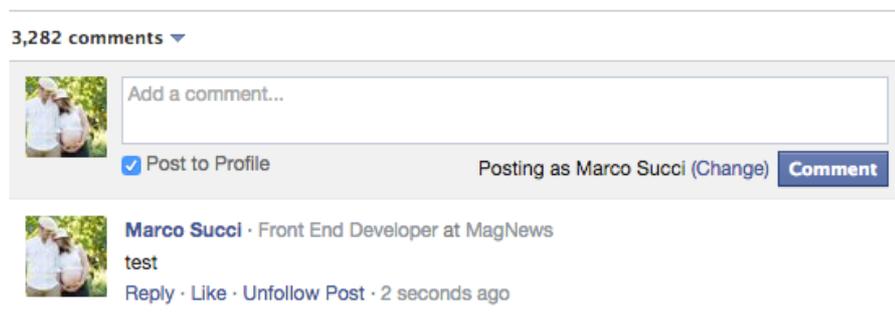
Per estendere le funzionalità di questo pulsante agli oggetti descritti tramite Open Graph, Facebook ha reso disponibile lo *Share Dialog*, ovvero un'interfaccia, modificabile e personalizzabile con le azioni scelte, con la quale è possibile pubblicare sulla propria timeline l'esecuzione di un'azione predefinita, come ad esempio, l'ascolto di una canzone.

### 2.3.3 Comments Plugin

Il *Comments plugin* è un utile strumento per permettere l'inserimento di commenti all'interno di una pagina web. A caratterizzare però questo plugin vi è, come consuetudine, la facilità nell'autenticazione degli utenti data dal login con Facebook e inoltre la possibilità di poter moderare in commenti prima della loro pubblicazione, previo consenso dell'amministratore, e la possibilità di ordinare i commenti non solo per data d'inserimento, ma anche per rilevanza sociale.

Questo ultimo tipo di ordinamento, caratteristica distintiva rispetto agli usuali sistema di commenti, è basato non solo sul numero di risposte al singolo commento, ma anche in base al numero di like ricevuti e di coinvolgimenti ottenuti dagli amici in termini di letture. Infatti, marcando, nel momento dell'invio del commento, l'opzione "*Post to profile*", questa azione verrà mostrata agli amici nel *news feed* e nel proprio *activity feed*.

In caso quindi venga effettuata la scelta di mostrare i commenti con questa tipologia di ordinamento, quelli riconosciuti come spam, vengono automaticamente rimossi da Facebook.



*Figura 2.7 - Comments plugin (Facebook)*

### 2.3.4 Activity e Recommendations Feed

Due plugin prodotti per mantenere costantemente aggiornati sull'attività sociale che avviene all'interno del nostro sito, come like o share, sono l'Activity e il Recommendations Feed. Entrambi aggregano le ultime azioni compiute in relazione ai contenuti web e li mostrano evidenziando chi o quanti hanno compiuto una specifica azione.

La differenza principale fra questi è che l'Activity feed mostra gli ultimi di questi contenuti sui quali è stata compiuta un'azione mentre il Recommendations espone per primi gli oggetti che hanno ricevuto maggiori interazioni.

## 2.4 Facebook Graph Search

Una delle ultime e più importanti novità presentate da Facebook è senza dubbio la *Facebook Graph Search*, presentata a Gennaio 2013 e attualmente disponibile unicamente per gli utenti statunitensi.

La Graph Search è un motore di ricerca, sostitutiva al sistema di ricerca attualmente presente, di tipo semantico poiché basato non solo su keyword-search ma su di un generatore d'inferenze in linguaggio naturale, capace di interrogare il grafo sociale generando associazioni fra le parole inserite e gli oggetti Open Graph.

Per aiutare l'utente finale nella generazione d'interrogazioni, la barra di ricerca fornisce inoltre una funzione di auto completamento, in grado di suggerire non

solo singole parole, ma anche frasi di senso compiuto analizzando ciò che si sta scrivendo.

Grazie a questo nuovo strumento, digitando parole come *movies*, *music*, *book*, *games* o *restaurant* queste saranno riconosciute automaticamente e associate a categorie di oggetti precise e ben tipizzate, fornendo quindi un significato ben più ampio alla ricerca stessa.

Ma forse ancor più importante, e strettamente legato al concetto di grafo sociale, sarà possibile ricevere risultati filtrati in base ai “mi piace” dei nostri amici o di alcuni sotto gruppi di loro come i colleghi, i compagni di classe o i concittadini.

Ecco quindi che potremmo chiedere a Facebook quali ristoranti vicino a dove ci troviamo sono piaciuti ai nostri amici o quale è il gruppo musicale preferito dai compagni di università.

Oppure più semplicemente, affidandosi agli algoritmi nativi di Facebook, potremmo richiedere quali libri, film o canzoni potrebbero piacermi, in base all’analisi del mio comportamento sociale.

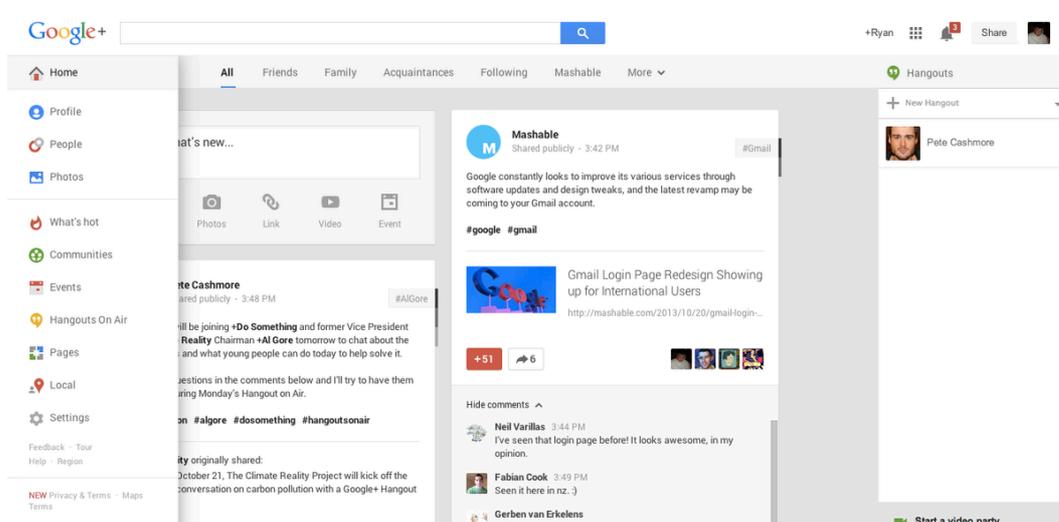


**Figura 2.8 - Facebook Graph Search**

Infine, per garantire una maggior completezza nei risultati della ricerca, oltre che a fornire quelli prodotti da Facebook, laddove individuati, all’insieme degli elementi restituiti saranno aggiunti anche i risultati di ricerca provenienti dal web, grazie alla collaborazione con il noto motore di ricerca *Bing* prodotto da Microsoft.

## 2.5 Google+, MyBusiness e Schema.org

Google+, social network made in Google nato nel 2011, dopo una breve parentesi di entusiasmo iniziale che ha portato a un importante picco di iscrizioni, è stato poi tendenzialmente snobbato poiché considerato poco innovativo. A dimostrarlo è senza dubbio il fatto che non è mai riuscito a impensierire il numero di utenti registrati o ad avvicinarsi al volume di attività social prodotta dai principali nomi della scena social come Facebook o Twitter.



*Figura 2.9 - Interfaccia home page(Google+)*

Di recente però, Google ha deciso di sfruttare, per aumentare il numero di utenti presenti e attivi sul proprio social network, il potere dato dall'insieme dei servizi da lui prodotti.

Un esempio evidente è dato dalla limitazione introdotta sul portale YouTube alla possibilità di scrivere e pubblicare commenti, diventata una funzionalità esclusiva riservata agli utenti Google+.

Altra importante introduzione è stata la possibilità di inviare email agli utenti Google+, sfruttando l'applicativo Gmail, consentendo di selezionare come destinatari anche utenti di cui non conosciamo l'indirizzo di posta ma con cui abbiamo un'interazione sociale o semplicemente seguiamo all'interno del social, come ad esempio una star del cinema o un musicista che apprezziamo.

Un insieme di cambiamenti che sono stati spesso considerati come forzature e quindi poco apprezzate dai suoi utenti e che continuano a sottolineare uno sforzo per cercare di far decollare un progetto nato, probabilmente, in ritardo e con poche funzionalità contraddistinguibili.

Una particolare nota va infine dedicata alla recente azione di unificazione, operata da Google, fra Google+ e lo strumento Google Places, servizio dedicato alla definizione di geo localizzazione di attività commerciali. L'unione dei due servizi ha prodotto il sistema Google MyBusiness, dove, tramite un processo di riconoscimento e autorizzazione basato su cartolina postale, è possibile descrivere la propria attività e collegarla ad un proprio sito web e ad un indirizzo rintracciabile tramite le mappe di Google Maps.

Un'ulteriore operazione volta a favorire, in questo caso, l'iscrizione di nuove aziende al social in suo possesso, promettendo in cambio un miglior punteggio page ranking e la garanzia di un miglior di un risultato di ricerca più completo.

Per lo stesso motivo, in collaborazione con altri importanti motori di ricerca (Bing, Yahoo, Yandex, ...), Google ha partecipato alla realizzazione di uno schema di markup dei dati detto Schema.org.

**Little Water Cantina - Eastlake - Seattle, WA**  
[www.yelp.com](http://www.yelp.com) > Restaurants > Mexican  
 ★★★★★ 90 reviews - Price range: \$\$  
 90 Reviews of **Little Water Cantina** "Three things are on my list when I eat out: great food, atmosphere, and

**Vegetarian Vegan Pizza No Cheese) Recipe - Food.com - 248865**  
[www.food.com/recipe/vegetarian-vegan-pizza-no-c...](http://www.food.com/recipe/vegetarian-vegan-pizza-no-c...)  
 ★★★★★ 2 reviews - 1 hr 32 mins - 242.9 cal  
 Aug 26, 2007 - This is from my dad, who developed some **vegan recipes** esn't have any cheese, and you

**Leonard Cohen – Free listening, videos, concerts, stats, & pictures at ...**  
[www.last.fm/music/Leonard+Cohen](http://www.last.fm/music/Leonard+Cohen)  
 Watch videos & listen to **Leonard Cohen**: Suzanne, Hallelujah & more, plus 132 pictures. **Leonard Cohen**, (born September 21, 1934 in Montréal, Quebec, ...

Track	Duration
<a href="#">Suzanne</a>	♫ 3:48
<a href="#">The Darkness</a>	♫ 4:29
<a href="#">Going Home</a>	♫ 3:51
<a href="#">Hallelujah</a>	♫ 6:12

*Figura 2.10 - Rich Snippet (Google)*

Basato su RDFa, microdati e microformati, consente di definire una vasta gamma di elementi e proprietà all'interno di una pagina web, arricchendone il significato e permettendo di ottenere risultati di ricerca differenziati e specifici in base ai tipi di oggetti definiti.

Tramite questo schema di dati, è quindi possibile definire una pagina web, come un prodotto, ad esempio una fotocamera digitale, marcando all'interno del codice ciò che può essere il nome, la descrizione, l'immagine ed il prezzo consentendo di visualizzare queste informazioni all'interno del risultato di ricerca prodotto dai motori di ricerca.

Altre tipologie di oggetti descrivibili tramite schema.org, riconoscibili da Google sono: recensioni, persone, prodotti, attività commerciali e organizzazioni, ricette, eventi e musica.

## 3. Open Graph Protocol

### 3.1 Dal “Mi Piace” all’Open Graph

Nei suoi dieci anni di presenza Facebook non solo è riuscito a raggiungere circa un miliardo di utenti attivi nel mondo e più di settanta traduzioni, ma è anche stato capace di stravolgere le nostre abitudini quotidiane, fornendo un senso decisamente più ampio ad una semplice espressione come “mi piace”.

Il “like” è di fatto stato uno delle basi fondanti di questo Social Network, pensato non solo per fornire ai suoi utenti un modo per esprimere una propria propensione, ma sviluppato per essere uno strumento per conoscere e comprendere i gusti e gli orientamenti di un numero sempre maggiore di utenti.

Gli esperti SEO (*Search Engine Optimization*) hanno da subito scommesso molto sui Social Network, vedendo in loro una nuova grande possibilità per generare un maggior traffico d’entrata verso i siti web da loro analizzati.

Superato però l’entusiasmo iniziale è apparso altrettanto presto evidente a molti che il “Facebook Like” non fosse in grado di generare grosse quantità di traffico referral.

Al contrario del “link”, che avrebbe dovuto in qualche modo sostituire, il “like” non riusciva, infatti, a portare un valore aggiunto capace di aumentare nel tempo.

Dalla necessità quindi da parte di Facebook di fornire uno strumento in grado di generare un traffico maggiore senza richiedere azioni aggiuntive agli utenti e dalla volontà di favorire un maggior valorizzare gli oggetti esterni condivisi su Facebook, è stato sviluppato il protocollo Open Graph.

## 3.2 L'Open Graph in pillole

Il protocollo Open Graph, essendo stato inizialmente basato sulla specifica RDFa (*Resource Description Framework in Attributes*), consente, tramite l'utilizzo di meta-tag dichiarati all'interno del tag <head> di una pagina web, di rappresentare quest'ultima sotto la forma di un oggetto.

Questa inclusione arricchisce il significato stesso della pagina e ne permette l'integrazione nel grafo sociale.

Il markup di base prevede la definizione di alcuni attributi essenziali, come: *type*, *title*, *URL*, *image* e *description*, ma il protocollo permette anche altri tipi maggiormente strutturati di oggetti come *notizie*, *località*, *tassonomie di tipi*, *articoli*, ecc.

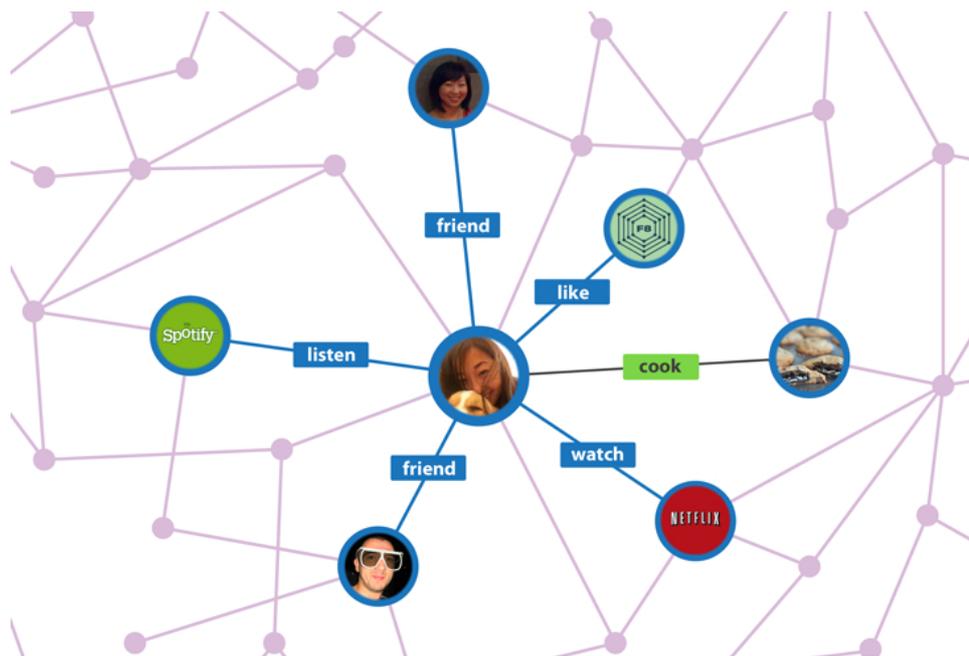


Figura 3.1 - The Social Graph (Business Insider)

In questo modo cliccando *Like* all'interno di una pagina web integrante il protocollo, Facebook non otterrà solo un semplice link ad una pagina web, ma un oggetto ben definito di una tipologia specifica.

Inoltre, questo click, ha il potere di generare una connessione fra la pagina e l'utente che l'ha compiuto. La pagina, infatti, apparirà nella sezione "Like e

interessi” dell’utente, fornendo eventualmente la possibilità di pubblicare aggiornamenti agli utenti.

I dati strutturati forniti a Facebook tramite l’Open Graph Protocol determineranno quindi come la pagina web apparirà quando interagirà con Facebook stesso.

Per favorirne un’ampia diffusione, promuovendo l’inserimento dei meta all’interno delle pagine web da parte dei web developer, Facebook ha reso il protocollo libero a tutti.

Per questo altri importanti social network come Twitter, Google+, LinkedIn e Pinterest leggono e riconoscono, alcuni in toto altri in parte, questi dati, riconoscendo anche loro le informazioni descrittive fornite tramite gli og tag.

Il costante aumento di piattaforme di blogging e di strumenti CMS, e il crescente numero di sviluppatori attenti a integrare il protocollo nelle loro applicazioni, sta quindi rendendo possibile agli utenti di Facebook di interagire nei loro profili con oggetti ricchi d’informazioni e consentendo al Social Network di mostrare ai propri utenti a chi piacciono gli stessi oggetti o quali potrebbero interessargli per affinità.

Con il protocollo Open Graph, Facebook ha quindi fornito uno strumento che consente di riconoscere e comprendere se la pagina web, con il quale l’utente sta interagendo, descriva in realtà un oggetto, specificatamente tipizzato, come un film, un libro o una ricetta, ecc...

Si può quindi asserire che Facebook sta svolgendo un ruolo importante nel processo di costruzione di un Web Semantico.

## ***3.3 Lo schema dei meta-dati***

### **3.3.1 I meta-dati di base**

Per consentire di trasformare le proprie pagine web in oggetti del grafo, è necessario innanzitutto aggiungere dei meta-dati di base tramite dei tag <meta> nell’<head> delle pagine.

Queste quattro proprietà da aggiungere in ogni pagina sono:

- `og:title` – Il titolo dell’oggetto così come comparirà nel grafo, ad esempio “Matrix”.

- `og:type` – Il tipo di oggetto, ad esempio “video.movie” nel caso di un film. In base al tipo definito potranno essere specificate informazioni aggiuntive.
- `og:image` – L’URL di un’immagine che dovrebbe rappresentare l’oggetto del grafo.
- `og:url` – L’URL dell’oggetto che sarà utilizzato come ID permanente nel grafo, ad esempio “http://www.imdb.com/title/tt0133093/”.

Si ha quindi che un esempio di meta-dati di base definiti per l’oggetto “Matrix” utilizzati dal sito IMDB è definito, all’interno del codice, come segue:

```
<html prefix="og: http://ogp.me/ns#">
<head>
  <title>Matrix (1999) - IMDb</title>
  <meta property="og:title" content="Matrix" />
  <meta property="og:type" content="video.movie" />
  <meta property="og:url" content="http://www.imdb.com/title/ tt0133093/" />
  <meta property="og:image" content="http://ia.media-imdb.com/images/matrix.jpg" />
  ...
</head>
...
</html>
```

### 3.3.2 I meta-dati opzionali

I seguenti meta-dati sono invece opzionali per ogni oggetto, ma comunque raccomandati:

- `og:audio` – L’URL di un file audio che rappresenta l’oggetto.
- `og:description` – Una breve descrizione dell’oggetto.
- `og:determiner` – L’articolo che appare prima del titolo di un oggetto in una frase. Al momento è possibile scegliere solo fra i termini a, an, the, "", auto.
- `og:locale` – La definizione della lingua locale scelta. Il default è `en_US`.
- `og:locale:alternate` – Un array di altre lingue in cui questa pagina è disponibile.
- `og:site_name` – Il nome del sito che incapsula l’oggetto Open Graph, ad esempio "IMDb".
- `og:video` – L’URL di un video che rappresenta l’oggetto.

```
<meta property="og:audio" content="http://example.com/bond/theme.mp3" />
<meta property="og:description" content="Sean Connery found fame and fortune as
the suave, sophisticated British agent, James Bond." />
<meta property="og:determiner" content="the" />
<meta property="og:locale" content="en_GB" />
<meta property="og:locale:alternate" content="fr_FR" />
<meta property="og:locale:alternate" content="es_ES" />
<meta property="og:site_name" content="IMDb" />
<meta property="og:video" content="http://example.com/bond/trailer.swf" />
```

### 3.3.3 I meta-dati strutturati

La definizione di alcune proprietà degli oggetti Open Graph permette di specificare ulteriori meta-data che ne aumentano l'espressività. Questi sono descritti come i meta-data visti in precedenza grazie ai parametri `property` e `content`, ma nello specifico il campo `property` si presenterà con un ulteriore simbolo .:

La proprietà `og:image` consente le seguenti ulteriori proprietà strutturate:

- `og:image:url` – Identico a `og:image`.
- `og:image:secure_url` – Un URL alternativo nel caso in cui la pagina utilizzi il protocollo sicuro HTTPS.
- `og:image:type` – Un MIME type per l'immagine.
- `og:image:width` – La dimensione in pixel della larghezza dell'immagine.
- `og:image:height` - La dimensione in pixel della altezza dell'immagine.

Un esempio di definizione completa per un'immagine prevede:

```
<meta property="og:image" content="http://example.com/ogp.jpg" />
<meta property="og:image:secure_url" content="https://s.exp.com/ogp.jpg" />
<meta property="og:image:type" content="image/jpeg" />
<meta property="og:image:width" content="400" />
<meta property="og:image:height" content="300" />
```

Il tag `og:video` ha le stesse proprietà di `og:image` . Eccone un esempio:

```
<meta property="og:video" content="http://example.com/movie.swf" />
<meta property="og:video:secure_url" content="https://s.exp.com/movie.swf" />
```

```
<meta property="og:video:type" content="application/x-shockwave-flash" />
<meta property="og:video:width" content="400" />
<meta property="og:video:height" content="300" />
```

Il tag `og:audio` ha solo 3 proprietà disponibili (considerando che le dimensioni non sono applicabili ad una traccia audio):

```
<meta property="og:audio" content="http://example.com/sound.mp3" />
<meta property="og:audio:secure_url" content="https://s.exp.com/sound.mp3" />
<meta property="og:audio:type" content="audio/mpeg" />
```

## 3.4 L'evoluzione (Video, libri, musica...)

### 3.4.1 Video

- Namespace URI: `http://ogp.me/ns/video#`

Di seguito i descrittori dei tipi OG video:

`video.movie`

- `video:actor` - Array dei profili degli attori del film.
- `video:actor:role` - Una stringa contenente il ruolo di uno specifico attore del film.
- `video:director` - Array dei profili dei registi del film.
- `video:writer` - Array dei profili degli scrittori del film.
- `video:duration` - Un integer contenente la durata del film in secondi.
- `video:release_date` - La data di uscita del film.
- `video:tag` - Un array di string rappresentanti i tag associati al film.

`video.episode`

- `video:actor` - Identici a `video.movie`
- `video:actor:role`
- `video:director`
- `video:writer`
- `video:duration`
- `video:release_date`
- `video:tag`
- `video:series` - Un oggetto `video.tv_show` facente riferimento a una serie a cui l'episodio appartiene.

video.tv\_show

Uno show televisivo multi episodio associato ad una serie di oggetti di tipo video.movie.

video.other

Un video che non appartiene a nessuna categoria di video ma che può essere caratterizzato dagli stessi descrittori di video.movie.

### 3.4.2 Music

- Namespace URI: <http://ogp.me/ns/music#>

Di seguito i descrittori dei tipi OG music

music.song

- `music:duration` – Un dato di tipo int che definisce la durata in secondi della canzone.
- `music:album` – Un array di album contenenti la canzone.
- `music:album:disc` – Il numero del disco dell'album in cui è inserita la traccia.
- `music:album:track` – Il numero della traccia nell'album.
- `music:musician` – Un array con i profili dei musicisti che hanno composto la canzone.

music.album

- `music:song` – Oggetti di tipo music.song rappresentanti le canzoni di questo album.
- `music:song:disc` - Il numero del disco dell'album in cui è contenuta una canzone.
- `music:song:track` – il numero della traccia nell'album.
- `music:musician` – I profili dei musicisti che hanno composto la canzone.
- `music:release_date` – La data di pubblicazione dell'album.

music.playlist

- `music:song` – Identico ad uno degli oggetti contenuti in music.album
- `music:song:disc`
- `music:song:track`
- `music:creator` – Il profilo dell'utente che ha creato la playlist.

music.radio\_station

- `music:creator` – Il profilo del creatore della stazione radiofonica.

### 3.4.3 Book

- Namespace URI: `http://ogp.me/ns/book#`

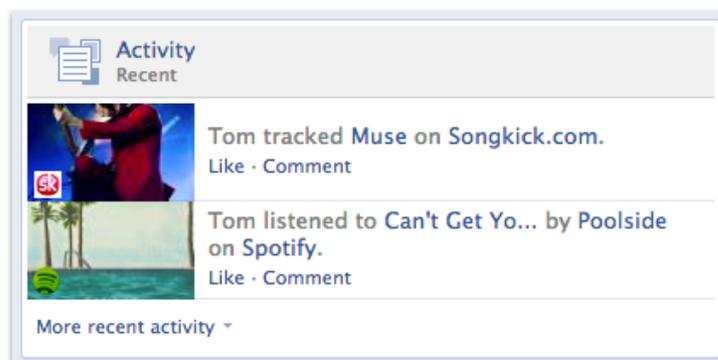
Di seguito i descrittori del tipo OG book

- `book:author` – Il profilo dell'autore che ha scritto il libro.
- `book:isbn` – Una stringa rappresentante il codice ISBN del libro.
- `book:release_date` – Data di uscita del libro.
- `book:tag` – Array di tag espressi in stringhe associati al libro.

## 3.5 Custom Stories

Tramite Open Graph è possibile descrivere, attraverso la definizione di un'applicazione, una storia. Una storia dettagliatamente personalizzabile, tramite la quale è possibile raccontare una determinata azione accaduta in un posto specifico, in un preciso momento, grazie alle informazioni descritte nei metatag Open Graph. Ma cos'è una storia?

Una storia è un insieme composto di quattro elementi principali, quattro protagonisti che raccontano e partecipano direttamente a un evento.



*Figura 3.2 - Esempi di Custom story Facebook*

Di seguito saranno analizzati questi quattro elementi prendendo in considerazione un esempio di storia descritto dalla frase:

“*Christopher ha finito di leggere The Name of the Wind on Goodreads*”.

- *L'attore*: è la persona che ha pubblicato la storia, in questo caso “Christopher”.
- *L'app*: è l'applicazione che pubblica la vicenda per conto dell'attore. Ogni storia è generata da un app e ogni storia include l'applicazione utilizzata per crearla.
- *L'azione*: è l'attività che svolge l'attore, nello specifico “ha finito di leggere”. Facebook fornisce una serie di azioni comuni che possono essere utilizzate per creare storie, come vedere, leggere o ascoltare oppure è possibile creare le proprie storie e azioni personalizzate.
- *L'oggetto*: è ciò con cui interagisce l'attore come ad esempio un libro, un film o un album musicale. Nell'esempio riportato è il libro “The Name of the Wind”.



**Figura 3.3 - Tripla di una Facebook story**

Gli oggetti possono essere creati in due modi diversi:

1. *Aggiungendo i metadati Open Graph* all'interno di una pagina web pubblicata in Internet. Come è già stato ampiamente descritto, grazie a questi meta-dati, ogni pagina web può divenire un oggetto coinvolgibile all'interno del Grafo Sociale;
2. Utilizzando le *Facebook Object API* per creare oggetti ospitati sui server di Facebook. Questi oggetti possono essere privati (di singoli utenti) o globali, nel contesto di una app.

Nel caso in cui non sia già presente un'azione comune capace di soddisfare le proprie esigenze, è possibile creare il proprio tipo di storia personalizzata.

Ad esempio, se si sta costruendo un app che deve tener traccia delle arrampicate su roccia si potrebbe creare un'azione “scalata” in cui l'oggetto è una montagna.

La storia può includere una foto e la posizione geografica della montagna.

In questo modo con l'aggiunta di dati aggiuntivi è possibile creare nuove storie che descrivano le azioni e gli oggetti che più ci interessano, per poterli condividere con i propri amici, rendendo la propria applicazione, uno strumento di comunicazione verso gli altri.

## 4. Graph API

Il centro nevralgico di Facebook è il *Social Graph*, composto e alimentato dagli utenti che lo compongono e dalle azioni che quotidianamente intraprendono. I *nodi* che compongono questo particolare grafo rappresentano le entità, ovvero le persone, le pagine o le applicazioni, mentre gli *archi* equivalgono alle connessioni fra queste entità.

Ogni oggetto o entità presente e partecipa ad un'azione all'interno di Facebook, corrisponde ad un nodo del grafo sociale. Ogni azione compiuta da un'entità, è rappresentabile da un arco etichettato uscente dal nodo ad esso correlata. Questa etichetta viene usualmente chiamata *verbo*.

Per permettere agli sviluppatori di interagire ed interrogare questo grafo sociale, Facebook ha sviluppato e reso pubbliche, nell'aprile del 2010, le *Graph API*, un'interfaccia di programmazione, richiamabile attraverso diverse tipologie di linguaggi, che hanno raggiunto nell'ottobre 2014 la versione 2.2.

Di seguito, verranno analizzati nel dettaglio le fondamenta di questo strumento e della struttura dei risultati prodotti.

### 4.1 REST API

Con il termine REST (*Representational State Transfer*), il cui nome è stato attribuito nel 2000 da Roy Fielding [FIE00], uno dei principali autori delle specifiche dell'*Hypertext Transfer Protocol* (HTTP), è definito un insieme di principi architetturali finalizzato alla progettazione di un sistema.

Non riferendosi a un sistema concreto e definito, e non trattandosi ancora di uno standard, rappresenta nella pratica uno stile architetturale, ovvero un insieme di linee guida per la realizzazione di un'architettura di sistema.

Questo stile architetturale è diventato largamente utilizzato, negli ultimi anni, per la realizzazione di Web service.

Alla sua base vi sono 5 principi, che saranno successivamente approfonditi:

1. *Identificazione delle risorse*
2. *Utilizzo esplicito dei metodi HTTP*
3. *Risorse autodescrittive*
4. *Collegamenti tra risorse*
5. *Comunicazione senza stato*

Le risorse sono gli elementi fondamentali su cui si basano i Web Service RESTful, a differenza di quelli SOAP-Oriented, incentrati sul concetto di chiamata remota.

Queste risorse, rappresentabili come un qualsiasi elemento elaborabile (ad esempio un libro, un articolo, ecc...), sono identificabili utilizzando il metodo fornito dallo schema URI (*Uniform Resource Identifier*).

Alcuni esempi di possibili identificatori di risorse potrebbero quindi essere:

- <http://www.myapp.com/clienti/1234>
- <http://www.myapp.com/ordini/2011/98765>
- <http://www.myapp.com/ordini/2011>
- <http://www.myapp.com/prodotti?colore=rosso>

Dove i primi due identificano rispettivamente un determinato cliente e un ordine mentre il quarto URI identifica l'insieme degli ordini del 2011 e l'ultimo URI denota l'insieme dei prodotti di colore rosso.

Per favorire una facile comprensione sul tipo di risorsa descritta, gli URI devono essere *autoesplicativi*, cercando di preferire l'utilizzo di nomi a quello dei verbi, contenerne la lunghezza, favorire uno schema posizionale alla presenza di argomenti, evitare le estensioni legate all'implementazione del Web Service.

Specificato il metodo con cui individuare una risorsa, occorre definire il meccanismo per indicare quali operazioni è possibile compiere su di esse.

Per questo compito, viene utilizzato l'uso esplicito dei metodi HTTP, cioè **GET**, **POST**, **PUT** e **DELETE**.

REST, quindi, mappa uno a uno le tipiche operazioni CRUD (create, read, update e delete) ai metodi HTTP:

- POST → Create (Crea una nuova risorsa)
- GET → Read (Ottiene una risorsa esistente)
- PUT → Update (Aggiorna una risorsa o ne modifica lo stato)

- DELETE → Delete (Elimina una risorsa)

Per quanto concerne la modalità di rappresentazione di una risorsa, i principi REST, non pongono nessun vincolo, consentendo di utilizzare il formato preferito, senza essere obbligati a seguire uno standard, anche se è consigliabile utilizzarne uno per semplificare l'interazione con i client.

Il tipo di rappresentazione inviata dal Web Service al client è indicato nella stessa risposta HTTP tramite un tipo MIME, così come avviene nella classica comunicazione tra Web server e browser.

Ulteriore aspetto legato ai principi REST, riguarda la necessità che le risorse siano tra esse collegate, attraverso la definizione di link ipertestuali.

In sostanza, tutto quello che un client deve sapere su una risorsa e sulle relative risorse ad essa correlate, va inserito nella sua rappresentazione o deve essere accessibile tramite collegamenti ipertestuali, in modo che il client possa accedere alle risorse correlate seguendo semplicemente i collegamenti contenuti nella rappresentazione della risorsa corrente.

Ultimo aspetto contenuto nei principi fondamentali di REST, è quello riguardante la *comunicazione stateless*, nota tematica agli sviluppatori Web, essendo infatti una delle caratteristiche principali del protocollo HTTP, nel quale ciascuna richiesta non ha alcuna relazione con le richieste precedenti e successive.

Il medesimo principio è applicabile ad un Web Service RESTful, dove quindi le interazioni tra client e server devono essere senza stato.

A motivare questa scelta vi è principalmente la scalabilità: mantenere lo stato di una sessione, infatti, ha un costo in termini di risorse sul server. Inoltre l'eventuale aumento del numero di client potrebbe rendere tale costo insostenibile.

Infine, con una comunicazione senza stato è possibile creare cluster di server che possono rispondere ai client senza vincoli sulla sessione corrente, ottimizzando le prestazioni globali dell'applicazione.

In conclusione, l'approccio REST tende quindi a conservare e ad esaltare le caratteristiche intrinseche del Web evidenziandone la predisposizione ad essere una piattaforma per l'elaborazione distribuita. Quindi, non è necessario aggiungere nulla a quanto è già esistente sul Web per consentire ad applicazioni remote di interagire.

## 4.2 JSON

*JSON*, acronimo di *JavaScript's Object Notation*, è un formato utilizzato per lo scambio dei dati in applicazioni client-server, basato sul linguaggio JavaScript Standard ECMA-262 3<sup>a</sup> edizione dicembre 1999 [WIK-JSON].

JSON è costituito da due sole strutture:

1. *una collezione di coppie* (nome, valore). In diversi linguaggi, questa è considerabile come un *object*, un record, una struct, un dizionario, una hash table, una lista con chiavi o un array associativo;
2. *una lista ordinata di valori*, rappresentabile, in molti linguaggi, come una lista, un *array* o una sequenza.

La sua semplicità, ne ha decretato un rapido utilizzo, specialmente in applicativi AJAX, dove ha ormai soppiantato il formato XML/XSLT per i messaggi di risposta dei servizi *Web*, quindi anche nelle API, poiché più leggero, nonostante sia caratterizzato dal medesimo potere espressivo.

## 4.3 Autenticazione OAuth

*Open Authorization*, o più semplicemente *OAuth*, è un protocollo aperto per l'autorizzazione, mediante il quale, una qualsiasi tipologia di applicazione, desktop, web o mobile, può verificare l'identità delle entità coinvolte attraverso transazioni sicure.

Ideato inizialmente da Blaine Cook nel 2006, durante l'implementazione Twitter di *OpenID*, è stato successivamente rilasciato in due versioni: OAuth 1.0 (Hammer - Lahav 2010) e OAuth 2.0 (Hardt 2012).

A differenza dell'autenticazione basata sul tradizionale sistema *client-server*, l'idea proposta da OAuth è fondata sull'introduzione di un terzo ruolo: l'utente finale detto anche proprietario delle risorse (*resource owner*).

In questo modello il client, detto *consumer*, che non è il proprietario delle risorse ma è colui che agisce in suo nome, deve richiedere il permesso all'utente per accedere alle sue risorse.

Inoltre viene permesso al server di verificare non solo che il client abbia l'autorizzazione, da parte del proprietario delle risorse, ma anche l'identità del client che fa la richiesta.

Per consentire quindi al client di poter accedere alle risorse, in primo luogo questo deve ottenere il permesso dal proprietario della risorsa.

Questa autorizzazione viene espressa sotto forma di *token* (gettone) ed uno *shared-secret*.

Il cuore del protocollo è proprio il token, che permette di gestire gli accessi senza la condivisione di password e diversamente dalle credenziali dell'utente, può essere rilasciato con delle restrizioni e/o con durata limitata, nonché essere revocato in maniera indipendente dalla volontà del client.

Tipicamente, questi token sono emessi dal server dopo una richiesta esplicita dell'utente, in seguito ad un'autenticazione di quest'ultimo, la quale solitamente avviene attraverso l'inserimento di username e password.

Il metodo più comune con cui viene fornito questo token è l'utilizzo di una la redirectione HTTP e l'user-agent del resource owner.

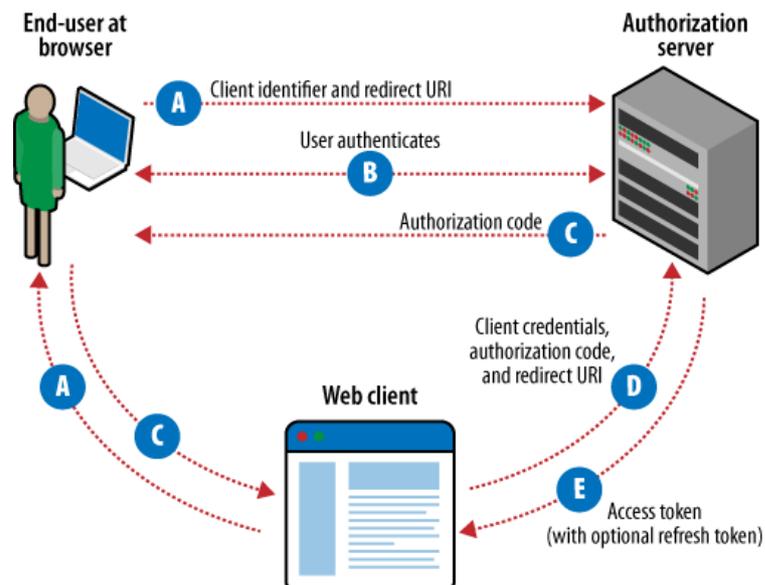
Questa metodologia si struttura in tre fasi:

1. il client ottiene un set di credenziali temporanee dal server sotto forma di un identificatore e un shared-secret. Le credenziali temporanee sono usate per identificare la richiesta di accesso durante tutto il processo di autenticazione;
2. il resource owner autorizza il server ad accogliere la richiesta di accesso da parte del client identificato dalle credenziali temporanee;
3. il client usa le credenziali temporanee per richiedere un set di token dal server, che gli permetteranno di accedere alle risorse protette dell'utente. A differenza quindi della classica autenticazione client-server, viene introdotto un terzo ruolo, quello del proprietario delle risorse, ovvero l'utente.

Una volta ottenuto il token da parte del client, le credenziali temporanee vengono revocate e per garantire maggiore sicurezza a queste viene associato un tempo di vita limitato [MAQ - 10].

Questo protocollo per l'autenticazione è diventato il più diffuso nelle integrazioni fra siti web e Social Network, come ad esempio Facebook o Twitter, per l'utilizzo dei rispettivi sistemi di autorizzazione.

La differenza principale presente fra le due versioni di *OAuth* insiste nell'obbligo, nel caso della seconda versione, di utilizzare una comunicazione sicura SSL. Solo in questo caso l'implementazione di *OAuth* è completa. Di fatti, in caso contrario, è necessario provvedere ad inserire meccanismi di protezione per assicurare l'*access token*.



**Figura 4.1 – Flusso di autenticazione OAuth**

In conclusione il protocollo *OAuth* oltre a semplificare le comunicazioni fra *client*, *server* e *content provider*, aumenta il livello globale di sicurezza durante la fase di autenticazione, prevenendo la propagazione attraverso di password.

## 4.4 Graph API nel dettaglio

Le Graph API sono riassumibili come un insieme di procedure atte a restituire una vista uniforme del grafo attraverso l'utilizzo di chiamate HTTP.

Tramite queste chiamate è possibile consultare le informazioni descritte da un sottoinsieme di nodi del grafo, come ad esempio i profili degli utenti, le loro foto o i loro eventi, e le connessioni che connettono fra di loro questi nodi, come ad esempio le relazioni di amicizia o i contenuti fra di loro condivisi.

Su questa logica è strutturato il funzionamento di qualsiasi comportamento tenuto su Facebook, costituendo questo il metodo stesso di lettura e scrittura dati del social network.

Si evince così che ogni qual volta che un utente scrive un post su Facebook o carica sul suo profilo un album fotografico, questo non sta solamente scrivendo del testo o eseguendo l'upload di un insieme di file, ma sta creando un insieme di relazioni strutturate fra nodi (di oggetti predefiniti) del grafo sociale.

A ognuna di queste entità viene associato un identificativo univoco grazie al quale può essere richiamato e interrogato.

Per potere accedere ad uno di questi nodi e analizzarne le sue proprietà è sufficiente richiamare l'indirizzo *https://graph.facebook.com/<id>*, dove <id> è l'identificativo associato all'entità.

In alternativa, ogni entità associata a un nome univoco, come ad esempio i profili degli utenti o le pagine, possono essere richiamati sostituendo questo campo al numero identificatore ID.

Ad esempio, utilizzando come identificativo il numero 15614977783739 associato alla pagina ufficiale di FacebookItalia, le sue proprietà, recuperabili all'indirizzo *https://graph.facebook.com/15614977783739* potranno essere visualizzate anche tramite l'indirizzo *https://graph.facebook.com/FacebookItalia*.

```
{
  "id": "15614977783739",
  "about": "Benvenuti sulla Pagina ufficiale Facebook Italia. Clicca su \"mi piace\" per rimanere aggiornato su nuovi prodotti, impostazione della privacy e sicurezza.",
  "can_post": false,
  "category": "Internet/software",
  "checkins": 0,
  "cover": {
    "cover_id": 692842557447789,
    "offset_x": 0,
    "offset_y": 0,
    "source": "https://scontent-b.xx.fbcdn.net/hphotos-xpfl/t31.0-8/q84/s720x720/10536720_692842557447789_6764044441954491528_o.jpg",
    "id": "692842557447789"
  },
  "description": "Benvenuti sulla Pagina ufficiale Facebook Italia. Qui troverete informazioni sulle nostre nuove funzionalit\u00e0, sulla sicurezza del vostro account e sulla privacy. Per unirvi a noi cliccate su Mi piace.\n\nPotete inoltre consultare il nostro Centro assistenza al seguente indirizzo: www.facebook.com/help.php",
  "has_added_app": false,
  "is_community_page": false,
  "is_published": true,
  "likes": 169060380,
  "link": "https://www.facebook.com/FacebookItalia",
  "mission": "Rendere il mondo pi\u00f9 aperto e connesso.",
  "name": "Facebook",
  "parking": {
    "lot": 0,
    "street": 0,
    "valet": 0
  },
  "talking_about_count": 645428,
  "username": "FacebookItalia",
  "website": "www.facebook.com/FacebookItalia",
  "were_here_count": 0
}
```

**Figura 4.2 - JSON informazioni pagina FacebookItalia**

Inoltre, come gi\u00e0 specificato, tutti gli oggetti del grafo sociale sono connessi tra di loro attraverso collegamenti esaminabili attraverso una struttura URL del tipo:

*[https://graph.facebook.com/<id>/<connection\\_type>](https://graph.facebook.com/<id>/<connection_type>)*

Considerando il collegamento relativo al richiamo del proprio utente *<https://graph.facebook.com/me>*, le connessioni accessibili dal proprio profilo includono:

- *Lista degli amici*: *[/friends?access\\_token=...](#)*
- *News feed*: *[/home?access\\_token=...](#)*
- *Timeline*: *[/feed?access\\_token=...](#)*
- *Like*: *[/likes?access\\_token=...](#)*
- *Like su oggetto di tipo Movie*: *[/movies?access\\_token=...](#)*
- *Like su oggetto di tipo Music*: *[/music?access\\_token=...](#)*
- *Like su oggetto di tipo Book*: *[/books?access\\_token=...](#)*
- *Note*: *[/notes?access\\_token=...](#)*
- *Tag nelle foto*: *[/photos?access\\_token=...](#)*
- *Album di foto*: *[/albums?access\\_token=...](#)*
- *Tag nei video*: *[/videos?access\\_token=...](#)*

- *Elenco dei propri eventi:* `/events?access_token=...`
- *Elenco dei gruppi a cui si è iscritti:* `/groups?access_token=...`
- *Posti:* `/locations?access_token=...`

A differenza delle chiamate relative alle entità del grafo, come i profili utenti o le pagine, i percorsi appena elencanti includono il parametro `access_token` in chiusura. Questo parametro viene utilizzato per garantire un accesso autenticato e autorizzato alle informazioni contenute dal grafo sociale di Facebook.

Per ottenere, infatti, queste informazioni aggiuntive è necessario ottenere l'autorizzazione da parte dell'entità a cui esse appartengono, al fine di garantire e tutelare le informazioni contenute all'interno di Facebook, impedendo di fatto l'estrapolazione di un'immagine completa dei dati del grafo social.

#### **4.4.1 Access Token**

Il meccanismo dedicato all'autenticazione utilizzato da Facebook è basato sul protocollo *OAuth 2.0*, per il quale è necessaria l'acquisizione di un access token.

Un token di accesso è una stringa che identifica un utente, un'applicazione, o una pagina e può essere utilizzato da un'app per effettuare chiamate Graph API.

Ogni token contiene, al suo interno, informazioni che ne definiscono la data in cui scadrà il suo possibile utilizzo e le indicazioni sull'app dal quale è stato generato.

Vi sono diversi tipi di access token, ognuno dedicato a supportare un diverso caso di utilizzo:

- *User Access Token:* E' il token più utilizzato, essendo necessario ogniqualvolta un'applicazione chiama un API per leggere, modificare o scrivere i dati Facebook di un utente. In generale, viene ottenuto tramite una finestra di login, nella quale viene richiesto di consentire alla nostra applicazione di ottenere uno.
- *App Access Token:* E' necessario per modificare e leggere le impostazioni dell'applicazione.
- *Page Access Token:* Questi token di accesso sono simili a quelli utente, salvo che quest'ultimi forniscono il permesso alle API di leggere, scrivere o modificare i dati appartenenti ad una pagina Facebook.
- *Client Token:* E' possibile incorporare, questo tipo di token, all'interno dei file relativi a una applicazione, mobile o desktop, per identificarla. Come

si può evincere dal suo metodo d'integrazione, non è finalizzato ad essere un codice segreto. Per ottenerlo è sufficiente accedere alla dashboard della propria app.

La prima tipologia elencata, come già anticipato, è quella maggiormente utilizzata, essendo questa impiegata per l'autorizzazione di un utente durante un'azione di login.

Quando infatti si accede ad un sito terzo utilizzando un login basato su un'autenticazione tramite Facebook, l'app finalizzata alla gestione dell'autorizzazione, richiederà il permesso di accedere ad alcune delle informazioni contenute nel nostro profilo Facebook, ottenendo e registrando così questi permessi necessari sotto forma di access token.

#### **4.4.2 Permessi**

I permessi sono suddivisi in categorie in base al modo in cui la relativa richiesta viene presentata agli utenti e in base al processo di revisione necessario per rendere pubblica l'app e le funzionalità ad essa collegate.

Alcuni gruppi di permessi sono per loro natura più sensibili di altri e proprio per questo agli utenti è concessa la possibilità di scegliere se fornirli o meno, anche se richiesti dall'app sviluppata.

Alcune autorizzazioni possono anche essere definite come opzionali, facendo sì che queste appaiono su una finestra separata durante il processo di login, evidenziandone un'importanza secondaria ai fini dell'app.

Per un sottoinsieme di permessi, come anticipato, è necessario superare un processo di controllo e autorizzazione del team di Facebook, il quale solitamente avviene entro tre giorni dalla richiesta di vaglio.

I permessi per i quali non è necessario questo processo e che quindi possono essere liberamente inclusi all'interno della propria app sono i seguenti:

- *Profilo pubblico*: questo permesso consente di accedere alle informazioni di base che compongono il profilo pubblico di un utente. Questo tipo di permesso viene incluso di default in ogni tipo di richiesta;
- *App friends*: questo permesso consente di leggere la lista degli amici che a loro volta utilizzano la nostra app;
- *Email*: fornisce l'accesso all'indirizzo email primario impostato nel profilo dell'utente.

Quelli per cui invece è necessario attendere e ottenere l'approvazione da parte di Facebook sono:

- *Proprietà estese di profilo*: permesso per la lettura di informazioni sensibili che possono essere state dichiarate o meno dall'utente;
- *Permessi estesi*: questi permessi includono, ad esempio, la possibilità di pubblicare storie sulla bacheca di un utente Facebook. La richiesta di ognuno di questi permessi sarà mostrata su una finestra separata durante il processo di login, fornendo all'utente una maggior possibilità di rifiutare questa tipologia di autorizzazione;
- *Permessi Open Graph*: grazie a questi permessi è possibile ottenere l'accesso a tutti i dati Open Graph memorizzati nel profilo utente di un utente;
- *Permessi Page*: permessi necessari per amministrare una qualsiasi pagina che un utente gestisce.

#### **4.4.3 Elaborazione dei risultati**

Quando si compie una chiamata utilizzando le *Graph API*, questa restituisce, di default, tutte le proprietà previste dall'oggetto del grafo sociale correlato alla query eseguita.

E' comunque possibile scegliere quali di questi campi si desidera ottenere da una richiesta, utilizzando il parametro *fields* all'interno della stringa di chiamata HTTP alle Graph API.

Nel caso in cui ad esempio fossimo interessati ad ottenere il solo nome e cognome di un utente, potremmo comporre la richiesta come segue:

*[https://graph.facebook.com/me?fields=first\\_name,last\\_name](https://graph.facebook.com/me?fields=first_name,last_name)*

Se invece fossimo interessati a recuperare in un'unica chiamata i dati di più oggetti del grafo, potremmo eseguire quest'operazione utilizzando il parametro *ids* seguito dai codici identificativi degli oggetti, ad esempio:

*<http://graph.facebook.com/?ids=12345678,98765432>*

Questo parametro, oltre ad accettare l'id associato ad un oggetto presente su Facebook, consente di analizzare anche URL di oggetti esterni, ma comunque parte del grafo sociale, grazie all'integrazione dell'*Open Graph Protocol*.

Ad esempio:

*<https://graph.facebook.com/?ids=http://www.imdb.com/title/tt0117500/>*

restituirà le informazioni relative al film “The Rock” (1996) descritte all’interno delle property Open Graph inserite nel codice della pagina.

## 5. Il Progetto “Volontieri”

### 5.1 Introduzione al progetto

Sempre più di frequente e da un numero in continua crescita di utenti, il web viene interrogato per la fruizione di informazioni riguardanti le più varie attività connotabili al vivere quotidiano.

Sempre di più infatti viene interrogato per la ricerca di consigli su dove pranzare, quale musica ascoltare o quale film al cinema potrebbe interessarci.

Informazioni spesso contenute o integrate in applicazioni con un’alta interazione sociale, capaci di influenzare in modo importante piccole scelte di tante persone.

Si viene così ad avere un insieme ricco d’informazioni d’interesse oggettivo, come i recapiti telefonici o gli orari di apertura di un ristorante, ad una collezione di racconti basati su esperienze soggettive come le impressioni o le critiche sui piatti serviti in quello stesso ristorante.

In questo contesto nasce il progetto *Volontieri*, sito web realizzato per la produzione della presente tesi di laurea.

L’idea di base consiste nel realizzare un’applicazione finalizzata a coinvolgere e favorire la partecipazione di utenti vicini e interessati al mondo del volontariato, a progetti di associazioni senza scopo di lucro, nazionali ed eventualmente internazionali, grazie al potenziale ottenibile dall’integrazione con le principali funzionalità social fornite dai social network maggiormente utilizzati come Facebook, Google+ e Twitter.

Durante la progettazione si è cercato di prestare particolare attenzione alla produzione di uno strumento semplice e coinvolgente per gli utenti, orientato a promuovere le loro interazioni.

Queste, infatti, ricoprono la parte fondamentale, cuore del progetto, essendo insieme all’attività sociale di condivisione e pubblicizzazione e all’impegno nella recensione e nella pubblicazione delle proprie esperienze, potranno favorire i

progetti presenti sul sito e aumentare la visibilità, la condivisione e l’attendibilità delle relative associazioni.

Per mantenere un iniziale livello di semplicità, dovuto alla versione attualmente finalizzata a scopo dimostrativo delle integrazioni sopra citate, la gestione degli utenti è stata ristretta e limitata alla sola porzione degli utenti finali, considerando come sviluppo futuro, la produzione della gestione delle utenze Associazioni e lo studio della soluzione più consona per l’inserimento, più o meno automatizzato, dei rispettivi progetti.

Di seguito verranno elencati gli strumenti utilizzati ed analizzati nello specifico le porzioni di codice governanti le logiche di funzionamento del progetto.

## **5.2 Architettura del progetto**

### **5.2.1 WordPress**

Per la produzione del progetto e per facilitare la gestione dei contenuti, si è scelto di utilizzare *WordPress*, un CMS (*Content Management System*) open source utilizzato principalmente per la generazione di siti web dedicati al blogging, scritto in PHP e utilizzante per l’archiviazione delle informazioni, una base di dati relazionale MySQL. Rilasciato per la prima volta nel 2003 è diventato oggi il servizio open source maggiormente utilizzato per la generazione di siti web finalizzati al blogging.

Giunto infatti alla versione 4.1 nel dicembre 2014, WordPress risulta essere la scelta maggiormente utilizzata fra i CMS attualmente presenti, con il 61,1% di installazioni nel totale dei siti utilizzanti questo tipo di tecnologia web, pari al 23,2% del totale dei siti web attualmente presenti a livello globale [WOR – 2014].

Le sue principali caratteristiche, capaci di motivarne questa vastissima distribuzione, sono:

- *una struttura modulare*, basata su un’architettura incrementale a plugin;
- *un sistema di templating* per la generazione delle viste dei contenuti e della gestione delle funzionalità;
- *una gestione degli utenti*, garante di un sistema di accesso sicuro e modificabile tramite livelli di accesso e permessi personalizzabili;

- *una ampia community*, costantemente impegnata nel garantirne il supporto e lo sviluppo.

I concetti principali disegnati dal CMS sono i *post* ovvero gli articoli e le *pages* cioè le pagine, caratterizzabili, i primi da un insieme di categorie, le seconde da strutture gerarchiche. Caratteristiche finalizzate a permetterne la navigazione corretta e una consultazione migliore grazie all'utilizzo di sistemi di navigazione come i *breadcrumb*.

Inoltre, per entrambi gli elementi, grazie all'utilizzo di template di vista personalizzabili, descrivibili in PHP, in WordPress sono possibili il richiamo e la presentazione di contenuti inseribili manualmente tramite backend o in modo automatico grazie a funzionalità custom.

Nel caso specifico dell'elaborato di tesi, per facilitare la gestione delle associazioni e dei relativi progetti nel backend dell'applicazione, al momento della definizione iniziale della struttura del sistema, sono state dichiarate due omonime astrazioni al concetto di base dei post, istanziando la funzionalità *Custom Post Type*. Grazie a questa utilità, è possibile definire oggetti, con comportamenti e capacità similari a livello di gestione al post, ma modellabile a piacimento in termini di nomi, categorie e campi aggiuntivi.

Inoltre, per aumentare le potenzialità di base fornite dal CMS sono stati installati due plugin aggiuntivi:

- *Advanced Custom Field*: un utile strumento sviluppato per estendere le tipologie di dato definibili tramite i custom field di WordPress e rendere disponibile un'interfaccia grafica e flessibile nel backend, per creare e gestire i campi personalizzati per ogni tipologia di post e di custom post type, di tassonomie, di categorie e di file multimediali o utenti;
- *WordPress SEO plugin*: un insieme di funzionalità aggiuntive finalizzate alla correzione e al miglioramento delle caratteristiche SEO di ogni singolo post o pagina contenuto nel sito. Consente anche la creazione automatica di *sitemap* e la possibilità di creare *breadcrumb* di navigazione.

### 5.2.2 Facebook PHP SDK

Le librerie PHP Facebook SDK forniscono agli sviluppatori, uno strumento nativo per l’accesso alle Graph API e fruire dei vantaggi offerti dall’utilizzo del Facebook Login. Queste possono essere utilizzate per la realizzazione dei propri siti web, come ausilio per la produzione di Canvas app, ovvero applicazioni contenute web inserite all’interno di Facebook o in completamento di funzionalità server-side ad applicazioni già utilizzando le librerie client-side Javascript SDK.

Per lo sviluppo di questo progetto sono state utilizzate le librerie di recente aggiornate, nella versione 4.0.

Quest’ultima versione delle SDK, annunciate e rilasciate durante l’F8 2014, è stata sottoposta ad una totale riscrittura rispetto alle precedenti librerie, richiedendo infatti, ad esempio, una versione di PHP 5.4 o superiore.

Oltre ad aver semplificato e aumentato le funzioni presente, questa versione vede una suddivisione del codice in 18 classi tipizzate, a differenza delle 3 presenti nella precedente versione, favorendo un miglior approccio di programmazione ad oggetti.

### 5.2.2 Alberatura del progetto

Il sito web presenta nell’header, un menu di primo livello, tramite il quale è possibile consultare i vari progetti presenti e le relative associazioni, oltre alle classiche voci rivolte alla pubblicazione e alla descrizione delle finalità del progetto, alle istruzioni d’uso e alla raccolta, tramite un apposito modulo di contatti, di richieste di informazioni.

Nella *homepage*, situati al di sotto dello slider contenente i *claim* di promozione dello strumento e sotto alla *call to action* indirizzata a incrementare la registrazione di associazioni al servizio, sono stati posizionati sei blocchi contenenti gli ultimi progetti pubblicati sul sito.

Nel caso in cui dalla homepage o dalla voce di menu, venga selezionato un progetto, l’utente verrà rimandato alla *pagina del progetto* contenente le informazioni dettagliate, le eventuali immagini pubblicate e una sezione contenente i racconti degli utenti e il voto dato da questi al progetto.

La funzionalità di voto è consentita soltanto agli utenti che hanno precedentemente effettuato il login al sito web, vista la presenza di azioni social necessitante del riconoscimento dell'utente.

La media di questi voti, se presenti, è riportata in alto sotto al titolo del progetto per darne maggior visibilità.

Dal progetto è possibile visualizzare il nome della relativa associazione ed approfondirne i dettagli cliccandoci sopra. In questo caso infatti, verremo reindirizzati all'interno della *pagina dell'associazione*, dove l'utente potrà visualizzare i dettagli, i contatti di riferimento, il numero di persone coinvolte in ambito social e l'elenco dei progetti attribuiti all'associazione.

In questo modo l'utente potrà navigare e approfondire in modo intuitivo tutti i progetti presenti sul sito.

Il *footer* dell'applicazione è stato suddiviso in tre aree: la prima dedicata alla pubblicizzazione del progetto più rilevante, ad aver cioè ricevuto il maggior numero di voti con punteggio positivo, la seconda alla pubblicazione e all'evidenziazione delle ultime recensioni inserite dagli utenti, ordinate per data, mentre la terza pensata per aggregare le notizie provenienti dai social network riguardanti il progetto Volontieri e il suo coinvolgimento sui social.

**Volontieri** Accedi con Facebook

Associazioni | Progetti | Scoprici | Sostienici | Contattaci

Q | ✉ | f | t

*Sei un associazione? Inviaci subito i tuoi progetti!*  
Un mondo di volontari è in attesa di conoscerti

**ISCRIVITI**

**Progetto C**  
Associazione Abbraccio  
Forlì (FC) - Emilia Romagna  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec. Lorem [...]

**Progetto B**  
Associazione Abbraccio  
Cesena (FC) - Emilia Romagna  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec. Lorem [...]

**Progetto A**  
Associazione Raggio di Sole  
Milano (MI) - Lombardia  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec. Lorem [...]

**Progetto Z**  
Associazione Raggio di Sole  
Roma - Lazio  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec. [...]

**Progetto Y**  
Associazione Viva la Vita  
Bologna (BO) - Emilia Romagna  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec. Lorem [...]

**Progetto X**  
Associazione Viva la Vita  
Forlì (FC) - Emilia Romagna  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec. Lorem [...]

**I più amati**  
Nome Progetto  
Nome Associazione  
Villaggio Mafalda  
"Lorem ipsum lorem dolor sit amet..."

**Gli ultimi commenti**  
25 gennaio 2015  
Progetto C  
"E' stata un'esperienza fantastica che mi accompagnerà per sempre"  
21 gennaio 2015  
Progetto Z  
"Test votazione di questa esperienza"

**Volontieri**  
One person likes Volontieri.

Find Us on Social

Copyright 2014 - Volontieri

Figura 5.1 - Volontieri - Home Page

## 5.3 Integrazioni social

In questo capitolo verranno descritti dettagliatamente i passaggi eseguiti per l'integrazione delle funzionalità social all'interno del progetto.

Nello specifico saranno esaminate la parte riguardante la creazione dell'App Facebook tramite la dashboard dedicata del social network e quella invece relativa alla parte di programmazione PHP per l'integrazione nel sito web.

### 5.3.1 Creazione dell'App Volontieri

Come visto precedentemente, nel capitolo riguardante la pubblicazione di un applicazione Facebook, per poter integrare le funzionalità social all'interno del proprio sito web o della propria applicazione, è necessario per prima cosa creare una nuova app Facebook, tramite la dashboard dedicata agli sviluppatori.

In questo caso, essendo il progetto finalizzato all'utilizzo sul web, durante la fase iniziale di scelta della destinazione dell'app, si è scelto di creane una nella quale l'ambiente di destinazione era proprio il WWW.

Sono poi stati definiti i parametri di configurazione, inserendo come namespace *volontieri* e come domain quello creato apposta per il testing del progetto.

Successivamente, tramite la sezione *Open Graph*, è stata definita l'azione e l'oggetto protagonisti della *Custom Stories* pensata per questa tesi, ovvero l'oggetto “Progetto” e l'azione “ha partecipato” che si è scelto di eseguire al momento di una determinata azione all'interno del progetto, quella di recensione e di racconto dei progetti delle associazioni.

### 4.3.2 Login

Una volta configurata l'applicazione tramite la dashboard di Facebook e inserite le librerie all'interno del progetto è possibile iniziare la composizione del codice PHP per la funzionalità di login.

Come prima cosa, nel codice richiamato per questa specifica funzionalità, viene inizializzato l'utilizzo della sessione all'interno dell'applicazione tramite il codice *session\_start()* e definiti, tramite l'uso di variabili dedicate, i parametri generati durante la creazione dell'applicazione e i permessi considerati, in fase di progettazione, necessari all'interno del sito e dell'applicazione.



Istanziato l'utilizzo delle classi necessarie, viene successivamente inizializzato l'impiego delle SDK, utilizzando i parametri dell'applicazione generati in fase di creazione dell'App Facebook.

Nello specifico vengono inseriti l'*app\_id* e l'*app\_secret*, parametri identificativi e univoci per dichiarare l'app utilizzata e il *redirect\_uri*, indirizzo della pagina web a cui deve essere reindirizzato l'utente una volta effettuato il login su Facebook.

```
// Caricamento librerie Facebook PHP SDK
require_once( get_template_directory() . '/facebooksdk/autoload.php' );

use Facebook\FacebookRequest;
use Facebook\GraphUser;
use Facebook\FacebookSession;
use Facebook\FacebookRedirectLoginHelper;

// Inizializzazione SDK
$facebook = FacebookSession::setDefaultApplication($app_id, $app_secret);

// Creazione dell'helper di login
$helper = new FacebookRedirectLoginHelper($redirect_uri);
```

**Figura 5.3 - Facebook Login - Inizializzazione classi**

Inizializzata l'applicazione, il passo successivo consiste nel creare la sessione, tramite la quale, sarà possibile mantenere attivo lo stato dell'utente e registrare le eventuali azioni che questo compirà all'interno del sito.

A questo scopo viene verificata innanzitutto la presenza di un *token* di sessione Facebook. Nel caso viene creata una variabile *\$session* successivamente validata per verifica.

In caso contrario viene verificata l'eventualità in cui la variabile *\$session* sia vuota. Se ciò è vero, si procede con il verificare che non ci si trovi in una sessione ancora non impostata poiché l'utente corrente proviene da un redirect dopo il login.

Se questa condizione risulti vera, viene salvato all'interno della variabile *\$session* il valore di questa sessione.

```
// Verifica l'esistenza di sessioni attive
if (isset($_SESSION) && isset($_SESSION['fb_token'])) {
    // Create una nuova sessione dall'access_token salvato
    $session = new FacebookSession($_SESSION['fb_token']);
    // Valida l'access_token per verificare se è ancora valido
    try {
        if (!$session->validate()) {
            $session = null;
        }
    } catch (Exception $e) {
        // Catch delle eccezioni
        $session = null;
    }
} elseif (empty($session)) {
    // in questo caso la sessione è vuota per cui ne creo una nuova
    try {
        // il visitatore viene rediretto dal login e viene creata una sessione
        $session = $helper->getSessionFromRedirect();
    } catch (FacebookRequestException $e) {
        // Facebook ha riportato un errore
        echo 'Facebook (session) request error: ' . $e->getMessage();
    } catch (Exception $e) {
        // Altri errori
        echo 'Other (session) request error: ' . $e->getMessage();
    }
}
}
```

**Figura 5.4 - Facebook Login - Verifica sessioni**

Da questo momento, è possibile utilizzare la sessione presente per governare le successive azioni da compiere, contenendo questa tutte le informazioni dell'utente che dovrà compiere il login.

In ordine, fra queste azioni, vi è il settaggio del percorso di logout per l'utente attivo, ovvero la pagina a cui atterrerà l'utente dopo essersi disconnessi da Facebook.

Per farlo viene utilizzata la variabile *\$helper* impostata in precedenza, utilizzando il metodo dedicato *getLogoutUrl*.

```
// Se esiste una sessione attiva
if (isset($session)) {
    // Salva la sessione
    $_SESSION['fb_token'] = $session->getToken();
    // Crea una nuova sessione usando il token salvato o quello creato durante il login
    $session = new FacebookSession($session->getToken());
    // Crea il logout url e dirige ad una pagina di eliminazione della sessione
    $logoutURL = $helper->getLogoutUrl($session, esc_url(home_url('/')) . 'logout');
```

**Figura 5.5 - Facebook Login - Logout url**

Tramite il metodo *FacebookRequest* viene effettuata una chiamata alle Graph API, in questo caso una *GET*, mirata a recuperare e leggere i dati relativi all’utente attualmente in sessione, identificabile dalla preposizione */me*.

Di questo utente vengono poi salvati, secondo lo schema *className* definito all’interno dell’oggetto *GraphUser*, all’interno dell’oggetto *\$user\_profile*.

Ora per garantire la permanenza di questi dati all’interno del sito e per poterli utilizzare anche nei successivi passi dell’applicativo, queste informazioni sono associate a variabili di sessione, utilizzando la funzione *getProperty* associata all’oggetto *GraphUser*.

```
// Viene effettuata la chiamata all'oggetto GraphUser relativo all'utente di sessione
$user_request = (new FacebookRequest($session, 'GET', '/me'));
$user_profile = $user_request->execute()->getGraphObject(GraphUser::className());

// Creo delle variabili di sessione in cui salvo le informazioni lette
// dall'oggetto Facebook
$_SESSION['FBID'] = $user_profile->getProperty('id');
$_SESSION['FBname'] = $user_profile->getProperty('first_name');
$_SESSION['FBsurname'] = $user_profile->getProperty('last_name');
$_SESSION['FBemail'] = $user_profile->getProperty('email');
$_SESSION['FBpicture'] = 'https://graph.facebook.com/'
    . $_SESSION['FBID'] . '/picture';

// Richiamo il documento PHP per la creazione del corrispettivo
// utente Wordpress
require_once 'wp_connect.php';
```

**Figura 5.6 - Facebook Login - Richiesta utente**

Di questi, vengono letti e memorizzati l’*id* dell’utente, il *nome*, il *cognome*, l’*email* e l’*immagine del profilo*, informazioni di base utilizzate per la creazione dell’utente in WordPress e per il saluto di benvenuto dell’utente al momento del login.

La lettura di questi sarà possibile solo nel caso in cui, nel momento del login, l’utente decida di accettare la richiesta dei permessi impostati.

Creazione dell’utente che avviene in seguito grazie all’inclusione, nel caso questi dati siano stati recuperati correttamente, del file *wp\_connect.php*.

In questa fase, a seguito di una verifica rispetto alla presenza di un utente che abbia effettivamente compiuto un login con Facebook, viene verificata la presenza di questo tramite il campo email, fra gli utenti attivi già presenti sul sito.

Nel caso in cui non esista un utente già attivo sul sito con lo stesso indirizzo di posta elettronica, viene generata una password randomica di 12 caratteri e

successivamente associati ai campi WordPress dedicati all’utente, le informazioni lette dalla request svolta verso le Graph API in precedenza.

A questo punto, grazie alla funzione `wp_insert_user()`, viene inserito il nuovo utente fra quelli attivi al sito.

<b>Nome</b>	
Nome utente	<input type="text" value="1547974935452416"/> <small>Il nome utente non può essere modificato.</small>
Ruolo	<input type="text" value="Sottoscrittore"/>
Nome	<input type="text" value="Open"/>
Cognome	<input type="text" value="User"/>
Nickname (obbligatorio)	<input type="text" value="1547974935452416"/>
Nome pubblico da visualizzare	<input type="text" value="Open User"/>
<b>Informazioni contatti</b>	
E-mail (obbligatorio)	<input type="text" value="open_wevmxvc_user@tfnw.net"/>

**Figura 5.7 - Wordpress - Utente backend**

Viceversa, nel caso in cui l’email risulti essere già in uso da utente attivo, si procede con il richiamo di questo utente grazie alla funzione `$user = get_user_by("login", $_SESSION['FBID'])`, inviando come parametro di login il codice *ID* letto dalla sessione di Facebook.

```
if (isset($_SESSION['FBID'])) {
    $user_id = username_exists($_SESSION['FBID']);
    if (!$user_id and email_exists($_SESSION['FBID']) == false) {

        $random_password = wp_generate_password($length = 12, $include_standard_special_chars = false);

        $userdata = array(
            'user_login' => $_SESSION['FBID'],
            'user_pass' => $random_password,
            'user_email' => $_SESSION['FBemail'],
            'first_name' => $_SESSION['FBname'],
            'last_name' => $_SESSION['FBsurname'],
            'description' => $_SESSION['FBpicture'] . "?width=100"
        );

        $user_id = wp_insert_user($userdata);
    } else {
        $user = get_user_by("login", $_SESSION['FBID']);
        if ($user != "FALSE") {
            wp_set_auth_cookie($user->ID);
        }
    }
}
```

**Figura 5.8 - Creazione utente Wordpress**

Successivamente si autorizza l'utente come attivo per la sessione in corso, grazie all'utilizzo di un cookie impostabile con la funzione di WordPress `wp_set_auth_cookie($user->ID)`. Solitamente si utilizza un'altra funzione dedicata al login utente su WordPress, chiamata `wp_signon()`, ma in questo caso non è stato possibile dato che come parametro di ingresso è richiesto il campo password, valore da noi generato casualmente in fase di creazione utente e non noto durante il login.

Conclusa questa parte dedicata alla gestione dell'utente, l'ultima parte del codice è dedicata alla produzione degli strumenti di login e logout.

```
// Stampo a video il messaggio di benvenuto utente e la sua foto
// ed un link per il logout dall'applicazione
echo '<h2>Ciao, ' . $_SESSION['FBname'] . ' ' . $_SESSION['FBsurname'] .
'</h2>'
. '<div class="logout"><a id="logout" href="' . $logoutURL . '">Logout</a></div>';
} else {
// Non ci sono sessioni attive
// Crea l'url per il login e stampa a video il pulsante relativo
$loginUrl = $helper->getLoginUrl($permissions);
?>
<div id="facebook">
<p><a href="<?php echo $loginUrl; ?>">Accedi con Facebook</a></p>
</div>
<?php } ?>
```

**Figura 5.9 - Facebook Login - Logica pulsante**

Continuando dal punto precedente, e quindi nel caso in cui l'utente abbia effettuato un login e i suoi dati siano stati letti con successo, questi vengono stampati nell'header del sito con un messaggio di benvenuto.

Viene anche inserito un collegamento di logout, contenente l'url per il logout da Facebook, e la pagina su cui atterrare una volta compiuta quest'azione.

Nel caso in cui invece ci trovassimo senza utenti attivi sul sito, ovvero visitatori o utenti che non hanno ancora effettuato il login, viene stampato il pulsante di accesso tramite Facebook, collegato all'url prodotto grazie alla funzione `GetLoginUrl()`, alla quale vengono inviati i permessi da richiedere all'utente al momento del login.

Riassumendo, l'utente, cliccando sul pulsante di login verrà reindirizzato ad una pagina interna a Facebook nella quale dovrà inserire le proprie credenziali di accesso.

Successivamente, nel caso in cui non lo abbia ancora fatto, grazie ad una finestra contenente un'immagine riconoscitiva ed il riassunto dei permessi richiesti, dovrà accettare l'applicazione Facebook collegata al login.

Ulteriore passaggio, opzionale e quindi non discriminante ai fini del processo dell'applicazione, dovrà concedere o meno il permesso all'applicazione a pubblicare contenuti per lui sulla sua bacheca.

Terminati questi passaggi, in egual modo se li aveva precedentemente già compiuti, l'utente viene ridirezionato al sito web Volontieri, dove i suoi dati verranno letti e processati e la sua sessione verrà inizializzata autenticandolo.



**Figura 5.10 - Volontieri - Flusso Login**

Cliccando invece il collegamento di logout, la sessione dell'utente verrà eliminata e questo verrà ridirezionato alla homepage dove visualizzerà nuovamente il pulsante di login.

### 5.3.3 Metadati Open Graph

Una dei passaggi principali di questo progetto, è stato quello riguardante l'inserimento dei metadati Open Graph all'interno delle pagine del sito.

Per farlo si sarebbe potuto utilizzare uno dei diversi plugin messi a disposizione di WordPress per lo scopo, ma per poter avere un maggior controllo ed evitare di appesantire il progetto con funzionalità non desiderate, si è scelto di inserire manualmente il codice.

Nello specifico si è fatta distinzione fra la homepage, le pagine dedicate ai progetti, quelle relative alle associazioni, ed in generale le rimanenti.

Questa differenziazione è soprattutto derivante dalla eterogenea natura dei contenuti e del tipo di informazioni descritte da queste pagine, e dal diverso scopo ultimo nell’utilizzo dei tag Open Graph.

Innanzitutto, per ogni pagina, viene definito l’id dell’app Facebook richiamata nel corso dell’utilizzo delle funzionalità del sito.

Si sono poi descritte le proprietà esclusive descriventi la homepage, come l’immagine di riferimento, il titolo, contenente il nome del sito e il relativo motto, e la descrizione di questo, con una breve introduzione allo scopo immaginato.

```
<?php
global $post;
echo '<meta property="fb:app_id" content="1439954866269898" />';
if (is_front_page()) {
    echo '<meta property="og:image" content="http://volontieri.altervista.org/wp-content/uploads/2014/12/volontieri.jpg" />';
    echo '<meta property="og:title" content="Volontieri - Le tue esperienze diventano guida" />';
    echo '<meta property="og:description" content="Volontieri è uno spazio dedicato alle associazioni,
    dove inserire i propri progetti, e dove poterli promuovere grazie ai racconti dei propri volontari." />';
} else {
    echo '<meta property="og:title" content="'. the_title() .'" />';
    $post_object = get_post($post->ID);
    echo '<meta property="og:description" content="'. $post_object->post_content .'" />';
    if (get_the_post_thumbnail($post->ID)){
        echo '<meta property="og:image" content="'. get_the_post_thumbnail($post->ID) .'" />';
    } else {
        echo '<meta property="og:image" content="http://volontieri.altervista.org/wp-content/uploads/2014/12/volontieri.jpg" />';
    }
}
// Se è la pagina di un progetto
if (is_singular('progetto')) {
    echo '<meta property="og:type" content="volontieri:progetto" />';
} // Se è la pagina di una associazione
} else if (is_singular('associazione')){
    echo '<meta property="og:type" content="volontieri:associazione" />';
} else echo '<meta property="og:type" content="website" />';
echo '<meta property="og:url" content="'. the_permalink() .'" />';
```

**Figura 5.11 - Open Graph - Integrazione metadati**

Per le altre pagine invece si è attribuito alla *property* “*og:title*” il titolo del post su WordPress, alla “*og:description*” il contenuto del post e come “*og:image*”, se presente l’immagine in evidenza, detta thumbnail, dell’articolo, altrimenti un’immagine di default.

L’“*og:url*” è stata l’unica proprietà descritta per tutte le pagine in egual modo, ovvero grazie all’utilizzo della funzione *permalink()* di WordPress, restituente l’url del post o della pagina che si sta visualizzando.

Infine, tramite un sistema di verifica basato sul riconoscimento della tipologia di contenuto visualizzato, si è definito per il tipo di pagina contenente un progetto, il tag *og:type* di tipo “*volontieri:progetto*”, per le pagine di tipo associazione il tag “*volontieri:associazione*” e per le restanti, il generico “*website*”.

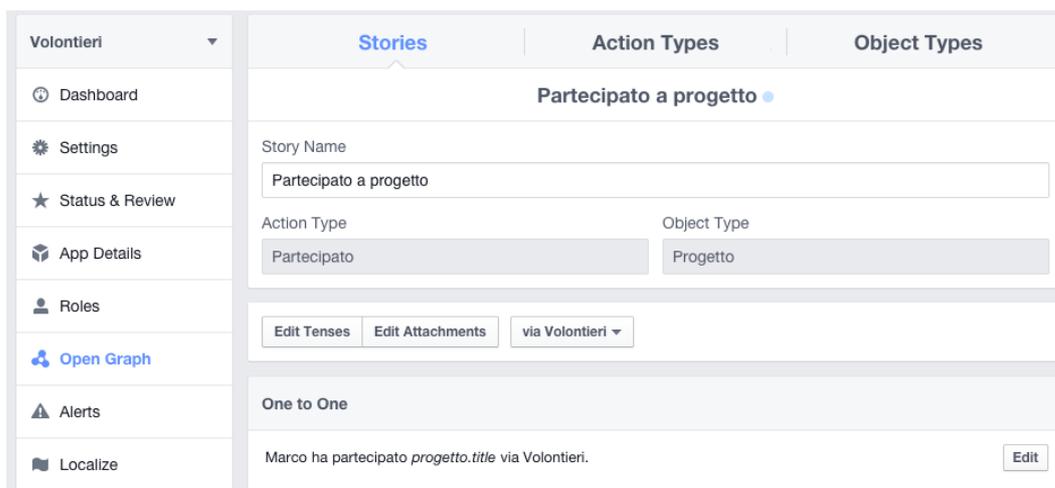
Queste due tipologie di tipo, sono valorizzazioni personalizzate create grazie alla definizione degli oggetti Open Graph tramite la Facebook dashboard, e sono necessarie per il funzionamento delle azioni scaturenti dalle Custom stories.

### 5.3.4 Custom story e Facebook feed

L’attività principale eseguibile dagli utenti connessi al sito Volontieri, è quella di recensire i progetti di volontariato delle associazioni presenti, raccontando la propria storia e lasciando un proprio voto relativo all’esperienza vissuta.

Per aumentare la visibilità di questa azione, è stato pensato di collegare all’azione di invio della recensione, l’esecuzione di una *Custom Story* Facebook.

Per raggiungere questo obiettivo, si sono innanzitutto definiti l’oggetto e l’azione relativi alla storia, utilizzando la dashboard Facebook dedicata agli sviluppatori.

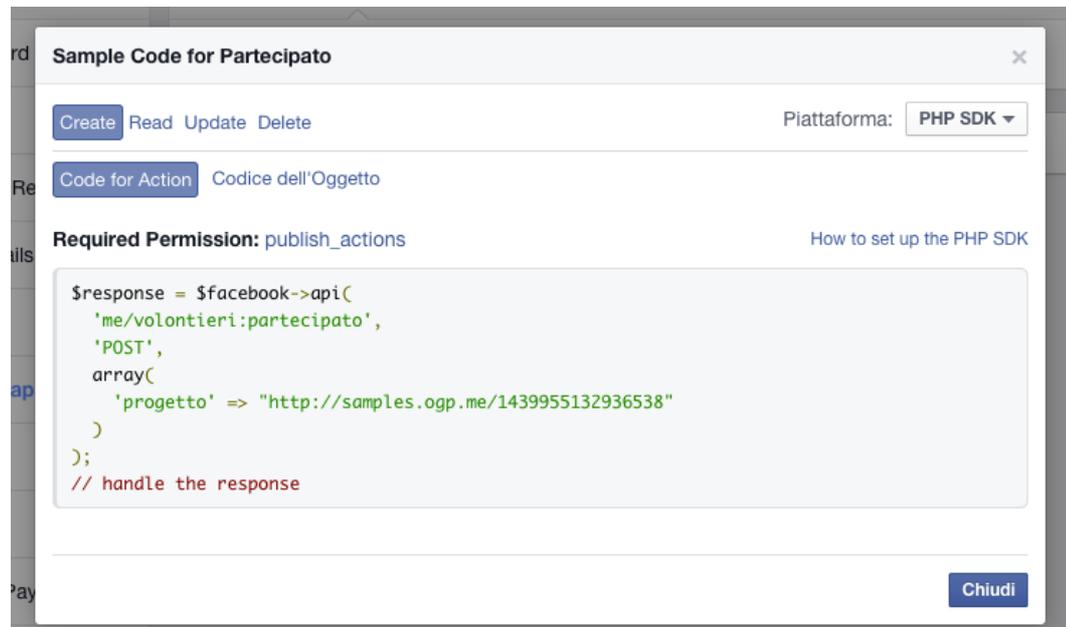


**Figura 5.12 - Facebook dashboard - Definizione Custom Story**

Nello specifico, l’oggetto è stato definito con il nome di *Progetto* mentre l’azione è stata definita come *Partecipato*, generando quindi una storia chiamata *Partecipato a progetto*.

Grazie all’interfaccia di creazione, sono stati poi definite le declinazioni nelle varie forme, singolari e plurali, nei diversi tempi grammaticali, garantendo così che nei news feed delle persone collegate agli utenti che compiranno queste azioni, compariranno delle storie nella forma più consona e corretta.

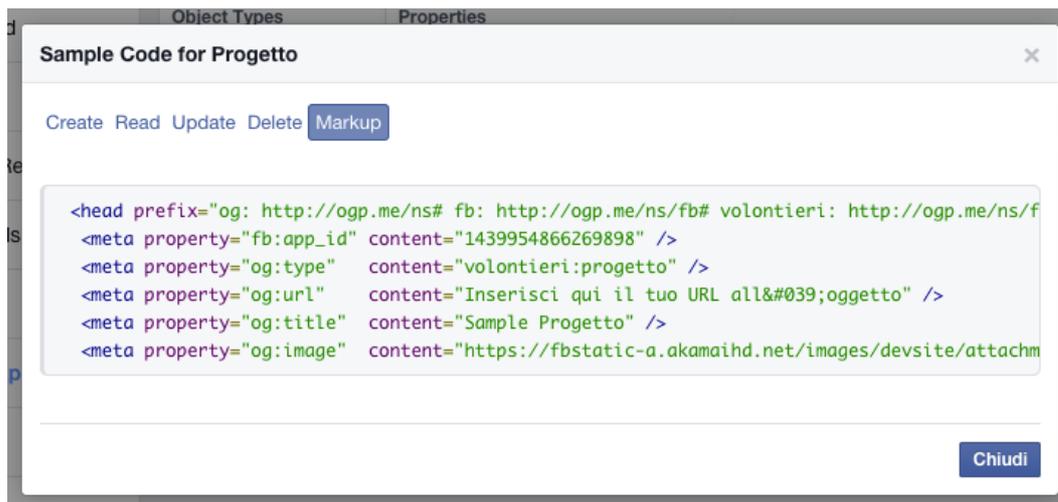
Salvata la custom story, Facebook fornisce, a seconda del linguaggio di programmazione selezionato per l'integrazione, il codice base di esempio da inserire nei propri progetti, per le diverse tipologie di azioni da compiere, come lettura, scrittura, aggiornamento e cancellazione della storia.



**Figura 5.13 - Facebook dashboard - Codice esempio oggetto Custom Story**

Inoltre viene fornito un markup di partenza da inserire all'interno dell'*head* delle proprie pagine descrittive oggetti di tipo custom, nel nostro caso quelle dedicate ai progetti.

Nel dettaglio vanno iniettati dei *prefix* al tag `<head>` nel quale viene dichiarato l'utilizzo del protocollo Open Graph e il namespace dell'applicazione utilizzata.



*Figura 5.14 - Facebook dashboard - Markup Open Graph oggetto Custom Story*

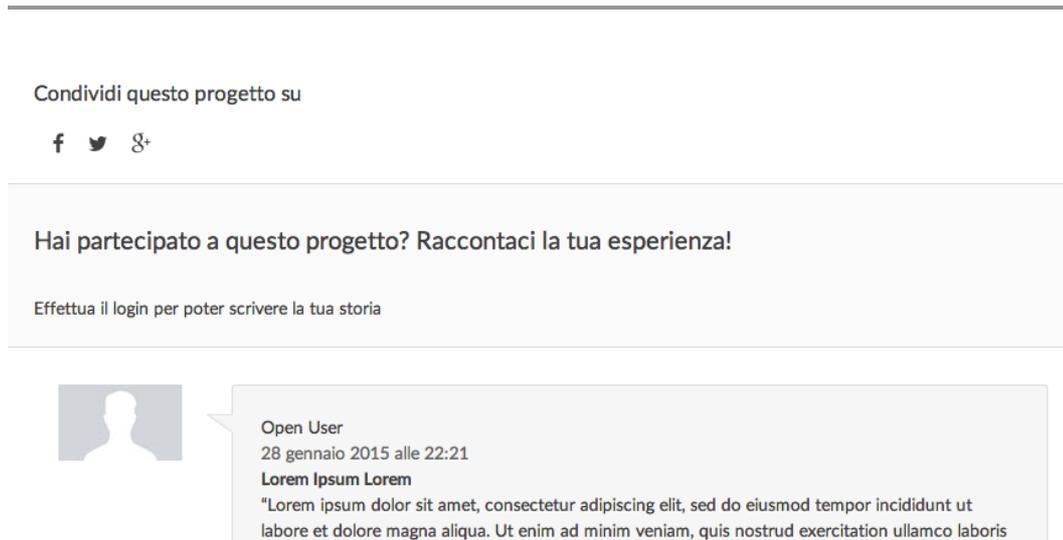
Poi si definiscono l'app-id, il type dell'oggetto descritto nelle custom story e le property più comuni come url, title e image.

Questi metadati sono stati inseriti in un file dedicato come descritto in precedenza, mentre ora verrà descritto come è stato utilizzato il codice necessario per il compimento della storia.

Per recensire i progetti e poter moderare facilmente queste recensioni, è stato estesa la funzionalità di utilizzo dei commenti messa a disposizione da WordPress per qualsiasi post.

In questo modo un eventuale amministratore del sito, potrà essere informato tempestivamente di un nuovo messaggio inviato da un'utente, consentendogli di decidere se mantenere il commento o eliminarlo.

Inoltre, utilizzando questo componente, si è potuto limitare la possibilità di recensire i progetti ai soli utenti registrati e loggati al sito, grazie alle logiche di base configurabili in modo globale o per singolo elemento, tramite le impostazioni di WordPress.



**Figura 5.15 - Volontieri - Recensione, utente non riconosciuto**

Per poter inserire un campo aggiuntivo al template ed al sistema di base della funzionalità di WordPress, si è estesa questa con degli hook, chiamate a funzioni alla base del core del CMS, principalmente finalizzati a descrivere un nuovo campo per la funzionalità di rating e per iniettare la custom story Facebook.

Per descrivere queste nuove funzionalità, è stato inserito ed esteso un plugin chiamato Extend Comment, all'interno tramite del quale, attraverso gli hook `comment_form_after_fields` e `comment_form_logged_in_after`, è stato aggiunto un ai campi previsti dalla funzionalità di commento, un campo obbligatorio di tipo radio button, trasformato tramite l'utilizzo di fogli di stile con l'uso di immagini a forma di cuore, adibito ad esprimere un voto da 1 a 5.

Hai partecipato a questo progetto? Raccontaci la tua esperienza!

Autenticato come TestNome TestCognome. Logout?

Titolo della tua esperienza:\*

Il tuo voto a questo progetto:\*

La tua esperienza:

Invia la tua storia

**Figura 5.16 - Volontieri - Recensione, utente riconosciuto**

Grazie ad un ulteriore hook, ovvero *comment\_post*, tramite il quale è possibile richiamare l'esecuzione di una funzione nel momento l'invio del form è stato eseguito e non si è ancora stati reindirizzati ad una nuova pagina, si è proceduto con il salvataggio del rating assegnato al progetto.

Per farlo, viene letto il precedente totale dei voti assegnati al progetto, se presente, memorizzato all'interno di un custom field chiamato *total\_votes*, al quale viene sommato il valore attribuito dalla recensione. Poi viene letto un ulteriore custom field chiamato *counter\_votes*, contenente un intero relativo al numero totali di voti, incrementando questo di un'unità. Il rapporto fra questi due valori viene quindi arrotondato per eccesso ad un solo valore, per poi essere memorizzato all'interno di un ulteriore campo chiamato *average\_rating*.



**average\_rating**  
Media dei voti

4

**counter\_votes**  
Contatore dei voti

3

**total\_votes**  
Somma totale dei voti

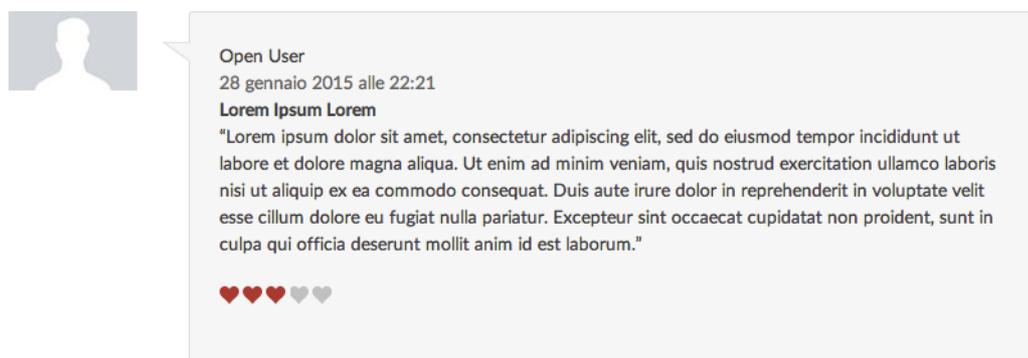
11

**Figura 5.17 - WordPress - Custom field rating**

Una volta ricaricata la pagina, il valore della media dei voti e il contatore di questi, associati al progetto, vengono letti e stampati al di sotto del titolo e delle informazioni di testata, per fornire la loro immediata visibilità all’utente, accrescendo o diminuendo l’attendibilità e il valore del progetto osservato.

La recensione viene invece salvata in attesa di moderazione da parte dell’amministratore, il quale verrà avvertito, tramite una notifica email, della presenza di un nuovo messaggio.

Una volta approvata, questa verrà pubblicata automaticamente come primo messaggio della pagina, al di sotto del modulo di invio, rispettando quindi un ordine decrescente cronologico.



**Figura 5.18 - Volontieri - Recensione utente**

Durante queste operazioni legate puramente all’amministrazione e alla pubblicazione delle recensioni e dei voti assegnati dagli utenti ai progetti presenti,

viene eseguito l’insieme di azioni riguardanti la Custom Story “*Ha partecipato al Progetto*” precedentemente creata tramite la dashboard di Facebook.

```
if (isset($_POST['submit'])) {  
    if ($session) {  
        try {  
            //Scrittura Custom Story  
            $response = (new FacebookRequest(  
                $session, 'POST', '/me/volontieri:partecipato', array(  
                    'progetto' => get_permalink()  
                )  
            ))->execute()->getGraphObject();  
        }  
    }  
}
```

**Figura 5.19 - Wordpress - Creazione Custom Story**

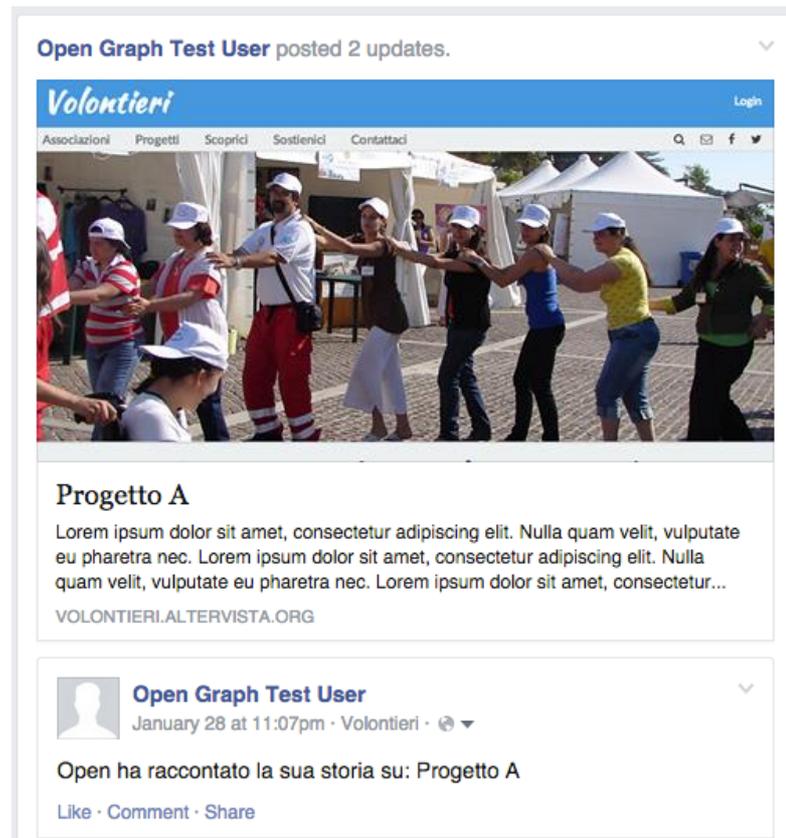
Dopo la verifica della presenza di una sessione attiva associata ad un utente, tramite la classe *FacebookRequest*, istanziabile grazie alle librerie Facebook PHP SDK, viene eseguito il codice, inizialmente descritto, recepito da Facebook per la scrittura di una Custom Story, indicando come URL, relativo all’oggetto *Progetto* partecipe della storia, il *permalink* WordPress del progetto che si è recensito e che si sta visualizzando.

In questo modo viene registrata la Custom Story, la quale andrà a leggere tutti i parametri Open Graph presenti nella pagina individuata all’indirizzo indicato, generando nel feed delle attività dell’utente un messaggio che ne conferma l’avvenuta esecuzione.



**Figura 5.20 - Facebook - Custom Story all’interno delle attività recenti Utente**

Inoltre, gli amici dell’utente protagonista, se coinvolti con le sue attività, ovvero se attenti alle sue pubblicazioni o ad argomenti simili presenti su Facebook, suggerenti quindi all’algoritmo alla base della generazione del proprio *News Feed* di inserire sue notizie all’interno di questo, potranno visualizzare un messaggio che descrive il compimento della Custom Story.



**Figura 5.21 - Facebook - Custom Story (News Feed)**

A questa azione, una volta compiuta, viene assegnato un ID che ne permette in qualsiasi momento l'identificazione e l'approfondimento della sua struttura, permettendo, allo sviluppatore dell'app, di poter risalire agli oggetti e all'utente che la compongono.

Grazie alla chiamata Graph API al nodo `/me/feed` di tipo POST, si è scelto di pubblicare sulla pagina dell'utente, un link al progetto recensito da questo, per poter aumentare la portata e la visibilità dell'azione compiuta.

```
//Scrittura messaggio sulla bacheca utente
$responseFeed = (new FacebookRequest(
    $session, 'POST', '/me/feed', array(
        'link' => get_permalink(),
        'message' => $_SESSION['FBname'] .
            ' ha raccontato la sua storia su: ' .
            get_the_title($post->ID)
    ))->execute()->getGraphObject();
```

**Figura 5.22 - WordPress - Pubblicazione post sulla bacheca utente**

Il link al progetto, durante la condivisione sulla bacheca, verrà analizzato e le informazioni Open Graph contenute nella descrizione della pagina saranno utilizzate per la pubblicazione.

Ad accompagnare queste informazioni, si è scelto di accompagnare un messaggio contenente il nome dell’utente che ha compiuto l’azione sul sito Volontieri, l’azione, ovvero “ha raccontato la sua storia su:”, e il titolo del progetto recensito. Nel post pubblicato, sotto al nome dell’utente del profilo, verrà automaticamente pubblicato anche il nome del sito dal quale ha avuto origine la pubblicazione, in questo caso Volontieri.



**Figura 5.23 - Facebook - Custom Story (Profilo utente)**

Infine, per migliorare ed incrementare l’esperienza sociale, all’interno delle pagine dei progetti, si è scelto di mostrare gli amici dell’utente connesso che hanno a loro volta recensito la pagina del progetto in esame.

Nel caso infatti che altri utenti amici del profilo connesso al sito abbiano a loro volta compiuto l’azione di recensione del progetto, le immagini dei rispettivi profili verranno pubblicati sotto alla descrizione del progetto.

Nello specifico, per realizzare questa funzionalità, è stata utilizzata la chiamata `/me/friends?limit=50&fields=name,volontieri:partecipato` di tipo GET.

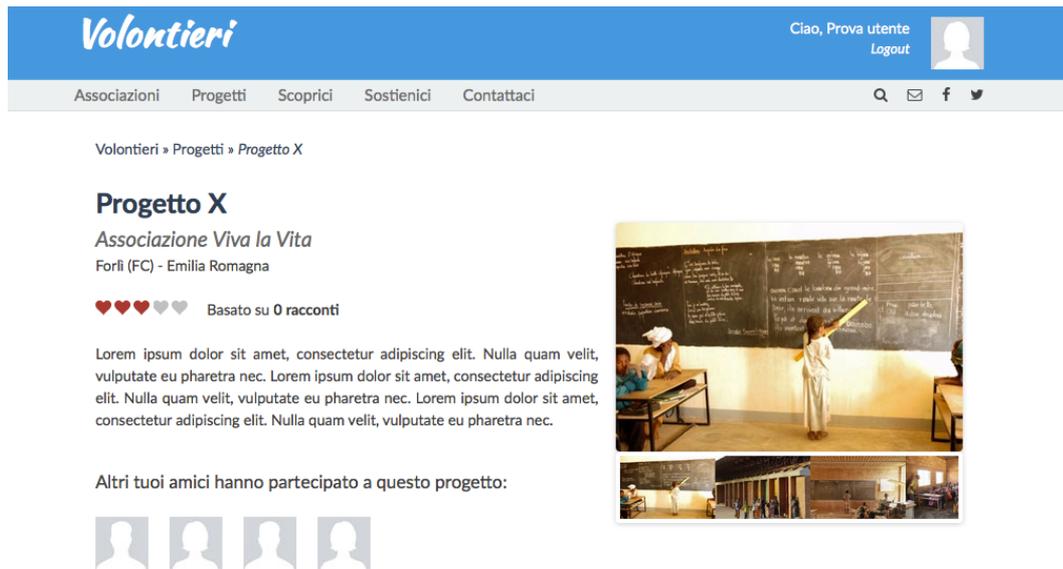


Figura 5.24 - Volontieri - Plugin LikeBox

### 5.3.5 Plugin sociali

Per aumentare il coinvolgimento sociale, oltre alle custom stories integrate e approfondite in precedenza, sono stati integrati altri plugin sociali all’interno delle pagine del progetto.

Nel footer, oltre ai collegamenti alle rispettive pagine sui principali social network, è stato inserito un *LikeBox* collegato alla pagina Facebook del progetto Volontieri, grazie al quale sarà possibile compiere con un solo click l’azione Like sulla pagina Facebook del sito e iniziare così a riceverne gli aggiornamenti.



Figura 5.25 - Volontieri - Plugin LikeBox

Inoltre, grazie alla funzionalità *FacePile*, nel momento in cui un utente avrà effettuato il login, potrà visualizzare le foto dei profili degli amici che hanno compiuto l'azione Like sulla suddetta pagina.

Per le pagine relative alle Associazioni, sono invece stati inseriti i *FollowButton* di Facebook, Twitter e Google+ essendo questi di fondamentale importanza per mantenere aggiornati gli utenti, anche su Facebook, delle proprie attività ed eventi.

Nello specifico si è scelto di introdurre la versione dei pulsanti con annesso conteggio del numero totale dei followers, altra indicazione tesa a rendere immediato, l'affidabilità delle associazioni che con il tempo sono riuscite ad accerchiarsi di diversi utenti attivi e interessati alle proprie attività.

## Associazione Viva la Vita

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique, tortor mauris molestie elit, et lacinia ipsum quam nec dui. Quisque nec mauris sit amet elit iaculis pretium sit amet quis magna. Aenean velit odio, elementum in tempus ut, vehicula eu diam. Pellentesque rhoncus aliquam mattis. Ut vulputate eros sed felis sodales nec vulputate justo hendrerit. Vivamus varius pretium ligula, a aliquam odio euismod sit amet. Quisque laoreet sem sit amet orci ullamcorper at ultricies metus viverra. Pellentesque arcu mauris, malesuada quis ornare accumsan, blandit sed diam.



### Contatta Associazione Viva la Vita

✉ info@example.com  
☎ 000 0000000  
🏠 http://www.example.com

### Segui i progetti di questa associazione su

31m	85	1.004
Follow	Segui	Tweet

**Figura 5.26 - Volontieri – Follow Button**

Non esistendo nella realtà le associazioni inserite nel progetto Volontieri, i pulsanti sono stati creati utilizzando le funzionalità di generazione automatica fornite dai tre social network, inserendo collegamenti a pagine esterne.

Infine, nelle pagine dei singoli progetti sono stati inseriti i collegamenti per la condivisione sui social network dei relativi contenuti.

Ad ogni pulsante è stata associata la rispettiva sintassi utilizzabile per lo scopo.

Nello specifico sono stati inseriti i seguenti pulsanti di condivisione:

- *Facebook*: <http://www.facebook.com/sharer/sharer.php?u=>
- *Twitter*: <https://twitter.com/intent/tweet?url=>
- *Google+*: <https://plus.google.com/share?url=>

Avendo inserito in ogni pagina i tag Open Graph, in automatico le pagine condivise saranno estremamente dettagliate, essendo arricchite con diverse informazioni associate a queste come, titolo, descrizione, immagine e tipologia.

### 5.3.6 Inserimento Schema.org

Vista la missione alla base del progetto, ovvero cercare di pubblicizzare il più possibile le associazioni ed i relativi progetti di volontariato contenuti e raggiungere un numero sempre maggiore di persone interessate a questi, si è scelto di introdurre, oltre e alle integrazioni sociali ed ai tag Open Graph di Facebook, i metadati dichiarati all'interno di *Schema.org*.

Essendo difatti presente un sistema di “votazione” collegata all'attività narrativa disponibile all'utente, si è scelto di introdurre, nel codice delle pagine dei progetti dedicate a questa funzione, i metadati finalizzati a descrivere il *Rating*.

Nello specifico, è stato utilizzato l'itemtype *AggregateRating* per il quale sono stati definite gli itemprop:

- *itemReviewed*: il titolo dell'elemento votato;
- *ratingValue*: il voto ottenuto dal progetto, ovvero la media dei voti ottenuti durante le recensioni degli utenti;
- *bestRating*: il voto massimo ottenibile dal progetto, in questo caso pari a 5;
- *reviewCount*: il numero di voti assegnati dagli utenti al progetto.

Questa integrazione consentirà di ottenere un risultato di ricerca, per i progetti di cui sono state pubblicate recensioni e quindi è stato attribuito un voto, ricco di informazioni immediatamente visibili e attrattive, aumentano le possibilità i navigatori propendano per questo rispetto ad altri risultati semplici.

## **5.4 Estensioni future**

Come anticipato in fase introduttiva, la versione attuale del progetto “Volontieri” non prevede ancora l’automatizzazione del processo di registrazione ed inserimento dei relativi progetti, considerata parte secondaria ai fini esplicativi dell’argomento di tesi, ma comunque necessaria per una sua completezza e quindi solo rimandata ad un futuro dell’applicazione.

Possibili estensioni al progetto potrebbero riguardare l’integrazione di altri Social Network all’interno dell’applicativo per la fase di autenticazione e di condivisione automatica delle operazioni di recensione, come ad esempio Twitter o Google+, cercando di intercettare anche quegli utenti che preferiscono utilizzare o che possiedono soltanto queste tipologie di profili sociali.

Altro importante sviluppo riguardare un utilizzo ancor più ampio ed approfondito delle chiamate alle Graph API, in modo da riuscire a fornire un numero maggiore di informazioni correlate ad un progetto.

Un esempio, mirato a coinvolgere ancora di più l’utente e spingerlo ad approfondire la conoscenza di un progetto, potrebbe vedere il suggerimento di un secondo livello di conoscenza all’utente, in sostanza suggerendo, nello specifico, se anche un amico di un proprio amico ha eseguito attività recensorie di un progetto.

# Conclusioni

Grazie a questo progetto di tesi ho potuto approfondire uno dei mondi di maggiore interesse in ambito web dell'ultimo decennio, quello dei Social Network.

Un settore che ha visto una crescita ed una diffusione senza pari, al punto tale da essere diventato protagonista di un'importante parte delle azioni quotidiane di miliardi di persone.

Nell'ambito della redazione della tesi, gran parte dell'attenzione è stata focalizzata sul social network Facebook. Prima di tutto per l'importanza e il ruolo che ricopre, visto il suo primato in termini di utenza e di utilizzo, in secondo luogo data la varietà e la completezza degli strumenti messi a disposizione degli sviluppatori per la sua integrazione con sistemi web esterni.

Quest'ultimo aspetto spiega, infatti, il motivo per cui è sempre maggiore il numero dei siti internet nel quale è possibile incontrare integrazioni social atte a consolidare ed aumentare il bacino dei propri utenti.

Questa "contaminazione" all'interno dei siti web, consente di velocizzare e garantire una maggior semplicità nel compimento delle comuni azioni eseguibili all'interno di un sito, come l'operazione di login o di commento di un articolo, ma anche di assicurare un miglior coinvolgimento degli utenti grazie alla ormai comune familiarità con gli strumenti social e all'estensione delle azioni eseguibili, come porre un "like" ad un articolo o dividerlo sulla propria bacheca.

Sulla base di questo insieme di evidenze e di considerazioni è nata l'idea di sviluppare e realizzare un applicativo web che potesse sfruttare le potenzialità intrinseche alla base delle reti sociali sopra descritte, per promuovere le attività svolte da realtà di volontariato. Associazioni per le quali è di vitale importanza far conoscere ciò per cui quotidianamente si adoperano ed impegnano.

Durante la fase di sviluppo iniziale, una volta impostato l'ambiente di lavoro le cui basi poggiano sull'utilizzo del CMS WordPress, si è da subito iniziata l'analisi

e l'inserimento dello schema di metadati forniti da Facebook per la definizione arricchita di contenuti, l'Open Graph Protocol.

L'utilizzo di questo semplice linguaggio di markup, reso disponibile come protocollo aperto, si è dimostrato intuitivo e veloce, essendo basato su uno standard noto come RDFa e utilizzando un naming facile da interpretare.

La sua estensione, in termini di oggetti modellabili, definibile utilizzando le Custom Story di Facebook, mi ha permesso di approfondire e conoscere come sia immediato, grazie agli strumenti forniti dalla platform Facebook per gli sviluppatori, aumentare le possibilità di utilizzo e il livello di personalizzazione del contenuto descrivibile dal protocollo.

Naturalmente l'obiettivo massimo dovrebbe essere quello di poter descrivere in modo nativo un maggior numero di tipologie di contenuto, cosa che ad oggi è consentito solo oggetti di tipo website, musica, libri e film. Così facendo non ci sarebbero contenuti dipendenti da un'applicazione necessariamente legata ad un solo ed unico social network.

Altra implementazione studiata ed integrata all'interno del progetto è stata quella riguardante l'autenticazione tramite social login, pratica oggi frequente e comunemente utilizzata in un vasto numero di importanti siti internet ma soprattutto in svariate applicazioni mobile.

Grazie all'utilizzo di un'applicazione, creata per lo scopo, si è integrata la possibilità di registrarsi al sito, eseguendo un semplice collegamento con la propria identità sociale.

L'utilizzo di questa tecnica non solo è capace di garantire un aumento della probabilità di ricevere dati veritieri dagli utenti, essendo questi pervenuti da informazioni contenute nelle identità Facebook di questi ultimi, ma anche una velocizzazione del processo di registrazione e di reperimento dei dati utente, evitando di dover richiedere loro, un ennesimo inserimento di abbinato username e password per l'autenticazione, ovvero ulteriori informazioni da dover ricordare.

Grazie a questo particolare passaggio di sviluppo si è potuto evincere come sia aumentato, con il prosieguo dei rilasci delle versioni delle Graph API, le quali hanno raggiunto nell'ultimo anno la versione 2.2, il rispetto nei confronti dei dati e della privacy dei propri utenti, garantendo loro di poter decidere quali informazioni fornire alle applicazioni di terze parti, durante la fase di accettazione di queste,

portando gli sviluppatori a fare un uso maggiormente consapevole della richiesta dei permessi. Il risultato voluto, infatti, è quello di richiedere loro solo lo stretto indispensabile per le finalità dei singoli progetti, inducendoli inoltre a gestire anche possibili situazioni in cui utenti differenti possano fornire differenti permessi.

Ultima principale funzionalità analizzata e implementata, fulcro dell'applicativo e dell'analisi, ha riguardato l'utilizzo delle Custom Story di Facebook, componente fondamentale per coniugare l'esperienza sociale a comuni operazioni web.

Grazie a questa implementazione è stato possibile la descrizione avanzata di azioni, sotto forma di una tripla *utente-azione-oggetto*, simile ai concetti disegnabili grazie alle triple RDF, tramite la quale è stata definita una storia collegata alla recensione di un progetto, dove l'utente potrà raccontare la propria esperienza vissuta.

Di base questa azione, se fosse stata limitata soltanto ad un'operazione di pubblicazione di un commento sul sito Volentieri, probabilmente non avrebbe la capacità di raggiungere il numero sperato di utenti, necessario per promuovere un progetto nel modo migliore.

Utilizzando questa implementazione invece l'oggetto sociale "progetto", diventato tale grazie all'utilizzo dell'Open Graph Protocol, apparirà all'interno di una notifica di azione nel news feed delle amicizie Facebook dell'utente che l'ha compiuta permettendone una visibilità decisamente maggiore.

Di qui l'importanza descrittiva fornita dalle Custom Stories, capaci quindi di estendere il dizionario di azioni base di Facebook come il "like di un articolo" o "l'ascolto di una canzone", grazie alla definizione di nuovi oggetti e nuove azioni.

In conclusione, si è potuto quindi evincere che l'integrazione dei metadati Open Graph sta diventando sempre di più una consuetudine da parte degli sviluppatori, aumentandone incredibilmente la diffusione e il numero di servizi capaci di interpretarli. Spesso però la sua estensione è limitata all'utilizzo all'interno di uno solo di questi servizi e a differenti sistemi e metodi di interrogazione.

Data quindi la percentuale di utilizzatori di queste informazioni all'interno dei propri siti e alle possibili evoluzioni implementate nelle future versioni del protocollo, nonostante non possa essere considerato come la svolta risolutiva in ambito semantico, ha sicuramente dato e continuerà a dare una forte spinta e un

importante aiuto alla descrizione di un World Wide Web che negli anni ha saputo crescere in modo esponenziale in termini di quantità di informazioni descritte e di tecnologie con il quale descriverle, ma che ancora non ha ancora saputo strutturare in modo completo le sue potenzialità al fine di permetterne una migliore interpretazione automatizzata.





---

# Bibliografia

- [BER01] T. Berners-Lee, “*Giant Global Graph*”,  
<http://dig.csail.mit.edu/breadcrumbs/node/215>
- [BMF14] D. Brickley, L. Miller, “FOAF Vocabulary Specification 0.99”, 2014  
<http://xmlns.com/foaf/spec/>
- [CUR13] A. Curtis, “*The Brief History of Social Media*”,  
<http://www2.uncp.edu/home/acurtis/NewMedia/SocialMedia/SocialMediaHistory.html>, 2013
- [BEL07] D.M. BOYD, N.B. ELLISON, “Social Network Sites: Definition, History”, and Scholarship, 2007  
<http://onlinelibrary.wiley.com/doi/10.1111/j.1083-6101.2007.00393.x/full/>
- [COI – 2014] Facebook Company Info, <http://newsroom.fb.com/company-info/>
- [FAR07] D. Farber, “*Facebook's Zuckerberg uncorks the social graph*”,  
<http://www.zdnet.com/blog/btl/facebooks-zuckerberg-uncorks-the-social-graph/5156>, 2007
- [FIE00] R. T. Fielding, “Architectural Styles and the Design of Network-based Software Architectures”,  
<http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>
- [FUR07] J. Furrer, “*Official Definition of Facebooks Social Graph – Dave Morin from Facebook*”, <http://podtech.wordpress.com/2007/10/09/official-definition-of-facebooks-social-graph-dave-morin-from-facebook-graphing-social-patterns-day-2/>, 2007
- [MAQ - 10] L. Mancini, R. Queri, “*Il protocollo OAuth*”  
<http://www.dmi.unipg.it/bista/didattica/sicurezza-pg/seminari2010-11/oauth/OAuth%20-%20Sicurezza%20Informatica%202010:2011%20-%20Mancini%20Queri.pdf>
- [WIK-FAC] Wikipedia l’enciclopedia libera, “*Facebook*”,  
[http://en.wikipedia.org/wiki/History\\_of\\_Facebook](http://en.wikipedia.org/wiki/History_of_Facebook)
- [WIK - FOA] Wikipedia l’enciclopedia libera, “*FOAF*”,
-

<http://it.wikipedia.org/wiki/FOAF>

[WIK-GIA] Wikipedia l'enciclopedia libera, "*Giant Global Graph*",

[http://en.wikipedia.org/wiki/Giant\\_Global\\_Graph](http://en.wikipedia.org/wiki/Giant_Global_Graph)

[WIK-JSON] Wikipedia l'enciclopedia libera, "*Social Login*",

<http://it.wikipedia.org/wiki/JSON>

[WIK-SEM] Wikipedia l'enciclopedia libera, "*Web Semantico*",

[http://it.wikipedia.org/wiki/Web\\_semantico](http://it.wikipedia.org/wiki/Web_semantico)

[WIK-SOC] Wikipedia l'enciclopedia libera, "*Social Login*",

[http://en.wikipedia.org/wiki/Social\\_login](http://en.wikipedia.org/wiki/Social_login)

[WOR – 2014] W3Techs, "*Usage statistics and market share of WordPress for websites*", <http://w3techs.com/technologies/details/cm-wordpress/all/all>