



Educating to free software, with free software. For a democratic and emancipatory education Educare al software libero, con il software libero. Per un'educazione democratica ed emancipatrice

Pietro Corazza

Department of Education Studies "Giovanni Maria Bertin", Università degli Studi di Bologna – pietro.corazza2@unibo.it
<https://orcid.org/0000-0002-1706-3118>

OPEN ACCESS



DOUBLE BLIND PEER REVIEW

ABSTRACT

This article aims to highlight the educational implications of a central issue regarding digital technologies: that relating to the type of control that the creators of a program exercise over it, which is expressed by the distinction between Free Software and proprietary software. In the first part of the article, the history of the birth of proprietary and free software will be retraced, also highlighting the distinction between Free Software and Open Source. Starting from this historical retrospective, the second part of the article will consider the educational dimension of the issue, showing how the Free Software approach appears to be consistent with the objectives of an emancipatory education, while that of proprietary software risks instead to accentuate the tendency towards a neo-liberal evolution of educational institutions. Then, two complementary directions for promoting Free Software in education will be proposed. The first consists in the adoption of Free Software by public educational institutions, in order to oppose the current tendency to contract out the management of data and digital infrastructures to private multinationals. The second consists in fostering an educational approach to the study and creation of computer programs that is animated by the philosophy of collaboration and sharing that underlies Free Software and hacker pedagogy.

Questo articolo ha l'obiettivo di mettere in luce le implicazioni educative di una questione di importanza centrale nell'ambito delle tecnologie digitali: quella relativa al tipo di controllo che i creatori di un programma informatico esercitano su di esso, la quale si esprime nella distinzione tra software libero e software proprietario. Nella prima parte dell'articolo verrà ripercorsa la storia della nascita del software proprietario e di quello libero, evidenziando anche la distinzione, interna al mondo del software libero, tra Free Software e Open Source. A partire da questa retrospettiva storica, nella seconda parte verrà presa in considerazione la dimensione più prettamente educativa della questione, mostrando come l'approccio del software libero (soprattutto nella declinazione del Free Software, piuttosto che dell'Open Source) appaia maggiormente in linea con l'orizzonte di un'educazione emancipatrice, mentre quello del software proprietario rischi invece di accentuare la tendenza verso un'evoluzione in senso neoliberalista delle istituzioni educative. A partire da tale analisi, verranno proposte due direzioni, complementari, di promozione del software libero in ambito educativo. La prima consiste nell'utilizzo di software libero da parte delle istituzioni educative pubbliche, per contrastare l'attuale tendenza ad appaltare a multinazionali private la gestione dei dati e delle infrastrutture digitali. La seconda consiste nel favorire un approccio allo studio e alla creazione dei programmi informatici animata dalla filosofia di collaborazione e condivisione che sta alla base del software libero e della pedagogia hacker.

KEYWORDS

Free Software, Emancipatory Education, Hacker Pedagogy, Platformisation of Education, Digital Sovereignty
Software libero, Educazione emancipatrice, Pedagogia hacker, Piattaformizzazione dell'educazione, Sovranità digitale

Citation: Corazza, P. (2023). Educating to free software, with free software. For a democratic and emancipatory education. *Formazione & insegnamento*, 21(1), 71-80. https://doi.org/10.7346/-fei-XXI-01-23_10

Copyright: © 2023 Author(s).

License: Attribution 4.0 International (CC BY 4.0).

Conflicts of interest: The Author(s) declare(s) no conflicts of interest.

DOI: https://doi.org/10.7346/-fei-XXI-01-23_10

Received: February 9, 2023 • **Accepted:** April 18, 2023 • **Published:** April 30, 2023

Pensa MultiMedia: ISSN 2279-7505 (online)

1. Introduzione

Negli ultimi decenni, a partire dalla comparsa del Web, le trasformazioni nel campo delle tecnologie digitali si stanno susseguendo a un ritmo incalzante, inducendo numerosi studiosi e studiose di differenti discipline a interrogarsi sull'intreccio tra innovazione tecnologica e mutamenti sociali. Anche in ambito pedagogico simili riflessioni stanno assumendo sempre maggiore importanza (Cappello, 2010; Margiotta, 2019; Panciroli et al., 2020). Adottando uno sguardo generale, è possibile chiedersi in che misura gli sviluppi tecnologici stiano contribuendo a ridefinire il significato di alcuni degli elementi centrali del mondo educativo: il ruolo di insegnanti, studenti e genitori; i confini e le caratteristiche degli ambienti educativi (formali, non formali e informali); i concetti stessi di apprendimento ed educazione (Selwyn, 2011). Più nello specifico, appare cruciale confrontarsi con le caratteristiche di quella che è probabilmente la tipologia di organizzazione socio-tecnica più rilevante nel panorama odierno: la piattaforma digitale (Srniczek, 2017). Le piattaforme digitali costituiscono una particolare tipologia di azienda che funziona come infrastruttura di intermediazione finalizzata a far interagire diverse tipologie di attori, come ad esempio clienti, inserzionisti, prestatori di servizi, produttori, fornitori, ma anche oggetti fisici e macchine (Bratton, 2015). Comprendere il funzionamento delle piattaforme digitali appare oggi quantomai necessario perché si tratta di un modello organizzato che sta espandendo la propria influenza in quasi ogni ambito della vita umana (Burrell & Fourcade, 2021), tra cui quello educativo (Poell & van Dijck, 2018).

Lo sviluppo pervasivo delle piattaforme digitali pone in evidenza alcune questioni centrali per chi si occupa di educazione: in che modo il funzionamento delle piattaforme genera nuove forme di inclusione o esclusione digitale, e come queste si intrecciano con le disuguaglianze socio-economiche? Quali dinamiche economiche condizionano e orientano il funzionamento delle piattaforme digitali, e come queste si riflettono sulla crescente "datificazione" dei processi educativi? In che modo le piattaforme digitali stanno contribuendo a ridefinire il concetto di apprendimento in base alla suddivisione tra apprendimento umano e apprendimento svolto dalle macchine? Quanto la penetrazione di grandi piattaforme digitali private negli ambienti educativi, e specialmente nelle istituzioni scolastiche, ne sta condizionando metodi e obiettivi? (Selwyn et al., 2020).

Questo articolo intende contribuire al dibattito pedagogico che ruota attorno a queste domande, a partire dall'indagine di una questione centrale, ovvero quella relativa al tipo di controllo che i creatori di un programma informatico esercitano su di esso, questione che si condensa nella distinzione tra software libero e software proprietario. Il software libero si caratterizza per il fatto di garantire quattro libertà fondamentali:

- 1) la libertà di eseguire il programma in qualunque contesto, per qualsiasi scopo;
- 2) la libertà di studiare come funziona il programma e di modificarlo in modo da adattarlo alle proprie esigenze;
- 3) la libertà di ridistribuire co-

pie del programma al fine di aiutare altri programmatori;
- 4) la libertà di migliorare il programma e di distribuirne pubblicamente i miglioramenti, in modo che tutti ne traggano beneficio (Free Software Foundation, 2022a).

Nel caso del software proprietario, invece, l'utilizzo, la ridistribuzione e la modifica sono proibite oppure richiedono un permesso, il quale solitamente dipende da una qualche forma di pagamento, diretto o indiretto, nei confronti del proprietario. In prima battuta, questa distinzione tra software libero e proprietario potrebbe apparire una questione di esclusiva pertinenza giuridico-economica, della quale non appaiono ben chiare le implicazioni e le ricadute sociali e politiche. Lo scopo di questo articolo è pertanto quello di mettere in luce tali implicazioni, concentrandosi in particolare su quelle riguardanti il mondo educativo.

L'articolo si comporrà di due parti. Nella prima verrà ripercorsa la storia della nascita del software proprietario e di quello libero, nella convinzione che la conoscenza di tali sviluppi storici costituisca un presupposto fondamentale per comprendere le caratteristiche e le relazioni tra questi due modelli. In questa prima parte verrà anche messa in luce una ulteriore distinzione interna al mondo del software libero, ovvero quella tra Free Software e Open Source. Si vedrà come, nonostante questi due termini vengano spesso accomunati o utilizzati in modo interscambiabile, essi indichino in realtà due approcci che si fondano su presupposti economico-politici significativamente diversi: il Free Software è nato con un intento politico, ovvero la difesa della libertà del software da forme di chiusura proprietaria, mentre l'Open Source ne costituisce una rielaborazione in cui tale obiettivo politico viene subordinato, e in alcuni casi soppiantato, da finalità commerciali.

Nella seconda parte verrà presa in considerazione la dimensione più prettamente educativa della questione, mostrando come l'approccio del software libero (soprattutto nella declinazione del Free Software, piuttosto che dell'Open Source) appaia maggiormente in linea con l'orizzonte di un'educazione emancipatrice, mentre quello del software proprietario rischi invece di accentuare la tendenza verso un'evoluzione in senso neoliberalista delle istituzioni educative. A partire da tale analisi, verranno proposte due direzioni, complementari, di promozione del software libero in ambito educativo. La prima consiste nell'utilizzo di software libero da parte delle istituzioni educative, per contrastare l'attuale tendenza ad appaltare a multinazionali private la gestione dei propri dati e delle proprie infrastrutture digitali. La seconda consiste nel favorire un approccio allo studio e alla creazione dei programmi informatici animata dalla filosofia di collaborazione e condivisione che sta alla base del software libero e di quella che è possibile definire come pedagogia hacker.

2. Software proprietario, Free Software e Open Source

Oggi l'idea che i software che utilizziamo siano considerati proprietà di un'azienda, che spesso ne con-

cede l'utilizzo in cambio di un compenso, sembra essere la norma, ma non è sempre stato così. Al contrario, il software è nato in un contesto di libera condivisione, e solo in un secondo momento è stato sottoposto a limitazioni basate sul concetto di proprietà (Ippolita, 2005). Infatti, fino alla fine degli anni Settanta il lavoro nei luoghi in cui si studiavano e si scrivevano i programmi, ovvero le università e i laboratori di ricerca, si basava in larga parte sulla libera condivisione dei codici, che costituiva un presupposto fondamentale per lo sviluppo della ricerca¹. Questa prassi di condivisione era legata al fatto che il mercato informatico era concentrato sull'hardware: i produttori competevano tra loro per vendere le macchine, che costituivano allora la componente più costosa e difficile da realizzare, e offrivano gratuitamente il software necessario al loro funzionamento, nella speranza di attrarre i clienti.

Negli anni Ottanta però la situazione è cambiata: è aumentata in modo esponenziale l'importanza commerciale del software, che ha smesso di rappresentare una componente secondaria rispetto all'hardware per diventare invece un terreno di contesa cruciale nella competizione tra le aziende tecnologiche. Questo cambiamento è stato formalizzato, anche a livello legislativo, nel "Computer Software Copyright Act" del 1980, che ha ampliato la precedente legge sul copyright definendo i limiti di applicazione del concetto di diritto d'autore nell'ambito dei programmi informatici (Seligmann, 2021). Così la pratica della libera condivisione ha ceduto il posto alla distribuzione proprietaria: i programmi informatici venivano sviluppati dalle aziende e rilasciati in una modalità che ne consentiva l'uso ma non l'accesso al codice sorgente. Quando un software viene rilasciato senza consentire l'accesso al codice sorgente significa che chi ne entra in possesso non può analizzarne il funzionamento, né apportarvi modifiche. In precedenza, i programmi erano una sorta di opera collettiva in continua evoluzione, alla quale numerosi programmatori collaboravano, perché ognuno aveva accesso alla totalità del codice e poteva modificarlo ed ampliarlo. Dall'avvento dell'approccio proprietario invece i software hanno iniziato ad essere concepiti come prodotti finiti e non modificabili dagli utenti o da altri programmatori, perché la loro realizzazione è diventata prerogativa esclusiva dall'azienda produttrice, la quale si occupa di correggerli e ampliarli, rilasciando nel corso del tempo versioni più aggiornate del programma stesso. Un esempio paradigmatico di questa tendenza è l'ascesa di Microsoft, che nel corso degli anni Ottanta e Novanta è riuscita a guadagnarsi una straordinaria posizione di monopolio: il suo sistema operativo Windows si è affermato sostanzialmente come unica alternativa, presente su ogni computer ad eccezione di quelli prodotti da Apple (che rappresenta un'eccezione in quanto produce sia il proprio hardware che i propri software, creando un ecosistema completo e il più possibile chiuso nei confronti dell'esterno).

È dunque in questo clima orientato alla privatizzazione e alla commercializzazione dei codici che Richard Stallman ha dato vita al movimento del Free Software. La spinta alla creazione di tale movimento si è manifestata quando Stallman, che lavorava nel laboratorio di Intelligenza Artificiale del MIT, si è reso conto di alcuni malfunzionamenti della stampante presente nel laboratorio, che a suo parere si sarebbero potuti risolvere con semplici modifiche al codice informatico della stampante stessa. Egli ha quindi chiesto alla Xerox, l'azienda produttrice della stampante, di poter accedere al codice sorgente per apportarvi alcune modifiche, ma la Xerox si è rifiutata, appoggiandosi al "Computer Software Copyright Act". Questa vicenda ha convinto Stallman della necessità di combattere l'approccio proprietario al software, perché questo risulta contrario all'interesse degli utilizzatori dei software e impedisce loro di comprenderne il funzionamento e di contribuire al loro miglioramento (Williams, 2002). Il Free Software è nato quindi con un intento dichiaratamente politico, i cui fondamenti sono riassunti nelle quattro libertà già citate.

Uno dei principali obiettivi verso cui è stata condotta la battaglia del Free Software è l'applicazione del *copyright* ai software: ad esso è stato quindi contrapposto il concetto di "*copyleft*", che ne costituisce il ribaltamento. Il *copyleft* consiste in un permesso che l'autore di un codice appone su di esso, dichiarando che quest'ultimo può essere liberamente distribuito e modificato, a condizione però che le copie o le versioni modificate vengano distribuite anch'esse attendendosi ai principi del *copyleft*, cioè che non vengano sottoposte a forme di chiusura proprietaria (Free Software Foundation, 2022b). Questa specifica è necessaria per garantire che i software liberi rimangano tali: precedentemente, infatti, esisteva la possibilità di rilasciare codici definendoli come "di pubblico dominio", ma questi potevano essere poi utilizzati per realizzare software proprietario, smettendo quindi di essere accessibili.

Il movimento del Free Software aveva anche risvolti etici ed economici: l'intento era quello di privilegiare il contributo creativo e spontaneo dei programmatori al progetto comune, rispetto alla loro sottomissione a rigide logiche aziendali, che dettavano compiti e tempistiche sulla base delle esigenze commerciali. Ciò non significava rinunciare ad ogni forma di compenso per il lavoro svolto, ma piuttosto proporre una tipologia di remunerazione diversa da quella dominante in ambito aziendale: invece che fondare i profitti sulla vendita dei prodotti (i programmi, che dovevano per questo essere proprietari), il movimento propose di ricavarli dalla vendita di servizi, nella forma di assistenza offerta dai programmatori agli utenti che usufruivano dei loro software. Si sperava in questo modo di organizzare la produzione di software secondo criteri più attenti ai bisogni umani dei programmatori, che avrebbero ricevuto un compenso minore rispetto al modello aziendale, ma avrebbero in cambio beneficiato di ritmi di lavoro meno stressanti e più gratificanti, orientando il lavoro sulla base della passione e di quello che ritenevano più utile ad un migliore funzionamento dei programmi, e non sulla spinta delle esigenze commerciali che li avrebbero costretti a realizzare continuamente

1 Occorre precisare che già a partire dagli anni Sessanta sono esistiti casi di software distribuiti in forma proprietaria, come ad esempio il sistema operativo OS/360 di IBM e l'edizione speciale del sistema operativo PDP-8 della DEC. Tuttavia, queste forme di software proprietario rappresentavano delle eccezioni, mentre la libera condivisione del software era la norma.

nuove versioni dei software con il solo fine di conquistare maggiori fette di mercato (Ippolita, 2005).

A partire dal 1983 Stallman è riuscito a radunare attorno a sé una ampia comunità di programmatori animati dalle sue medesime convinzioni, la quale, adottando una peculiare modalità organizzativa decentralizzata definita metodo “bazar” (Raymond, 1999), è riuscita nell'intento di scrivere collettivamente un sistema operativo libero. Negli anni Novanta il sistema operativo GNU/Linux è stato ultimato ed era pronto per essere diffuso e tentare di mettere in discussione il monopolio di Microsoft nel campo dei sistemi operativi.

Ma è in questo stesso periodo che si è verificata la svolta che ha portato il movimento Free Software, fino ad allora unitario, alla biforcazione da cui è nata la variante dell'Open Source. Il 1998 ha segnato un momento cruciale da questo punto di vista, in quanto ha visto la comparsa del browser Mozilla, rilasciato dall'azienda Netscape, che è stato realizzato e distribuito secondo i principi dell'Open Source, e progressivamente è riuscito a diffondersi fino a fare concorrenza a Explorer, il browser proprietario venduto da Microsoft. Parallelamente, sul piano teorico, il concetto di Open Source è stato definito in alcune pubblicazioni che ne hanno esplicitato le differenze rispetto al Free Software (Raymond, 1998; Barley & O'Mahony, 2002). Occorre soffermarsi sulla distinzione tra questi due approcci perché, nonostante nascano da una matrice comune e vengano spesso accomunati o addirittura usati in maniera interscambiabile, incarnano due visioni sensibilmente diverse.

Il termine “Open Source” è stato infatti coniato per una ragione prettamente economica: quella di rendere l'approccio del Free Software più appetibile alle aziende tecnologiche, per fare in modo che questo lo adottassero come proprio modello organizzativo e produttivo (Ippolita, 2005). Infatti, il termine “free”, che in inglese ha il duplice significato di libero e gratuito, poteva trasmettere l'idea che il software dovesse rimanere necessariamente gratuito, e quindi rendesse impossibile applicare questo approccio in ambito commerciale (anche se, come si è visto, il Free Software non esclude ogni tipo di risvolto commerciale, semplicemente non considera il profitto come una priorità). Il termine “open” invece si limita ad affermare che il software è aperto, ovvero liberamente accessibile: si tratta dunque di un termine politicamente più neutro, che permette più facilmente di integrare il movimento all'interno di logiche di mercato.

Infatti, l'obiettivo economico del movimento Open Source appare evidente fin dal rilascio di Mozilla: il suo scopo è quello di sottrarre fette di mercato al dominio di monopolisti come Microsoft, permettendo ad altre aziende di competere in un ambito che fino ad allora era rimasto praticamente inaccessibile. Questo è possibile perché la collaborazione volontaria e gratuita da parte dei numerosissimi partecipanti ai progetti di Open Source permette di risparmiare enormemente sui costi di produzione del software, dal momento che non è necessario pagare il lavoro dei programmatori, garantendo così alle aziende che utilizzano codici aperti di colmare il divario rispetto a giganti come Microsoft. Dunque, la differenza fondamentale tra il Free Software e l'Open Source è che il secondo è esplicitamente orientato a finalità commer-

ciali, mentre il primo aveva come priorità la difesa di un principio politico: quello della libertà del software da forme di chiusura proprietaria.

Tale libertà non è più completamente garantita nel movimento Open Source, poiché al suo interno è possibile rilasciare software vincolati da diversi tipi di licenze, alcune delle quali non rispettano i principi del *copyleft*, bensì consentono di incorporare i codici liberi dentro software proprietari, sottoponendoli dunque a pratiche di chiusura (Stallman, 2007). Questo cambiamento sostanziale ha permesso che software prodotti in ambito Open siano stati integrati nei processi produttivi di grandi aziende *hi-tech* come IBM: questa impresa, che negli anni Settanta dominava il mercato, ha vissuto un tracollo che l'ha portata vicino al fallimento verso la fine degli anni Ottanta, così per risollevarsi ha scelto di ristrutturare radicalmente il proprio modello aziendale incorporando pratiche di produzione collaborativa. «*Cooperate on standards, compete on solutions*» è ciò che recita un motto dell'azienda: la strategia è quella di acconsentire a rilasciare apertamente alcune parti basilari dei software prodotti al proprio interno per beneficiare in cambio della collaborazione gratuita di moltissimi programmatori volontari, riservandosi poi uno spazio di competizione nella realizzazione delle componenti più sofisticate che vengono aggiunte ai programmi di base. In questo modo le aziende possono sfruttare la collaborazione gratuita dei programmatori, ma al tempo stesso mantenere segrete le componenti più strategiche dei propri codici. Ciò è evidente nella pratica di rilasciare pubblicamente le proprie API, di cui si servono i principali giganti del Web come Google, Apple e Amazon. Le API (*Application Programming Interface*) sono dispositivi che permettono a programmatori indipendenti di accedere ad alcune parti del codice delle aziende per contribuire a migliorarle ed ampliarle, mantenendo però inaccessibili le sezioni più profonde e strategiche: così è possibile beneficiare dei contributi esterni pur continuando a custodire i propri segreti industriali. In questo modo Google può presentarsi come un'azienda animata dalla filosofia Open Source, mentre custodisce scrupolosamente segreto il funzionamento del proprio algoritmo di ricerca.

In sintesi, si potrebbe affermare che il modello economico del Free Software è paragonabile ad una bottega di artigiani, mentre quello dell'Open Source ad una multinazionale (Ippolita, 2005). Differiscono sia per le dimensioni dei progetti che per quanto riguarda l'etica del lavoro, ma entrambi si collocano nel conteso del mercato e rispondono alle sue leggi. Tra i due, l'Open Source è riuscito a diffondersi maggiormente e a risultare più capace di fare concorrenza all'approccio proprietario al software: per questo motivo, è contro di esso che si levano le critiche e le accuse delle principali aziende fondate sul modello proprietario, Microsoft in primis (Jardin, 2005). La battaglia di Microsoft contro l'Open Source rappresenta una contrapposizione tra due diverse strategie di organizzazione aziendale: Microsoft rimane più legata ad una modalità tradizionale di scrittura dei programmi (definita “a cattedrale”), mentre le aziende che si basano in parte sull'Open Source, come Google o IBM, sfruttano maggiormente le potenzialità dell'organizzazione “a bazar” e i vantaggi economici

derivanti dal lavoro non pagato offerto dai programmatori che contribuiscono volontariamente alla scrittura di codice (Formenti, 2011).

3. Quali implicazioni per il mondo educativo?

Una volta chiarita le differenti origini del software proprietario e di quello libero, e messa in luce la ulteriore distinzione tra Free Software e Open Source, è possibile trarne alcune riflessioni riguardanti il mondo educativo.

La prima e fondamentale considerazione è che l'approccio del software libero appare maggiormente in linea con l'orizzonte di un'educazione democratica ed emancipatrice, mentre l'utilizzo del software proprietario rischia invece di accentuare la tendenza verso un'evoluzione in senso neoliberista delle istituzioni educative. Ciò risulta piuttosto evidente se si considera una elementare questione di potere: chi controlla l'infrastruttura su cui si basa un certo sistema ha la possibilità di esercitare un'influenza notevole sul sistema stesso. Questo vale anche per le istituzioni educative, la cui dipendenza dalle tecnologie digitali sta progressivamente crescendo.

Appare ormai assodato che nella nostra epoca i dati sono diventati una delle risorse più preziose, il che ha generato una competizione serrata per riuscire a raccoglierne il più possibile (Srnicek, 2017; Zuboff, 2019). In questa competizione, un ruolo cruciale è svolto dalle infrastrutture digitali attraverso cui i dati vengono raccolti, trasmessi, immagazzinati, integrati e analizzati: chi possiede le piattaforme digitali che svolgono una o più di queste funzioni si trova dunque in una posizione di enorme potere.

Anche il funzionamento delle nostre istituzioni educative è quotidianamente basato su ingenti flussi di dati significativi e sensibili: in primo luogo si tratta di dati su studenti e studentesse, relativi ad esempio alle loro modalità di apprendimento, ai risultati ottenuti, alla frequenza, ma anche alle informazioni personali di ogni tipo che emergono da ciò che scrivono e dicono in classe. E considerazioni analoghe valgono per i dati riguardanti i docenti, il personale scolastico e le famiglie. La domanda che emerge è dunque semplice ma cruciale: a chi vogliamo affidare il controllo di questi dati così importanti? La questione, ovviamente, non è limitata solamente alla scelta tra il software libero e quello proprietario, ma riguarda l'uso delle tecnologie digitali in generale.

Attualmente le scuole italiane, nella stragrande maggioranza dei casi, stanno delegando la gestione di buona parte dei propri dati a grandi multinazionali private. Questa tendenza ha registrato un'impennata di proporzioni inaudite durante la pandemia del Covid-19, durante la quale si è presentata la necessità imprevista e improvvisa di svolgere a distanza molte attività che precedentemente erano condotte in presenza. La maggior parte delle istituzioni educative si sono trovate impreparate: non disponevano di piattaforme adatte alla didattica a distanza (DAD), né erano familiari con il loro utilizzo. La scelta delle piattaforme da adottare è rimasta competenza delle singole scuole, che inizialmente si sono dovute organizzare in totale autonomia, mentre da marzo 2020 il Ministero dell'Istruzione ha fornito sul proprio sito un

elenco di risorse consigliate per lo svolgimento della DAD: nella sezione dedicata alle piattaforme, il sito del Ministero presentava come opzioni solamente *G-Suite for Education* di Google, *Office 365 Education* di Microsoft e *Weschool* di Tim (quest'ultima è stata aggiunta in un secondo momento). Riferisce Zoja (2022) di un'indagine svolta coinvolgendo la quasi totalità delle scuole italiane, che permette di osservare come queste tre piattaforme siano state quelle maggiormente adottate a livello nazionale, con una netta predominanza di Google: l'86,3% delle scuole ha dichiarato di utilizzare Google, il 18% Microsoft, il 6,2% Weschool, mentre solo il 14,7% ha affermato di utilizzare altre piattaforme rispetto alle tre precedenti. Come si può notare dai dati, alcune scuole hanno dichiarato di utilizzare più di una piattaforma. Anche per quanto riguarda i servizi di video-chiamata utilizzati in aggiunta a quelli resi disponibili dalle piattaforme, è riscontrabile una prevalenza di programmi commerciali come GoToWebinar (36,7%), Whatsapp (32,9%), Zoom (31,9%), Skype (12,8%) e Webex (7,9%), rispetto a opzioni open source come Jitsi (7,3%) (Zoja, 2022).

A un primo sguardo, si potrebbe supporre che la scelta di affidarsi a grandi multinazionali sia stata motivata da ragioni di praticità e rapidità, dal momento che i servizi offerti dalle grandi aziende mediamente sono considerati più affidabili ed efficienti, poiché l'utilizzo da parte di molti utenti fornisce alle piattaforme grandi quantità di dati, che vengono usati per migliorare il servizio stesso. Ma, a ben guardare, questa scelta non appare così fondata, soprattutto da un punto di vista educativo. Infatti, i servizi per la DAD offerti da Google e Microsoft non sono stati originariamente pensati con uno scopo didattico, bensì aziendale, e all'avvento della pandemia sono stati frettolosamente riadattati aggiungendo alcune funzioni quali l'alzata di mano e pochi altri dettagli secondari. Invece esistevano alternative open source progettate espressamente per fini didattici che sono state ignorate. E questa mancanza di diffusione delle alternative non proprietarie non dipende solo dalla loro minore visibilità prima dell'avvento della pandemia, ma anche dalla scelta ministeriale di presentare sul proprio sito solamente tre opzioni commerciali.

Oltre al fatto che determinate piattaforme fossero o meno state progettate con intenti specificamente didattici, il problema fondamentale rimane la tendenza a cedere ad aziende private un ruolo strategico nel funzionamento di istituzioni educative pubbliche. Queste aziende stanno espandendo in modo impressionante la loro penetrazione in ambito educativo: basti pensare che Google nel febbraio 2021 dichiarava di aver raggiunto i 150 milioni di utenti del software *Classroom* a livello mondiale, mentre solo un anno prima ne contava 40 milioni (Lazare, 2021). Tale penetrazione può apparire innocua, o persino conveniente, per le istituzioni educative, perché offre servizi utili e prevalentemente gratuiti. Ma è evidente che se un'azienda agisce in questo modo è perché ottiene un tornaconto di diverso tipo: è possibile evidenziare per lo meno due dimensioni cruciali di tale tornaconto, ed entrambe risultano problematiche.

In primo luogo, come già rilevato, tali multinazionali ottengono la possibilità di raccogliere dati preziosi, e altrimenti difficilmente reperibili, riguardanti

le molteplici attività che riguardano un'istituzione educativa. Come afferma Giulia Schneider, docente di Diritto dell'economia all'Università Cattolica di Milano, le piattaforme che operano in ambito educativo, sia a livello universitario che scolastico, «si dimostrano molto vaghe, anche ambigue, sulle modalità di gestione dei dati che raccolgono, nonché sulle finalità. Per i possibili usi secondari parlano del miglioramento del servizio, ma così aprono a una varietà di usi commerciali» (Zoja, 2022). Infatti, è stato messo in luce ormai chiaramente che le piattaforme digitali, in qualsiasi ambito operino, tendono sempre ad accumulare più dati possibile, di ogni genere, poiché anche quando non sanno ancora esattamente come utilizzarli, possono stare sicure che in futuro quei dati potranno risultare utili o che qualcuno sarà interessato ad acquistarli (Zuboff, 2019). Occorre inoltre ricordare che eseguire una copia dei dati non lascia tracce; perciò, non è possibile verificare con certezza quante copie vengano fatte e dove vengano conservate.

Un secondo obiettivo fondamentale di queste aziende, oltre alla raccolta dei dati, è il cosiddetto *vendor lock-in*, ovvero la fidelizzazione: l'azienda inizialmente offre servizi vantaggiosi in forma gratuita per attirare gli utenti, fino a sviluppare in loro una dipendenza da cui diventa poi molto difficile svincolarsi. Basti pensare a quanto risulta difficile e dispendioso per un'istituzione educativa cambiare il fornitore dei propri servizi informatici, perché significa dover formare tutto il personale all'utilizzo di nuovi sistemi e dover trasferire una grande mole di dati. Una volta consolidato un tale posizionamento strategico, l'azienda potrebbe decidere di utilizzarlo in vari modi. In primo luogo, potrebbe decidere di iniziare a richiedere un pagamento per i propri servizi. Oppure potrebbe condizionare in modo sempre più significativo le pratiche educative, ad esempio orientando le modalità di interazione, la tipologia e la modalità di utilizzo dei materiali didattici, le modalità di valutazione, ecc. Alcuni studiosi ipotizzano che, se la penetrazione delle multinazionali digitali continuasse ad intensificarsi, esse potrebbero cavalcare la tendenza già esistente alla privatizzazione delle istituzioni educative, fino ad arrivare ad ottenere un ruolo primario nella definizione degli obiettivi e dei metodi che le guidano (Hillman et al., 2020).

Dunque continuare ad appaltare con disinvoltura a grandi multinazionali alcune componenti chiave dell'infrastruttura digitale della scuola e dell'università pubblica risulta essere un atteggiamento fortemente problematico, che appare conveniente ed economico nell'immediato, ma risulta clamorosamente miope in una prospettiva più lungimirante.

Non si tratta tuttavia di un destino ineluttabile. Una possibile strategia alternativa consiste nel basarsi sul software libero. Tale strategia potrebbe essere sviluppata lungo due direttrici: da un lato, promuovendo lo sviluppo di software libero e utilizzandolo nelle istituzioni educative, dall'altro, favorendo all'interno di tali istituzioni un approccio allo studio e alla creazione dei programmi informatici animato dalla filosofia di collaborazione e condivisione che sta alla base del software libero.

3.1 Il software libero come condizione necessaria per mantenere il controllo democratico sulle istituzioni educative pubbliche

In primo luogo, al fine di mantenere il controllo democratico sui sistemi educativi, appare cruciale investire nello sviluppo e nell'adozione di software pubblico sul quale basare il funzionamento delle istituzioni educative. Va specificato che la questione non può esaurirsi nel definire chi controlla il software, ma occorre anche porsi il problema di come quest'ultimo viene costruito: se infatti rimanesse software proprietario, anche nel caso fosse controllato dallo stato, continuerebbe ad essere chiuso e non sottoponibile al pubblico scrutinio, mentre l'utilizzo di software libero consente un maggiore controllo democratico, oltre che la possibilità di allargare la comunità di programmatori e programmatrici che contribuiscono a svilupparlo e migliorarlo. È inoltre importante tenere in considerazione la differenza, illustrata precedentemente, tra Free Software e Open Source, che si esprime concretamente nei diversi tipi di licenze sotto le quali può essere rilasciato un software: se una specifica licenza rispetta le quattro libertà fondamentali del software libero rientra nel Free Software; se invece, ad esempio, consente di introdurre una qualche forma di chiusura proprietaria, può rientrare nell'ambito dell'Open Source ma non è più considerabile Free Software.

L'adozione di software libero nelle scuole e nelle università non soltanto è possibile, ma è anche esplicitamente prescritta dalla Legge 134/2012, che riguarda tutte le pubbliche amministrazioni, la quale stabilisce che «l'acquisto di software in licenza proprietaria sarà possibile soltanto quando la valutazione comparativa abbia dimostrato l'impossibilità di accedere a soluzioni in software libero o già sviluppate dalla pubblica amministrazione ad un prezzo inferiore». Tuttavia, tale norma è costantemente ignorata e disattesa. Questa sistematica violazione della legge è possibile perché non esistono sanzioni previste per assicurarne il rispetto. Tuttavia, alcuni giuristi vicini alla causa del software libero sostengono che esista la possibilità che i dirigenti di una istituzione educativa vengano chiamati a rendere conto delle proprie politiche di adozione del software di fronte alla Corte dei Conti (Meo, 2022).

L'adozione di software libero non è però una responsabilità che si possa scaricare sui singoli istituti scolastici, già oberati di lavoro e in carenza di risorse: certo potrebbero esserci esempi virtuosi, ma è facilmente prevedibile che la maggioranza delle scuole continui a scegliere di affidarsi ad aziende private, sia per mancanza di consapevolezza che di tempo e risorse. È invece fondamentale mettere in atto scelte di sistema, a livello nazionale, che offrano alle istituzioni educative la possibilità di passare al software libero in modo più agevole. Ad esempio, per incentivare e supportare la transizione, occorrerebbe investire risorse nella formazione del personale all'utilizzo di software liberi.

Scelte di questo tipo sono state già messe in atto a livello locale: ad esempio la provincia di Bolzano ha avviato nel 2004 la migrazione al software libero all'interno della pubblica amministrazione (Rossi et al., 2009), e dal 2005 ha lanciato un progetto di adozione

del software libero nelle scuole. Quest'ultimo, denominato progetto FUSS (Free Upgrade for a digitally Sustainable School) coinvolge attualmente tutte le scuole pubbliche primarie e secondarie in lingua italiana della Provincia Autonoma di Bolzano, oltre ad alcune scuole dell'infanzia (<https://fuss.bz.it/>).

È tuttavia importante sottolineare che l'adozione di software libero non è sufficiente per garantire l'indipendenza da attori privati: occorre anche offrire la possibilità di conservare ed elaborare i dati utilizzando infrastrutture che ricadono sotto il controllo pubblico, come ad esempio server e data center statali.

Diversi paesi europei già si stanno muovendo in questa direzione. La Francia si è impegnata, già dall'inizio della pandemia, per offrire alle scuole un insieme di servizi open source a licenza aperta, controllati dallo Stato e operanti su server statali: una piattaforma educativa, una piattaforma di teleconferenza e un servizio di cloud. Anche in Spagna e in Germania, dove le istituzioni scolastiche sono organizzate secondo una maggiore autonomia regionale, diverse regioni si sono orientate verso l'utilizzo di software libero (Zoja, 2022).

Anche in Italia esistono avanguardie virtuose in tal senso, come il Politecnico di Torino, che già da tempo stava lavorando allo sviluppo di software libero per la didattica e quindi si è trovato pronto quando è scoppiata la pandemia: tutte le attività in videoconferenza sono state realizzate su *BigBlueButton*, una piattaforma open source creata a scopo educativo, e tutti i dati sono stati raccolti ed elaborati all'interno di data center gestiti dal Politecnico stesso. Si tratta di un modello che potrebbe essere replicato, oltre che da altri atenei, anche a livello di sistema scolastico nazionale: un primo passo in questa direzione è stato mosso nel 2021, con la decisione del Parlamento di stabilire un fondo per un disegno di legge volto alla creazione di *Unire*, una nuova rete unica di interconnessione delle scuole italiane.

3.2 Software libero e pedagogia hacker: imparare a de-costruire e ricostruire le tecnologie

L'adozione di software libero nelle istituzioni educative è dunque una condizione necessaria per mantenere il controllo pubblico e democratico dell'infrastruttura digitale su cui esse si reggono. Tuttavia, se si adotta la prospettiva di un'educazione emancipatrice (Freire, 2002), un tale cambiamento risulta necessario ma non sufficiente. Se infatti si intende mettere tutti e tutte nelle condizioni di impadronirsi degli strumenti culturali necessari per vivere da soggetti attivi e consapevoli all'interno della società, non si può trascurare il fatto che nel mondo contemporaneo stanno assumendo un ruolo sempre più cruciale le capacità di comprendere, analizzare criticamente e contribuire alla creazione di programmi informatici.

Questo significa, in primo luogo, promuovere lo sviluppo di quello che può essere definito un approccio *hacker* alle tecnologie. Occorre chiarire che il termine *hacker* non è qui da intendersi nell'accezione che spesso gli viene attribuita nel discorso mediatico, ovvero quella di individui malintenzionati che utiliz-

zano le proprie competenze informatiche per compiere attività criminali, ma in un senso più ampio e comprensivo della pluralità e della varietà di espressioni in cui l'attitudine *hacker* si manifesta: si tratta di un approccio verso le tecnologie mosso da curiosità e intraprendenza, voglia di comprenderne il funzionamento "mettendoci le mani sopra", smontando e rimontando, modificando e migliorando. La «pedagogia hacker» può quindi essere definita come un approccio pedagogico volto a promuovere curiosità e spirito critico nei confronti delle tecnologie, che si realizza primariamente in maniera esperienziale, cimentandosi nella comprensione e nella creazione di oggetti tecnologici specifici (siano essi di natura immateriale come il software, oppure materiale come l'hardware o altri tipi di artefatti tecnologici) (Ippolita, 2017). Quando si tratta della scrittura di software, è evidente che un approccio pedagogico *hacker* può sorgere solo in un contesto animato dalla filosofia del software libero, cioè fondato sull'apertura e la trasparenza del codice, mentre appare impraticabile quando si ha a che fare con il software proprietario, che non consente agli esterni di capire il funzionamento di un programma, ma solo di usufruirne come utenti.

Una pedagogia *hacker* non può però limitarsi a fornire competenze tecniche relative al funzionamento di una determinata tecnologia. Quando ci si occupa di educare al rapporto con le tecnologie, occorre infatti confrontarsi con una constatazione fondamentale, che molte volte e in varie forme riaffiora nella riflessione filosofica occidentale e non: sapere è potere, padroneggiare una tecnologia significa trovarsi in una posizione di potere. Appare dunque quantomai importante che chi impara ad utilizzare o creare una determinata tecnologia si cimenti anche in una riflessione profonda sulla concezione del potere e su come rapportarsi ad esso. Tale riflessione, a sua volta, non può rimanere puramente teorica, ma risulta inscindibile dall'esperienza vissuta delle concrete relazioni di potere in cui ci si trova, a partire ovviamente da quelle che si danno nei luoghi educativi, nella consapevolezza che queste ultime costituiscono una delle fondamentali palestre in cui le nuove generazioni imparano a vivere in comunità e fanno esperienza di vari tipi di relazione di potere, tra pari e con gli adulti.

Si tratta quindi di promuovere, nei contesti educativi, uno sguardo capace di analizzare i sistemi socio-tecnici in cui ci troviamo immersi: infatti, come ha messo in luce, tra gli altri, Bruno Latour, pressoché ogni aspetto delle nostre società è costituito dall'intreccio di elementi tecnico-scientifici e di dinamiche sociali e politiche, e tentare di comprendere separatamente una sola di queste due polarità ci condurrebbe ad una comprensione fortemente parziale e fuorviante della realtà in cui ci troviamo (Latour, 2018). È infatti evidente che le tecnologie occupano un ruolo centrale nelle nostre società, e che la loro gestione non è una questione meramente tecnica, bensì chiama sempre in causa interessi e scelte politiche: quali tecnologie vengono sviluppate, da chi e a beneficio di chi? Perché ne vengono incentivate alcune e non altre? Chiedersi che tipo di tecnologie scegliamo di costruire implica una riflessione su quale tipo di società intendiamo edificare.

In questo senso, la filosofia che anima il movimento Free Software offre una prospettiva valoriale abbastanza definita: propone di preferire la creazione collaborativa e la libera condivisione dei prodotti della propria attività, rispetto alla logica della chiusura proprietaria che incentiva le persone a considerarsi proprietarie dei prodotti del proprio lavoro e a decidere se e come dividerlo in base al tornaconto che ne possono ricevere. Educare le nuove generazioni ad apprendere e a lavorare in modo cooperativo nella creazione di software può quindi avere un valore che non riguarda solamente l'ambito della programmazione, ma potrebbe consentire loro di interiorizzare un habitus collaborativo che può poi riverberarsi in qualsiasi ambito della vita: ciò risulterebbe in linea con la prospettiva secondo cui una delle modalità più feconde per promuovere l'educazione etico-sociale consiste nel proporre pratiche che siano di per sé strutturate sulla base dei principi etico-sociali che si intende promuovere (Baldacci, 2020).

La valenza educativa, oltre che etico-politica, dell'utilizzo del software libero a scuola è esplicitata con chiarezza nella descrizione degli obiettivi del già citato progetto FUSS:

La scelta di usare software libero nella scuola è anzitutto, al di là delle ragioni economiche o tecniche, una scelta etica e politica. È cioè la scelta di rifarsi, nell'insegnamento, ai valori della libertà e della condivisione del sapere, e non solo quella di usare software efficiente, stabile e sicuro. [...] L'utilizzo del software libero rende possibile la partecipazione diretta al suo sviluppo da parte di studenti ed insegnanti, non solo come scrittura di codice, ma soprattutto in termini di suggerimenti sul funzionamento, produzione di documentazione, traduzioni, realizzazioni di contenuti, ecc. Tutto ciò in un modello di scuola vista come una comunità in cui tutte le sue componenti partecipano attivamente al processo di costruzione della conoscenza. Obiettivo a medio-lungo termine di questo progetto è favorire metodologie di didattica collaborativa, attraverso il coinvolgimento attivo sia dei docenti che degli studenti nello sviluppo del progetto stesso. (<https://fuss.bz.it/>).

In ogni caso, anche al di là degli specifici posizionamenti valoriali, è cruciale che il nostro sistema educativo sia capace di promuovere di pari passo le competenze tecnico-scientifiche e le sensibilità umanistiche e sociopolitiche: se infatti venisse a mancare la seconda componente, ci limiteremmo a formare buoni tecnici che si troveranno ad essere ingranaggi di un sistema di cui faticano a comprendere gli obiettivi e le implicazioni etiche e socio-politiche. Al contrario, averne consapevolezza di tali implicazioni è il primo passo necessario per agire come soggetti e non solo come ingranaggi. Ad esempio, è fondamentale che chi scrive programmi informatici si interroghi sul contesto nel quale si inserisce il proprio lavoro: rimanendo sul tema delle licenze, è possibile chiedersi se si stia contribuendo al funzionamento di un'azienda basata sul modello software proprietario oppure ad un progetto collaborativo e, nel caso si tratti di un progetto definito collaborativo, indagare se esso rientri nella generica definizione di Open Source oppure

operi con licenze che garantiscono il rispetto delle quattro libertà del Free Software.

Occorre tuttavia sottolineare che la promozione di un approccio hacker non può essere limitata ad una minoranza della popolazione, considerata più predisposta nei confronti delle discipline tecnico-scientifiche: questo significherebbe infatti arrendersi all'idea che la maggioranza della popolazione non possa che limitarsi ad usufruire delle tecnologie in veste di utenti scarsamente consapevoli del loro funzionamento e delle implicazioni socio-politiche ad esse connesse, confinando così questa maggioranza ad una seria condizione di subalternità. Perciò senza voler negare l'esistenza di capacità, propensione o interesse nei confronti delle tecnologie, occorre ribadire la possibilità e la necessità di promuovere in tutti i cittadini e le cittadine lo sviluppo di competenze tecniche di base, e parimenti di un'attitudine riflessiva nei confronti delle tecnologie.

4. Conclusioni

Nella prima parte dell'articolo è stata ripercorsa la storia del software libero e del software proprietario: si è visto come originariamente la creazione di software si sia sviluppata in un contesto di libera condivisione, e solo negli anni Ottanta i codici informatici siano stati sottoposti a limitazioni proprietarie, basate sul concetto di diritto d'autore. Il movimento del Free Software è quindi sorto nel 1983 in contrapposizione a questa tendenza alla privatizzazione del codice, proponendo un modello alternativo, ovvero lo sviluppo di una comunità di programmatori e programmatrici impegnate a contribuire alla scrittura collettiva di software, i quali vengono poi condivisi utilizzando licenze che garantiscono il rispetto delle quattro libertà fondamentali del software libero.

Si è poi visto come all'interno del movimento del software libero sia nata, alla fine degli anni Novanta, la variante dell'Open Source, che riprende le modalità collaborative del movimento orientandole però verso obiettivi più marcatamente commerciali. È stato tuttavia evidenziato come un elemento di problematicità dell'Open Source consista nel fatto che al suo interno sono permesse anche licenze che consentono di applicare forme di chiusura proprietaria, tramite le quali un'azienda può appropriarsi di un codice scritto collettivamente e gratuitamente da volontari per apportarvi miglioramenti che diventano però proprietà privata e non più accessibile.

Sulla base di questa retrospettiva storica, nella seconda parte dell'articolo è stato messo in luce come l'approccio del software libero (in particolare nella declinazione del Free Software, piuttosto che dell'Open Source) appaia maggiormente in linea con l'orizzonte di un'educazione democratica ed emancipatrice, mentre l'utilizzo del software proprietario rischi invece di accentuare la tendenza verso un'evoluzione in senso neoliberalista delle istituzioni educative. Infatti, a partire dalla constatazione che le tecnologie digitali rappresentano oggi una infrastruttura strategica, e dunque che chi ne detiene il controllo si trova in una posizione di potere estremamente rilevante, la questione che si pone è: a chi vogliamo affidare il controllo dei dati e delle piattaforme digitali su cui si basa

il funzionamento delle istituzioni educative pubbliche?

Attualmente la maggioranza delle scuole italiane sta delegando la gestione di buona parte dei propri dati a grandi multinazionali private, tra le quali si registra una netta predominanza di Google, la cui piattaforma educativa viene utilizzata dall'86% delle scuole italiane. Questa tendenza ad appaltare ad aziende multinazionali l'infrastruttura digitale degli istituti pubblici presenta due fondamentali elementi di criticità. In primo luogo, significa affidare una notevole quantità di dati sensibili e preziosi a soggetti privati che, essendo aziende quotate in borsa e soggette agli imperativi del mercato, hanno come obiettivo primario la ricerca del profitto, non certo la promozione dell'educazione pubblica. Questo accade, oltretutto, senza che ci sia la possibilità di verificare con certezza in che modi questi dati potrebbero essere utilizzati. In secondo luogo, tali aziende tendono alla fidelizzazione, ovvero a penetrare nelle organizzazioni educative offrendo inizialmente servizi gratuiti, fino al punto di sviluppare una dipendenza da cui diventa poi molto difficile svincolarsi: una volta affermatesi come una componente chiave dell'infrastruttura su cui si basano le istituzioni educative, è possibile immaginare che queste aziende intendano esercitare un'influenza più o meno esplicita anche sui contenuti e/o sulle modalità che caratterizzano le attività educative.

Quello appena delineato non rappresenta tuttavia un destino ineluttabile. Una possibile strategia alternativa consiste infatti nell'orientare le istituzioni educative verso il software libero, il che potrebbe essere fatto seguendo due direttrici.

Per prima cosa, occorre promuovere lo sviluppo di software libero e la sua adozione da parte delle istituzioni educative. Esiste già una legge che prescrive alle pubbliche amministrazioni di preferire l'adozione di software libero rispetto a quello proprietario, ma è inapplicata e costantemente disattesa. Lo sforzo iniziale necessario a trasferire i propri sistemi su piattaforme di software libero non può essere una responsabilità che viene scaricata sulle singole scuole, le quali spesso mancano della consapevolezza e delle risorse per attuare tale cambiamento da sole, è piuttosto necessaria una decisione politica a livello nazionale, simile a quella intrapresa da altri paesi europei, che incentivi l'adozione di software libero e metta a disposizione piattaforme educative pubbliche e sottoposte al controllo democratico.

In secondo luogo, appare altresì auspicabile favorire all'interno delle istituzioni educative un approccio allo studio e alla creazione di programmi informatici animato dalla filosofia di collaborazione e condivisione che sta alla base del software libero e della pedagogia hacker. Ciò significa, da un lato, promuovere un atteggiamento di curiosità e sperimentazione, che mira a comprendere le tecnologie "smontandole e rimontandole"; dall'altro, stimolare la riflessione sul nesso che lega tecnologie e potere: infatti, se è vero che padroneggiare una tecnologia significa trovarsi in una posizione di potere, risulta cruciale che l'acquisizione di competenze tecniche proceda di pari passo con lo sviluppo di un'attitudine riflessiva e critica rispetto alle dimensioni etiche, sociali e politiche che sono inscindibilmente connesse alle tecnologie.

Riferimenti bibliografici

- Baldacci, M. (2020). *Un curriculum di educazione etico-sociale: Proposte per una scuola democratica*. Roma: Carocci.
- Barley, S., & O'Mahony, S. (2002). *The emergence of a new commercial actor: Community managed software projects*. Stanford: Stanford University Press.
- Bratton, B. H. (2015). *The stack: on software and sovereignty*. Cambridge, Massachusetts: MIT Press.
- Burrell, J., & Fourcade, M. (2021). The Society of Algorithms. *Annual Review of Sociology*, 47, 213–37. <https://doi.org/10.1146/annurev-soc-090820-020800>
- Cappello, G. (2010). Analisi critica vs. produzione creativa: le nuove sfide della media education nell'era digitale. *Form@re*, 70(10), 37 – 44. <https://doi.org/10.13128/formare-12535>
- Formenti, C. (2011). *Felici e sfruttati: capitalismo digitale ed eclissi del lavoro*. Milano: EGEA.
- Free Software Foundation. (2022a). What is Free Software?. *gnu.org*. Retrieved January 15, 2023, from <https://www.gnu.org/philosophy/free-sw.html>
- Free Software Foundation. (2022b). What is Copyleft?. *gnu.org*. Retrieved January 15, 2023, from <https://www.gnu.org/licenses/copyleft.en.html>
- Freire, P. (2002). *Pedagogia degli oppressi*. Torino: EGA.
- Hillman, T., Bergviken Rensfeldt, A., & Ivarsson, J. (2020). Brave new platforms: a possible platform future for highly decentralised schooling. *Learning, Media and Technology*, 45(1), 7 – 16. <https://doi.org/10.1080/17439884.2020.1683748>
- Ippolita. (2005) *Open non è free: comunità digitali tra etica hacker e mercato globale*. Milano: Elèuthera.
- Ippolita. (2017). *Tecnologie del dominio: lessico minimo di autodifesa digitale*. Milano: Meltemi.
- Jardin, X. (2005). Bill Gates: Free Culture advocates = Com-mies. *Boing Boing*. Retrieved January 15, 2023, from <https://boingboing.net/2005/01/05/bill-gates-free-cult.html>
- Latour, B. (2018). *Non siamo mai stati moderni*. Milano: Elèuthera.
- Lazare, M. (2021). A peek at what's next for Google Classroom. *Blog.google*. Retrieved January 15, 2023, from <https://blog.google/outreach-initiatives/education/classroom-roadmap/>
- Margiotta, U. (2019). Responsabilità pedagogica e ricerca educativa: intelligenza collaborativa, formazione dei talenti e tecnologie dell'artificiale. *Formazione e insegnamento*, 1, 13-18. https://doi.org/10.7346/fei-XVII-01-19_01
- Meo, A. R. (2022). Scuola e università: perché preferire il software libero. *Gli Asini*, 99. Retrieved January 15, 2023, from <https://gliasini rivista.org/scuola-e-universita-perche-preferire-il-software-libero/>
- Panciroli, C., Rivoltella P. C., Gabbriellini M., & Zawacki Richter O. (2020). Intelligenza artificiale e educazione: nuove prospettive di ricerca. *Form@re*, 20(3), 1 – 12. <http://dx.doi.org/10.13128/form-10210>
- Poell, T., & van Dijck, J. (2018). *Social Media Platforms and Education*. In J. Burgess, A. E. Marwick, & T. Poell (Eds.), *The SAGE handbook of social media* (pp. 546 – 561). Sage.
- Raymond, E. S. (1998). Goodbye, "free software"; hello, "open source". *CatB*. Retrieved January 15, 2023, from <http://www.catb.org/~esr/open-source.html>
- Raymond, E. S. (1999). *The cathedral & the bazaar: musings on Linux and open source by an accidental revolutionary*. Cambridge, MA: O'Reilly Media.
- Rossi, B., Russo, B., & Succi, G. (2009). Migrazione a Software Open Source: Caso di Studio della Provincia Autonoma di Bolzano-Bozen. In L. Bertini, T. Federici, A. Montemaggio, & P. Spagnoletti (Eds.), *Open Source nella Pubblica Amministrazione* (pp. 65 – 70). Raleigh: Lulu Enterprises.
- Seligmann, S. (2021). The Computer Software Copyright Act of 1980. *The Chicago Council on Science and Techno-*

- logy. Retrieved March 11, 2023, from <https://www.c-2st.org/the-computer-software-copyright-act-of-1980/>
- Selwyn, N. (2011). *Education and technology: key issues and debates*. London & New York: Continuum.
- Selwyn, N., Hillman, T., Eynon, R., Ferreira, G., Knox, J., Macgilchrist, F., & Sancho-Gil, J. M. (2020). What's next for Ed-Tech? Critical hopes and concerns for the 2020s. *Learning, Media and Technology*, 45(1), 1–6. <https://doi.org/10.1080/17439884.2020.1694945>
- Srnicek, N. (2017). *Platform capitalism*. Cambridge, UK: Polity.
- Stallman, R. (2007). Why Open Source misses the point of Free Software. *gnu.org*. Retrieved January 15, 2023, from <https://www.gnu.org/philosophy/open-source-misses-the-point.html>
- Williams, S. (2002). *Free as in Freedom: Richard Stallman's Crusade for Free Software*. Sebastopol, CA: O'Reilly Media.
- Zoja, S. (2022). Così il controllo sulla didattica rafforza lo strapotere delle multinazionali. *Altreconomia*, 246. Retrieved January 15, 2023, from <https://altreconomia.it/la-scuola-italiana-al-mercato-dei-dati/>
- Zuboff, S. (2019). *The age of surveillance capitalism: the fight for a human future at the new frontier of power*. New York: PublicAffairs.