

Ambienti per la didattica dell'Informatica

Indice

Introduzione	2
1. Analisi delle problematiche dei Laboratori	3
1.1. Problematiche dei Laboratori d'Informatica	3
1.2. Problematiche del Future Lab	5
2. Analisi dei vincoli e degli indirizzi per le soluzioni ICT nella scuola	6
2.1. Il quadro normativo sull'acquisto delle soluzioni ICT	6
2.2. Orientamenti in favore del Software Libero	8
2.3. Software compatibile con il CAD	13
3. Specifica dei requisiti	14
3.1. Requisiti dell'ambiente laboratoriale di Informatica	14
3.2. Requisiti sulla disponibilità di applicazioni	15
4. Ipotesi architetturali	18
4.1. Separazione delle sotto-reti	18
4.2. FUSS Server con Veyon ma dual boot	19
4.3. Macchine virtualizzate	21
Riferimenti	22

Il laboratorio

- I Biologi hanno microscopi e vetrini
- I Chimici hanno provette e Bunsen
- I Fisici hanno acceleratori di particelle
- Noi abbiamo **elaboratori e reti**

[6 p. 19]

— Renzo Davoli, ...tutto quello che avreste voluto sapere sui laboratori ma non avete mai osato chiedere...

Introduzione

Il presente documento, a cura del Dipartimento di Informatica, vuole a fissare i problemi rilevati dagli insegnanti afferenti al Dipartimento e vuole, soprattutto, fornire uno spunto per la Dirigenza, il Team dell'Innovazione, il progettista della rete dati, il Responsabile della Protezione dei Dati e i responsabili del Test Center ICDL affinché l'innovazione dei Laboratori e del Future Lab proceda in modo da risolvere le problematiche.

Il testo illustra in modo analitico le problematiche che si verificano negli ambienti laboratoriali per la didattica dell'Informatica ([par. 1](#)), indicando quelle imputabili alla configurazione delle reti e dei computer dei laboratori ([par. 1.1](#)) e quelle che dipendono dalla gestione degli spazi nel Future Lab ([par. 1.2](#)).

La risoluzione di alcune delle criticità comporta una necessaria riconfigurazione delle reti, senza oneri per l'acquisto di nuovi dispositivi, e la sostituzione del software. Quest'ultima deve avvenire nel rispetto dei vincoli e delle linee di indirizzo sull'approvvigionamento del software da parte di una Pubblica Amministrazione, dei quali viene fornita una breve ricognizione ([par. 2](#)). Il vincolo normativo sul software ([par. 2.1](#)) è di facile esposizione: il CAD [7] richiede il riuso del software sviluppato dalla PA o in alternativa del software libero. Questo documento include una breve rassegna di contributi sul software per la didattica presentati da illustri membri della comunità scientifica e delle reti scolastiche europee ([par. 2.2](#)). Per permettere la comparazione delle soluzioni, vengono presentati alcuni sistemi software ([par. 2.3](#)) per la gestione della rete didattica nel rispetto del CAD e del GDPR [4].

I requisiti per l'uso dei dispositivi sono brevemente formalizzati nel testo ([par. 3](#)), divisi in requisiti sulla gestione delle postazioni ([par. 3.1](#)) e di software installati ([par. 3.2](#)).

Il documento si conclude con delle ipotesi progettuali di massima ([par. 4](#)) che sembrano soddisfare i requisiti individuati. A livello della rete dati viene suggerita una possibile suddivisione della stessa in sottoreti ([par. 4.1](#)). Per quanto concerne la gestione del software sui calcolatori, sono presentate due soluzioni complementari, una basata sull'adozione di una soluzione completa per la gestione di una rete didattica ([par. 4.2](#)) e l'altra che prevede l'installazione di macchine virtuali sul server ([par. 4.3](#)) che sembrano essere adottate da un numero crescente di scuole.

1. Analisi delle problematiche dei Laboratori

1.1. Problematiche dei Laboratori d'Informatica

L'attività didattica in laboratorio richiede un'adeguata predisposizione dei PC e dei collegamenti che non sempre si riscontra nelle attuali condizioni. Alcune situazioni problematiche sono indicate nel prosieguo.

Uso delle tecnologie per fini non didattici

gli studenti possono usare i PC del laboratorio per fini non didattici, anche commettendo illeciti, senza che l'insegnante riesca a rilevare l'uso improprio dello strumento informatico. Esempi di usi impropri sono l'uso di video-giochi, già installati sul PC oppure online, la visualizzazione di video, social network, messaggistica istantanea o pagine web non inerenti l'attività didattica, lo svolgimento di compiti di altre materie al PC... La mancanza di strumenti di monitoraggio dà allo studente un senso di impunità in quanto l'insegnante non riesce a rilevare in modo esatto l'uso improprio e non può sanzionare il comportamento che non può osservare.

Visualizzazione difficoltosa dello schermo dell'insegnante

l'insegnante spesso usa il PC per illustrare contenuti e metodi propri della disciplina, come le tecniche di programmazione, e frequentemente le schermate possono risultare di difficile lettura da lontano. Gli studenti che non riescono a visualizzare facilmente la schermata possono perdere facilmente l'attenzione e il coinvolgimento attivo.

Mancanza di proiezione dello schermo dello studente

lo studente non può condividere il proprio schermo con l'insegnante e/o gli altri studenti rendono più complesse le dinamiche sociali dell'apprendimento e il coinvolgimento attivo.

Disponibilità del software e in molteplici versioni

nella configurazione attuale alle volte il software richiesto per l'attività didattica non risulta installato in alcuni PC, oppure lo è ma non è configurato correttamente, oppure è installato in versioni diverse e incompatibili con quella che si desidera usare per la didattica. Ciò crea difficoltà e frustrazione da parte degli studenti e dell'insegnante, riduce l'attenzione e il coinvolgimento.

Installazioni e aggiornamenti complessi e time-consuming

l'installazione o l'aggiornamento di un software richiede l'intervento dell'assistente tecnico e richiede tempo in quanto le stesse operazioni vengono replicate in ogni postazione. La mancanza di un gestore di pacchetti rende rischiosa l'installazione del software sia per l'eventuale presenza di malware che per il rischio di dipendenze insoddisfatte. La situazione induce l'insegnante a pensare che sia troppo oneroso installare o aggiornare un nuovo software per cui conveniente mantenere in uso il software obsoleto piuttosto che quello migliore per la didattica.

Gestione errata della persistenza dei dati degli utenti

in casi sporadici, con le versioni di Microsoft Windows ® installate nei laboratori, succede che

all'autenticazione segua la creazione di una sessione temporanea. In tale situazione alla disconnessione dell'utente tutti i dati salvati vengono cancellati senza possibilità di ripristino. La situazione non è controllabile e può causare frustrazione e perdita di dati rilevanti per studenti e insegnanti.

Impossibilità di sperimentare il software di rete

attualmente i laboratori sono collegati alla stessa rete in cui sono presenti altri servizi, anche critici, come quelli di supporto alle Segreterie. Vi sono problemi nel perimetro fisico dei dati che non consentono una sperimentazione sicura. Non si dispone di un diagramma che illustri l'infrastruttura, i servizi e i dispositivi. Questo non permette la sperimentazione sicura del software di rete. Le reti, invece, sono da considerarsi parte integrante del laboratorio di informatica [6 p. 19] e sono oggetto di studio in tutti gli indirizzi dell'Istituto che hanno l'insegnamento di Informatica.

Impossibilità di controllo della rete del Laboratorio

molte competizioni, tra cui i *Campionati italiani e internazionali di Informatica* e le *Olimpiadi di Informatica a Squadre*, richiedono di disattivare l'accesso ad Internet ad eccezione della pagina della gara. Anche per alcune verifiche potrebbe essere fatto divieto di accedere a risorse esterne al proprio PC oltre a quelle del servizio di erogazione della prova. Allo stato attuale il docente non ha nessun controllo sul traffico di rete.

Accesso illimitato a social network, sistemi di messaggistica e siti di streaming

attualmente gli studenti hanno accesso illimitato ai social network, alle piattaforme di messaggistica e a quelle di streaming; queste rappresentano potenziali distrazioni e generano la possibilità che gli studenti commettano, anche involontariamente, degli illeciti.

Impossibilità di blocco delle pubblicità

allo stato attuale le pubblicità non sono filtrate. Questo oltre a creare distrazione può sfociare in casi in cui si presentano contenuti non adatti all'ambito scolastico.

Impossibilità di comunicazione tra notebook connessi tramite WiFi e PC connessi tramite cavo

nei laboratori non è presente nessun access point e la WLAN è in una rete privata separata da quella LAN. Ciò condiziona gli insegnanti a raccogliere le prove di uno studente che opera con il suo dispositivo a scambiare i file in modo, insicuro, o usando dispositivi di storage (pericolo di malware) o tramite l'uso di piattaforme esterne di cui non ha il controllo.

Uso di soluzioni cloud esterne

allo stato attuale molti file sono gestiti tramite piattaforme cloud che non memorizzano i dati sul territorio nazionale e che rilasciano i servizi per la scuola allo scopo di creare futuri clienti. Così facendo non si hanno né la sovranità dei dati, ossia il controllo sulla crittografia e sull'accesso dei dati, né la sovranità operativa, ossia la visibilità e il controllo sulle operazioni del fornitore, e neppure la sovranità del software, ossia poter eseguire programmi senza dipendere dal software di un fornitore.

Uso esclusivo del sistema operativo a sorgente chiuso Windows ®

sulle postazioni del laboratorio è installato il solo sistema operativo Windows ®. Tale sistema, sebbene sia molto diffuso e permetta di far girare molto del software disponibile, è chiuso e non

permette lo studio del Sistema Operativo stesso. Molte applicazioni, tra cui alcune per il monitoraggio della rete, richiedono installazioni complesse. Questa situazione limita le possibilità di didattica sui Sistemi Operativi, le Reti di Elaboratori, delle rete, la Cyber Sicurezza.

Difficile gestione toolchain di sviluppo

Windows ® ha molti ambienti di sviluppo che sono distribuiti separatamente dal sistema operativo, richiedono un'installazione e non sono standard. Per avere una *toolchain* per il *Linguaggio C++* (classi terze ITE SIA e Liceo Scientifico sia Scienze Applicate che opzione Informatica), per lo Sviluppo del Web (tutti gli indirizzi con approfondimenti diversi), una *shell* dei *Sistemi Operativi* (tutti gli indirizzi), gli strumenti standard di gestione e analisi delle *Reti di Calcolatori* (tutti gli indirizzi), gli strumenti per la crittografia e la critto-analisi relative alla *Cyber Sicurezza* (tutti gli indirizzi sebbene con approfondimenti diversi) è necessario installare software e driver di dispositivo. Queste operazioni richiedono privilegi da super-utente e risultano difficoltose. Tutti i sistemi operativi che derivano da Unix ®, invece, includono questi strumenti come parte integrante della distribuzione.

Separazione delle risorse di calcolo

Gli estensori del documento non conoscono tecniche, sotto Windows ®, per condividere le risorse di calcolo in modo da sperimentare alcuni algoritmi particolarmente onerosi e che richiedono l'utilizzo di più PC come se questi fossero un'unica potente workstation.

Allo stato attuale le uniche modalità di mitigazione di alcune delle problematiche sopra esposte sono l'attenta sorveglianza da parte dell'insegnante e i tempestivi aggiornamenti del software da parte degli assistenti tecnici.

1.2. Problematiche del Future Lab

Il Future Lab viene impiegato come aula disciplinare d'Informatica e ivi si riscontrano le stesse problematiche sono le stesse dei laboratori con l'aggiunta delle seguenti.

Mancanza di separazione dall'atrio dell'Aula Magna

il Future Lab non è separato dall'ingresso dell'Aula Magna, luogo in cui frequentemente sono ospitati incontri. Quando ciò avviene si verificano distrazioni e confusione con perdita di attenzione da parte degli studenti.

Scarsa disponibilità di prese elettriche

la rete elettrica non dispone di punti luce ad eccezione di alcune colonne e delle postazioni fisse.

Scarsa disponibilità di punti prese di rete

la rete dati dispone di prese ethernet solo nelle postazioni fisse.

Indisponibilità di PC/notebook dedicati al Future Lab

non vi sono PC o notebook predisposti per questo ambiente e quindi si deve fidare nei dispositivi degli studenti.

2. Analisi dei vincoli e degli indirizzi per le soluzioni ICT nella scuola

2.1. Il quadro normativo sull'acquisto delle soluzioni ICT

Per la valutazione e la proposta di soluzioni software consideriamo il quadro normativo fissato dal Codice dell'Amministrazione Digitale [7], in particolare gli artt. 68 e 69.

Art. 68. Analisi comparativa delle soluzioni

1. Le pubbliche amministrazioni acquisiscono programmi informatici o parti di essi nel rispetto dei principi di economicità e di efficienza, tutela degli investimenti, riuso e neutralità tecnologica, a seguito di una valutazione comparativa di tipo tecnico ed economico tra le seguenti soluzioni disponibili sul mercato:

- a. software sviluppato per conto della pubblica amministrazione;
- b. riutilizzo di software o parti di esso sviluppati per conto della pubblica amministrazione;
- c. software libero o a codice sorgente aperto;
- d. software fruibile in modalità cloud computing;
- e. software di tipo proprietario mediante ricorso a licenza d'uso;
- f. software combinazione delle precedenti soluzioni.

1-bis. A tal fine, le pubbliche amministrazioni prima di procedere all'acquisto, secondo le procedure di cui al codice di cui al decreto legislativo n. 50 del 2016, effettuano una valutazione comparativa delle diverse soluzioni disponibili sulla base dei seguenti criteri:

- a. costo complessivo del programma o soluzione quale costo di acquisto, di implementazione, di mantenimento e supporto;
- b. livello di utilizzo di formati di dati e di interfacce di tipo aperto nonché di standard in grado di assicurare l'interoperabilità e la cooperazione applicativa tra i diversi sistemi informatici della pubblica amministrazione;
- c. garanzie del fornitore in materia di livelli di sicurezza, conformità alla normativa in materia di protezione dei dati personali, livelli di servizio

tenuto conto della tipologia di software acquisito.

1-ter. Ove dalla valutazione comparativa di tipo tecnico ed economico, secondo i criteri di cui al comma 1-bis, risulti motivatamente l'impossibilità di accedere a soluzioni già disponibili all'interno della pubblica amministrazione, o a software liberi o a codici sorgente aperto, adeguati alle esigenze da soddisfare, è consentita l'acquisizione di programmi informatici di tipo proprietario mediante ricorso a licenza d'uso. La valutazione di cui al presente comma è effettuata secondo le modalità e i criteri definiti dall'AgID.

— Codice per l'Amministrazione Digitale, Capo VI. SVILUPPO

Art. 69. Riutilizzo delle soluzioni e standard aperti

1. Le pubbliche amministrazioni che siano titolari di soluzioni e programmi informatici realizzati su specifiche indicazioni del committente pubblico, hanno l'obbligo di rendere disponibile il relativo codice sorgente, completo della documentazione e rilasciato in repertorio pubblico sotto licenza aperta, in uso gratuito ad altre pubbliche amministrazioni o ai soggetti giuridici che intendano adattarli alle proprie esigenze, salvo motivate ragioni di ordine e sicurezza pubblica, difesa nazionale e consultazioni elettorali.
2. Al fine di favorire il riutilizzo dei programmi informatici di proprietà delle pubbliche amministrazioni, ai sensi del comma 1, nei capitolati o nelle specifiche di progetto è previsto, salvo che ciò risulti eccessivamente oneroso per comprovate ragioni di carattere tecnico-economico, che l'amministrazione committente sia sempre titolare di tutti i diritti sui programmi e i servizi delle tecnologie dell'informazione e della comunicazione, appositamente sviluppati per essa.

2-bis. Al medesimo fine di cui al comma 2, il codice sorgente, la documentazione e la relativa descrizione tecnico funzionale di tutte le soluzioni informatiche di cui al comma 1 sono pubblicati attraverso una o più piattaforme individuate dall'AgID con proprie Linee guida.

— Codice per l'Amministrazione Digitale, Capo VI. SVILUPPO

Per l'applicazione si rimanda a [\[1\]](#).

Si tengono in considerazione, in particolare:

comma 1 lettera c

software libero o a codice sorgente aperto

comma 1-bis lettera a

costo complessivo del programma o soluzione

comma 1-bis lettera b

...utilizzo di formati di dati e di interfacce di tipo aperto...

Un'analisi estesa della normativa nel contesto delle scuole, sul perché è volutamente e dichiaratamente ignorata, e sulle possibili conseguenze per il decisore si trova in [\[16\]](#) ed è esposta nell'[estratto "da Scuola e università: perché preferire il Software Libero"](#).

2.2. Orientamenti in favore del Software Libero

Nella didattica dell'informatica si avverte una pressione che spinge le istituzioni scolastiche verso scelte opposte a quelle formativo ed educativo [\[17, 18, 20, 21\]](#):

- l'addestramento all'uso di tecnologie, hardware e/o software, proprietarie;
- l'uso di servizi come surrogato del software;
- la cessione dei dati alle grandi multinazionali che operano nel settore ICT.

Il messaggio commerciale raggiunge gli insegnanti, i dirigenti e, più in generale, il decisore pubblico in modo più forte e capillare del messaggio scientifico. La scuola risponde già una grande quantità di proposte dalle aziende del *settore education* ma con scarsa considerazione della normativa e degli orientamenti provenienti dalle organizzazioni con fini sociali e scientifici.

La Free Software Foundation argomenta che passaggio al Software Libero garantisce all'utente, insegnante o studente, quattro libertà in più [\[19\]](#):

0. Libertà di eseguire il programma, per qualsiasi scopo.
1. Libertà di studiare come funziona il programma ed adattarlo alle proprie necessità.
2. Libertà di ridistribuire copie in modo da aiutare il prossimo.
3. Libertà di migliorare il programma e distribuirne pubblicamente i miglioramenti, in modo tale che tutta la comunità ne tragga beneficio.

L'accesso al codice sorgente è un prerequisito per la seconda e la quarta libertà.

Per maggiori dettagli sul tema etico e sulla necessità di usare esclusivamente il Software Libero nelle scuole si rimanda agli articoli [\[11, 12, 13, 17\]](#).

Molti docenti universitari, principalmente quelli dei settori dell'Informatica, si sono fatti promotori del Software Libero nelle Università e nelle Scuole.

Davoli [\[5\]](#) offre molti spunti di riflessione sulla didattica dell'informatica in tutti i cicli scolastici indicando gli oggetti e gli strumenti di studio e per lo studio. Egli ritiene che la scuola deve usare soltanto Software Libero.

- Una scuola che non consenta di poter capire come funzionano gli strumenti che usa è la negazione di se stessa
- Una scuola senza sapere libero non istiga alla curiosità e spreca denaro pubblico
- La scuola mai deve essere parziale, e.g. i libri di testo non devono contenere messaggi pubblicitari
- La scuola deve pensare al futuro dei propri studenti (risparmi immediati, soluzioni "chiavi in mano", possono portare a costi di dipendenza futuri)
- La scuola deve sempre educare alla condivisione della conoscenza, deve essere un luogo di aggregazione di idee, dove si insegna il rispetto, l'aiuto vicendevole e l'impegno sociale.

[5 p. 23]

— Renzo Davoli, *L'informatica nella Scuola: strumenti e metodi*

Sempre Davoli [6] argomenta a favore dell'uso del Software Libero nelle università con motivazioni didattiche e filosofiche:

- stimola lo spirito critico; nulla è segreto o nascosto, ogni curiosità conoscitiva può essere soddisfatta
- dovere di imparzialità: il software è un mezzo, non un fine
- massima accessibilità e uguali possibilità a tutti gli studenti (non possiamo forzarli a pagare e/o piratare software)
- materiale di studio: non solo 5-minute exercises, hands-on su progetti software enormi
- perché usare Software Libero porta con sé un messaggio etico intrinseco

[6 p. 26]

— Renzo Davoli, ...tutto quello che avreste voluto sapere sui laboratori ma non avete mai osato chiedere...

Davoli non considera solo i motivi didattici ma sostiene che il Software Libero presenta è intrinsecamente migliore di quello proprietario in quanto presenta i seguenti vantaggi:

vantaggi di sicurezza

aumenta il numero di occhi che possono risolvere bug di sicurezza →
contrasta i pirati informatici

vantaggi qualitativi

tutti possono guardare il mio codice, devo fare bella figura!

vantaggi comunitari

l'accesso al codice favorisce la formazione di comunità interessate a migliorarlo

vantaggi per i power user

piego i miei strumenti al mio volere

vantaggi economici

minore TCO, indipendenza da possibili monopoli (il codice è [anche] mio e [se proprio devo] me lo gestisco io!)

[6 p. 25]

— Renzo Davoli, ...tutto quello che avreste voluto sapere sui laboratori ma non avete mai osato chiedere...

Meo concentra l'attenzione sul valore sociale [2, 3] del Software Libero e ne propone un'analisi storica sull'evoluzione della sua adozione in Italia [16]. Egli presenta gli aspetti normativi avendo presieduto la Commissione Stanca sul software Open Source (2003) da cui derivarono gli articoli 68 e 69 della legge 82/05 [7] e la seconda commissione sul software Open Source nella Pubblica Amministrazione (2007) istituita dal ministro Nicolais. Illustra con dovizia di particolari il quadro normativo italiano ed europeo, analizza l'aggravarsi dell'uso di software proprietari, severamente vietati dalla norma italiana e comunitaria, e ammonisce i decisori pubblici, che individua nei dirigenti scolastici, affinché non debbano motivare le loro scelte alla Corte dei Conti e al Garante della Protezione dei Dati Personali.

Molti studiosi ritengono che oggi l'informatica libera rappresenti l'unico strumento disponibile per il progresso tecnologico ed economico dei Paesi poveri e anche di un Paese come il nostro. Per questa ragione alcune norme di legge italiane e/o comunitarie impongono l'adozione di Software Libero in sostituzione del software proprietario. Sfortunatamente, quelle norme sono spesso disattese e, nell'area della scuola, sono volutamente e dichiaratamente ignorate (rimando al bellissimo articolo di Stefano Zoja La scuola italiana al mercato dei dati. Così il controllo sulla didattica rafforza lo strapotere delle multinazionali, "Altroeconomia").

Temo che la grande maggioranza dei dirigenti scolastici che hanno adottato software proprietario non sappia dove trovare una adeguata relazione comparativa che giustifichi la scelta e non disponga di dati importanti in un formato aperto. Di conseguenza quei dirigenti scolastici non potrebbero difendersi dalle accuse di "danno erariale" provenienti da un magistrato

della Corte dei Conti. Quasi tutti i dirigenti scolastici e molti rettori di università potrebbero essere colpiti da pesantissime sanzioni del Garante della protezione dei dati personali.

L'analisi della realtà induce a pensare che le accuse di danno erariale e le sanzioni del Gdpd siano attualmente molto improbabili. Tuttavia, per scelte politiche diverse la realtà potrebbe cambiare nell'arco di pochi giorni. A quel punto la giustificazione "così fanno tutti" non potrebbe essere giustificata perché i testi delle leggi vigenti sono molto chiari.

— Angelo Raffaele Meo, Scuola e università: perché preferire il Software Libero

Meo ha sollecitato varie volte il decisore pubblico nella scelta del Software Libero nella scuola, chiamando in causa, tra gli ultimi, la ministra Azzolina [15].

- i. Negli ultimi anni le tecnologie di intrusione malevola hanno registrato progressi clamorosi, per cui quello che è successo con Zoom potrebbe verificarsi in qualunque momento su qualunque altra tecnologia proprietaria. Per questa ragione noi riteniamo che nelle applicazioni della Pubblica Amministrazione che implicano il trattamento di dati personali, e in particolare nelle applicazioni per la scuola, si dovrebbe vietare per legge l'uso di prodotti dei quali non sia conosciuto il codice sorgente.
- ii. Ricordo anche che le scuole sono tenute a scegliere le soluzioni da acquisire solo dopo aver realizzato la valutazione comparativa prevista dall'art. 68 del D. Lgs. 82/2005, che impone di preferire Software Libero: sarebbe importante che il Ministero supportasse le scuole nell'adempiere a quest'obbligo.

Inoltre, Le chiedo il favore di adoperarsi per la promulgazione di una legge che:

- a. Proibisca l'uso di software proprietario nelle applicazioni della Pubblica Amministrazione che implicano il trattamento di dati personali
- b. Obblighi ad usare formati di file standard e aperti
- c. Consenta l'uso di infrastrutture IT in Cloud soltanto se queste sono nel controllo della pubblica amministrazione Italiana (Private Cloud).

[15]

— Angelo Raffaele Meo, Lettera alla ministra Azzolina dal professor Angelo Raffaele Meo...

Formiconi espone la sua visione sul Software Libero nelle scuole sul suo blog [8] e propone corsi MOOC sul Coding a scuola con Software Libero [9, 10].

A livello comunitario la [European Schoolnet](#) ha promosso un tutorial [14], tramite il portale School Education Gateway, che propone alle scuole vari software liberi.

- Per tutta la scuola

Per supportare la vita digitale degli studenti: [Mastodon](#) (per creare un social network scolastico locale e sicuro), [Etherpad](#) (per prendere appunti in collaborazione), [Mahara](#) (per creare portfolio online), [Zotero](#) (per raccogliere fonti e inserire bibliografie), [Klavaro](#) (touch typing, una tecnica che permette di scrivere più velocemente con una tastiera)

Nella biblioteca scolastica: [Koha](#) e [VuFind](#) (gestione della biblioteca)

Per l'apprendimento online: [Moodle](#) (per arricchire le lezioni in presenza), [Open edX](#) (per i corsi online)

Per l'amministrazione: [openSIS](#), [Open Admin](#) e [Gibbon](#) (gestione dati degli studenti)

Per gruppi e organi studenteschi e genitori: [Loomio](#) (discussione e processo decisionale)

Per il laboratorio informatico: [Veyon](#) (per vedere e gestire i computer)

Per test digitali: [TCExam](#) e [altri](#)

- Per gli insegnanti

Per storia: [TimelineJS](#) (per creare facilmente linee del tempo)

Per le lingue: [Anki](#) (carte con parole e immagini)

Per le arti: [Blender](#) (per creare film animati), [LenMus](#) (per studiare teoria musicale), [MuseScore](#) e [altri](#) (per creare e eseguire spartiti)

Per geografia: [Celestia](#) (esplorazione dello spazio), [Marble](#) (atlanti virtuali del globo)

Per le STEM: [FisicaLab](#) (per esplorare i problemi di fisica)

[14]

2.3. Software compatibile con il CAD

2.3.1. Progetti della PA per la gestione di laboratori e reti didattiche

Nel corso del tempo si la Pubblica Amministrazione ha sviluppato vari progetti per la gestione dei laboratori e delle reti didattiche.

2.3.1.1. So.Di.Linux

Il progetto So.Di.Linux è stato avviato nel 2003 con collaborazione tra il [Consiglio Nazionale delle Ricerche—Istituto Tecnologie Didattiche](#) (CNR-ITD) e l'[Associazione Italiana per il Calcolo Automatico](#) (AICA) finalizzata alla realizzazione e alla diffusione di una serie di strumenti didattici Open source nel mondo della scuola. Il prodotto realizzato è [So.Di.Linux](#), una distribuzione live che raccoglie al suo interno il software didattico libero sviluppato per Linux.

So.Di.Linux usa il software Veyon (vedi [par. 2.3.3.1](#)) per il monitoraggio delle postazioni.

2.3.1.2. FUSS

[FUSS](#) è una soluzione GNU/Linux completa per la gestione di una rete didattica. È anche un progetto di sostenibilità digitale che dal 2005 permette ad alunni e docenti di usare a casa gli stessi strumenti informatici installati a scuola, liberamente e senza alcun aggravio di costo. Il progetto è stato finanziato dal Fondo Sociale Europeo.

Il progetto pare essere il più attivo in Italia sia per rilasci, l'[ultimo](#) è del 28 luglio 2022, sia per numero di [persone coinvolte](#).

Anche FUSS usa il software [Veyon](#) per il monitoraggio delle postazioni.

2.3.2. Progetti della PA per la gestione degli Open Educational Resource (OER)

2.3.2.1. FARE

Per l'uso e la distribuzione di Open Educational Resource (OER) il Politecnico di Torino ha sviluppato il progetto [FARE \(Free Architecture for Remote Education\)](#). Sono fruibili circa 2000 materiali didattici in lingua italiana sviluppati secondo licenze Libere ed è possibile pubblicare il proprio materiale.

2.3.3. Software libero per la gestione di laboratori e reti didattiche

2.3.3.1. Veyon

[Veyon \(Virtual Eye On Networks\)](#) è un software libero e open source per monitorare e controllare i computer su più piattaforme. Veyon ha lo scopo di coadiuvare l'insegnamento in ambienti di apprendimento digitale. È disponibile come pacchetto nelle principali distribuzioni Linux e nei progetti della PA (vedi [par. 2.3.1.1](#) e [par. 2.3.1.2](#)).

2.3.4. Software libero per tutelare privacy nel cloud

2.3.4.1. Cryptomator

Le soluzioni di storage su cloud, anche se a detta dei fornitori del sono GDPR-compliant, risultano incompatibili con il Regolamento UE 2016/679 (GDPR) [4] perché i dati potrebbero essere conservati in Paesi nei non vige il Regolamento e perché, se non opportunamente cifrati, potrebbero essere visibili a persone non autorizzate. Il software Cryptomator fornisce uno strato di crittografia che rende meno rischioso l'uso servizi cloud e [compatibile con il GDPR](#).

2.3.5. Software con codice a sorgente aperto per le videoconferenze

2.3.5.1. Jitsi

[Jitsi](#) è una collezione di software per le video conferenze che posso essere eseguiti sul server controllati dall'Istituto. Tra gli strumenti vi sono [Jitsi Videobridge](#), il server per l'inoltro dei video e [Jitsi Meet](#), l'applicazione per visualizzare la videoconferenza.

3. Specifica dei requisiti

3.1. Requisiti dell'ambiente laboratoriale di Informatica

In considerazione delle problematiche descritte in [par. 1](#), l'insegnamento in ambienti di apprendimento digitale deve soddisfare le seguenti specifiche:

1. Il docente deve poter gestire in autonomia i PC del laboratorio
 - a. Il docente deve poter monitorare a distanza i PC del laboratorio
 - i. Il docente deve poter visualizzare la schermata dei PC del laboratorio dove sta insegnando
 - ii. Il docente deve poter salvare la schermata dei PC del laboratorio dove sta insegnando
 - b. Il docente deve poter controllare a distanza i PC del laboratorio
 - i. Il docente deve poter bloccare i dispositivi di input
 - ii. Il docente deve poter prelevare da remoto i file
 - iii. Il docente deve poter trasmettere lo schermo del proprio PC a quello degli studenti
 - c. Il docente e studenti devono poter condividere gli schermi
2. Il docente deve poter condividere applicazioni e servizi con gli studenti
 - a. Il docente deve poter gestire una macchina server, reale o virtuale, per creare e rimuovere utenti, applicazioni e servizi
 - b. La macchina server si collega al servizio LDAP oppure mantiene l'elenco degli utenti
 - c. Il docente deve poter installare pacchetti sulla macchina server
 - d. Il docente deve poter avviare e terminare servizi sulla macchina server

- e. I PC devono poter avviare le applicazioni sulla macchina server
 - i. I PC devono supportare il forwarding per le applicazioni grafiche
- 3. Il docente deve poter controllare gli accessi ad internet
 - a. Il docente deve poter impedire l'accesso ai file condivisi, ad eccezione dell'eventuale deposito della prova
 - b. Il docente deve poter impedire la ricerca di informazioni nella rete Internet su siti non esplicitamente consentiti (whitelist)
 - c. Il docente deve poter impedire lo scambio di messaggi con sistemi di chat (es: Google Chat), webmail (es: GMail) o altro (es: Whatsapp web, Google Drive)
- 4. Il docente deve poter controllare il DNS per rimuovere messaggi pubblicitari

3.2. Requisiti sulla disponibilità di applicazioni

La didattica dell'informatica in laboratorio, in relazione alle programmazioni dipartimentali dell'informatica, ai libri di testo adottati, alle Linee Guida e delle Indicazioni Nazionali, e alla [Proposta di Indicazioni Nazionali per l'insegnamento dell'Informatica nella Scuola](#) del CINI richiede l'installazione delle seguenti applicazioni:

- 1. Monitoraggio e controllo
 - a. [Veyon](#)
- 2. Applicazioni
 - a. Software per l'ufficio
 - i. [LibreOffice](#)
 - A. [LibreOffice Base](#)
 - B. [LibreOffice Calc](#)
 - C. [LibreOffice Charts](#)
 - D. [LibreOffice Draw](#)
 - E. [LibreOffice Impress](#)
 - F. [LibreOffice Math](#)
 - G. [LibreOffice Writer](#)
 - H. [LibreLogo](#)
 - ii. Software per la lettura dei PDF
 - b. Software per l'editoria
 - i. [TeX Live](#)
 - ii. [LyX](#)
 - iii. [Asciidoctor](#)
 - A. [Asciidoctor Diagram](#)
 - B. [Asciidoctor PDF](#)

- C. [Asciidoctor EPUB3](#)
 - D. [Asciidoctor reveal.js](#)
 - E. [asciidoctor-bibtex](#)
 - F. [Ruby rouge](#)
 - G. [Ruby coderay](#)
 - H. [Ruby pygments.rb](#)
- c. Stampa
 - i. [CUPS](#)
- d. Interpreti, compilatori, debugger, toolchain
 - i. [GCC, the GNU Compiler Collection](#)
 - ii. [Clang](#)
 - iii. [php](#)
 - iv. [Node.js](#) + [pnpm](#) + [TypeScript](#) + [NativeScript](#)
 - v. [Python 3](#)
 - vi. [R](#)
 - vii. [Ruby](#)
 - viii. [MiniZinc](#)
 - ix. [gdb](#)
 - x. [lldb](#)
 - xi. [rr](#)
 - xii. [Valgrind](#)
 - xiii. [ghidra](#)
 - xiv. [CMake](#)
 - xv. [GNU Make](#)
- e. Ambienti integrati di sviluppo
 - i. [VS Code](#)
 - ii. [Code::Blocks IDE](#)
 - iii. [vim](#)
 - iv. [Jupyter Lab](#)
 - v. [RStudio Desktop Open Source Edition](#)
 - vi. [Flowgorithm, sotto Linux](#)
 - vii. [LEGO® MINDSTORMS® Education EV3 Classroom](#) (solo su alcuni notebook)
 - viii. [Kojo](#)
 - ix. [Racket](#)
- f. Strumenti di versionamento

- i. [git](#)
- g. Accesso remoto
 - i. [ssh](#)
 - ii. [openVPN client](#)
- h. Audio e video
 - i. [Audacity](#)
 - ii. [FFMPEG](#)
 - iii. [Shotcut](#)
- i. Browser
 - i. [Firefox](#)
- j. CAD
 - i. [FreeCad](#)
- k. Grafica
 - i. [GNU Image Manipulation Program](#)
 - ii. [Inkscape](#)
 - iii. [GraphViz](#)
 - iv. [GnuPlot](#)
 - v. [imagemagick](#)
 - vi. [PlantUML](#)
 - vii. [Kroki](#)
- l. Calcolo scientifico e ingegneristico
 - i. [GNU Octave \(octave-forge\)](#)
 - ii. [Scilab](#)
 - iii. [GeoGebra](#)
 - iv. [R](#)
 - v. [miniconda](#)
- m. MindMapping
 - i. [FreePlane](#)
 - ii. [PlantUML](#)
- n. Software per l'analisi della rete
 - i. [WireShark](#)
 - ii. [ping](#)
 - iii. [ethtool](#)
 - iv. [dig](#), [dnsutils](#), [bind-utils](#)
 - v. [ip](#)

- vi. [netcat](#)
- vii. [nmap](#)
- o. Software per la configurazione del firewall
 - i. [gufw](#)
- 3. Servizi
 - a. Web server
 - i. [Apache](#)
 - ii. [Nginx](#)
 - b. Data Base Management System
 - i. [PostgreSQL](#)
 - ii. [MySQL](#)
 - c. GUI per SQLite
 - i. [DB Browser for SQLite](#)
 - ii. [SQLite Studio](#)
 - iii. [DBeaver](#)
 - d. Learning Management Systems
 - i. [Moodle](#)
 - ii. [Leemons](#)
 - e. Consegna dei compiti di programmazione
 - i. [CMS](#)
 - ii. [CMSSocial](#)
 - f. Piattaforma per il Cloud
 - i. [Nextcloud](#)
 - g. Tutela della Privacy
 - i. [Cryptomator](#)
 - ii. [GNU GPG](#)
 - iii. [Kleopatra](#)

4. Ipotesi architetturali

Il Dipartimento reputa interessanti alcune ipotesi da sottoporre al Team dell'Innovazione, al progettista della rete dati, al Responsabile della Protezione dei Dati e ai responsabili del Test Center ICDL.

4.1. Separazione delle sotto-reti

Le reti dei laboratori devono essere separate dalla restante infrastruttura dell'Istituto e

possibilmente separate tra loro.

Le reti devono essere monitorate con un software apposito quale Veyon.

Le reti devono essere filtrate da un firewall dedicato, sia esso hardware o software, che è controllato dal docente.

Se vi è un hotspot WiFi nei laboratori, questo deve essere configurato sulla stessa rete del laboratorio al fine di consentire le stesse operazioni delle postazioni fisse.

Una diagramma che illustra un possibile schema di indirizzi è quello in [Figura 1](#). Il *FUSS Server* indicato nel diagramma è un server dove è installato il software per il monitoraggio e il filtraggio della rete.

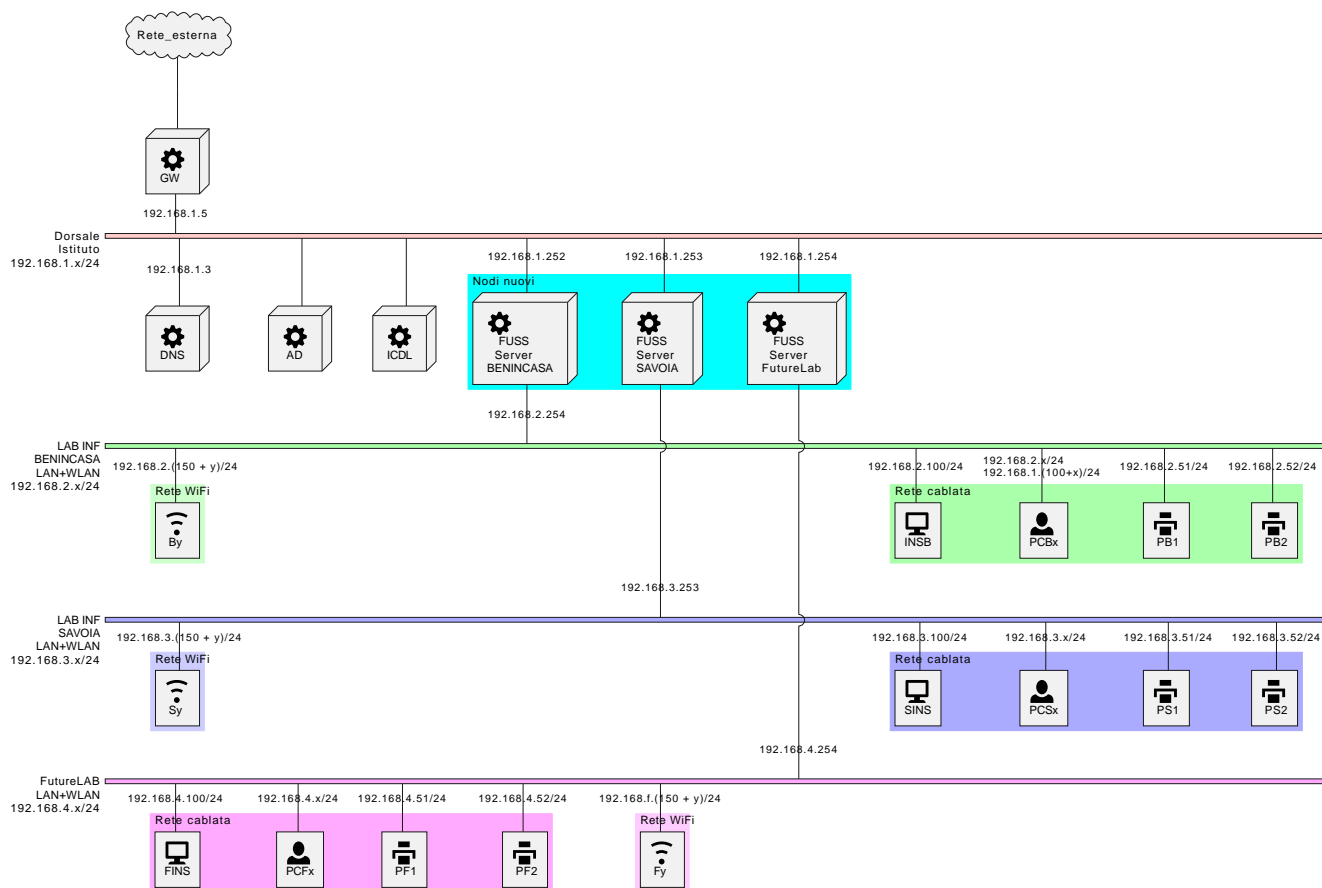


Figura 1. Architettura di rete

4.2. FUSS Server con Veyon ma dual boot

Una proposta mirata alla minimizzazione del lavoro necessario alla transizione del software è quella, una volta partizionata la rete e installato un server FUSS - su macchina fisica o virtualizzata - di installare i client FUSS mantenendo il dual boot con Windows. Il boot loader del client deve avviare in modo predefinito con FUSS e permettere la scelta di Windows ®. All'avvio di Windows deve avviarsi il servizio Veyon.

L'architettura di riferimento è quella del progetto [FUSS](#), meglio dettagliato nella [pagina](#)

informativa.

A differenza del progetto FUSS, l'idea è quella di avere la possibilità scegliere al boot se avviare la distribuzione FUSS oppure Windows.

Nel caso di avvio con Windows, sarebbe possibile usare, nel Laboratorio di Informatica del plesso Benincasa, la rete "192.168.1.x/24" con i precedenti indirizzi per DNS e Gateway, in modo da mantenere la compatibilità piena con la situazione attuale. Il FUSS Server si limita ad inoltrare i pacchetti provenienti e destinati alla rete "192.168.1.x/24". Se il boot avviene con FUSS Client, allora la rete è la "192.168.2.x/24", il default Gateway e il DNS puntano al FUSS Server e l'accesso alla rete avviene tramite autenticazione al portale Captive usando gli account in Active Directory.

I [servizi](#) dovrebbero essere quelli predefiniti nella distribuzione FUSS Server.

4.2.1. Transizione

Per gestire il periodo di transizione si deve tenere conto delle seguenti specifiche.

1. Il sistema deve permettere la coesistenza con le tecnologie correntemente in uso
2. La nuova configurazione non deve interrompere il servizio ICDL
3. La rete WiFi è attualmente gestita da MasterCom con politiche di amministrazione ad hoc
4. La nuova configurazione non deve interrompere il servizio di stampa
5. La nuova configurazione non deve sostituire il firewall ma integrarlo

Nella fase di transizione la rete WiFi all'interno del laboratorio può restare così com'è.

4.2.2. Servizi

I FUSS Server hanno un sistema operativo Linux con Gufw come interfaccia al firewall software UFW, inoltre si possono aggiungere regole al DNS.

Il sistema permette l'accesso in [ssh](#) anche con il server grafico [X](#). Il sistema deve configurare come suo default gateway il nodo 192.168.1.5 e come DNS il nodo 192.168.1.3.

La gestione della rete dovrebbe essere già preconfigurata ma nel caso si debbano gestire i precedenti indirizzi di rete si deve abilitare il solo IP forwarding, liberando risorse del kernel.

La RAM dovrebbe essere di almeno 16 GB, il numero di CPU pari ad almeno 8, lo spazio su disco di almeno 256 GB.

I nodi FUSS possono essere PC fisici o macchine virtuali.

Il docente deve possedere un account sui FUSS Server e deve essere nella lista dei [sudoers](#) (gruppo [sudo](#)).

Sui computer degli studenti deve essere installato FUSS Client con possibilità di *dual-boot*. Nel caso di avvio con Windows, deve essere impostato [Veyon](#) per consentire il monitoraggio.

I servizi offerti dal FUSS Server sono:

- Apache HTTPD Web server
- IP stateful firewall e router
- HTTP Proxy Cache
- HTTP Proxy Content Filter
- Proxy Secure Shell

Quelli offerti da FUSS Client sono:

- Autenticazione degli utenti tramite server LDAP
- Download e installazione dei certificati SSL necessari
- Mount delle directory HOME tramite NFS
- Installazione chiavi di autenticazione SSH
- Sincronizzazione NSCD
- Inserimento in configurazione cluster
- Creazione di utente locale di amministrazione

[L'installazione](#) ha un wizard.

Si potrebbe usare anche un solo FUSS Server creando due cluster, uno per laboratorio.

4.2.3. Pro e contro della soluzione proposta

4.2.3.1. Pro

- Tutte le richieste dei docenti di informatica sono soddisfatte
- Impatto sulla restante rete molto limitato

4.2.3.2. Contro

- Il nuovo nodo rappresenta un punto guasto aggiuntivo
- Se il nuovo nodo diventa indisponibile, la rete del laboratorio non è operativa
- la banda disponibile per PC studente si riduce notevolmente in quanto su un'unica scheda di rete ed un unico host viene gestita, via software, la connessione di tutte le postazioni del laboratorio
- il FUSS sever potrebbe essere congestionato aumentando i tempi di latenza

4.3. Macchine virtualizzate

Una soluzione adottata da molti Istituti è quella di fornire un ambiente di calcolo, detto *classe virtuale*, che esegue il software su un server dell'Istituto che è accessibile dai PC del laboratorio come da casa. Di norma sono realizzate tramite tante macchine virtuali installate su un server quante sono le classi omogenee, e possono avere sistemi operativi Windows - con costi di licenza - o Unix-like - senza costi. Nella macchina virtuale sono installati e configurati dall'assistente tecnico o

dal docente i software per la didattica.

Example 1. Uso delle classi virtuali

- L'utente accede alla propria classe mediante una porta TCP/IP ben specifica.
- Lo studente accede alle macchine virtuali con le stesse credenziali di accesso ai PC (cognome.nome e password come da sistema LDAP).

Esempio di connessione da scuola:

- MACCHINA VIRTUALE 1ATUR : ip= 10.2.2.13
- MACCHINA VIRTUALE 5FSA: ip=10.2.2.51

4.3.1. Pro e contro

4.3.1.1. Pro

- i computer del laboratorio si comportano come terminali di accesso alle macchine virtuali e necessitano di risorse minimali;
- si semplifica lo svolgimento dei compiti domestici e il loro controllo;
- si permette agli studenti di cambiare postazione nel laboratorio mantenendo l'accesso ai loro dati;
- si possono controllare gli accessi di ciascun utente connesso alla macchina virtuale, anche al fine del controllo dei compiti;
- si fornisce la possibilità di usare le macchine virtuali anche per l'ausilio alla docenza di altre materie. Se, ad esempio, il docente di Disegno ha bisogno di un CAD, potrebbe creare una macchina virtuale dedicata da condividere con i suoi studenti.

4.3.1.2. Contro

- Spreco di risorse interne in quanto i PC del laboratorio sarebbero utilizzati come *thin client*;
- Rinnovo costi di licenza e violazione del CAD nel caso di software proprietario (Microsoft Windows ®) sul server;
- Necessità di ingenti risorse di calcolo e di archiviazione sul server.
- Consumo di banda.

Riferimenti

[1] Agenzia per l'Italia Digitale AgID, «Linee guida su acquisizione e riuso di software per le pubbliche amministrazioni». https://www.agid.gov.it/sites/default/files/repository_files/lg-acquisizione-e-riuso-software-per-pa-docs_publicata.pdf, 2019.

[2] M. Berra e A. R. Meo, *Informatica solidale. Storia e prospettive del software libero*. Bollati Boringhieri, 2001.

- [3] M. Berra e A. R. Meo, *Informatica solidale 2. Libertà di software, hardware e conoscenza*. Bollati Boringhieri, 2006.
- [4] Commissione Europea, «Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance)». 2016, [Online]. Disponibile su: = <https://eur-lex.europa.eu/eli/reg/2016/679/oj>.
- [5] R. Davoli, «L'informatica nella Scuola: strumenti e metodi, scienza e tecnologia. La libertà di usare il software libero». http://erlug.linux.it/linuxday/2010/contrib/davoli_bo.gld.2010.odp , 2010.
- [6] R. Davoli, «... tutto quello che avreste voluto sapere sui laboratori, ma non avete mai osato chiedere ..». https://cs.unibo.it/renzo/00-uso_lab.pdf , 2012.
- [7] DECRETO LEGISLATIVO 7 marzo 2005 n 82, «Codice dell'Amministrazione Digitale». <https://www.normattiva.it/uri-res/N2Ls?urn:nir:stato:decreto.legislativo:2005-03-07;82> , 2006–2022.
- [8] A. R. Formiconi, «Azioni di ricerca nel segno di Dewey, Papert, Freire, Don Milani e Morin, soprattutto..». <https://iamarf.org/> , 2022.
- [9] A. R. Formiconi, «Coding a scuola con Software Libero (edX)». <https://www.edx.org/course/coding-a-scuola-con-software-libero> , 2022.
- [10] A. R. Formiconi, «Coding a scuola con Software Libero (Federica)». https://federica.eu/c/coding_a_scuola_con_software_libero/ , 2022.
- [11] Free Software Foundation (FSF), «Domande ricorrenti su software libero ed istruzione». <https://www.gnu.org/education/edu-faq.it.html> , 2021.
- [12] Free Software Foundation (FSF), «Software libero ed istruzione». <https://www.gnu.org/education/education.html> , 2022.
- [13] Free Software Foundation (FSF), «Perché le istituzioni educative devono usare ed insegnare il software libero». <https://www.gnu.org/education/edu-why.html> , 2022.
- [14] I. Georgiadi e K. Glotzbach, «Open Source Software: come usarlo a scuola e contemporaneamente aiutare la società». <https://www.schooleducationgateway.eu/it/pub/resources/tutorials/open-source-software.htm> , 2020.
- [15] A. R. Meo, «Lettera alla ministra Azzolina dal professor Angelo Raffaele Meo e le associazioni promotrici del software libero». <https://scuolalibera.continuity.space/lettera-professor-meo> , 2019.
- [16] A. R. Meo, «Scuola e università: perché preferire il software libero». <https://gliasinirivista.org/scuola-e-universita-perche-preferire-il-software-libero/> , 2022.
- [17] R. Stallman, «Perché la scuola deve usare esclusivamente software libero». <https://www.gnu.org/education/edu-schools.html> , 2021.
- [18] R. Stallman, «Quel server in realtà a chi serve?» <https://www.gnu.org/philosophy/who-does-that-server-really-serve.it.html> , 2021.

- [19] R. Stallman, «Cos'è il software libero?» <http://www.gnu.org/philosophy/free-sw.it.html> , 2022.
- [20] G. Tesio, «A scuola da Google». <https://gliasinirivista.org/a-scuola-da-google/> , 2022.
- [21] S. Zoja, «Così il controllo sulla didattica rafforza lo strapotere delle multinazionali». <https://altreconomia.it/la-scuola-italiana-al-mercato-dei-dati/> .

Gionata Massi per il Dipartimento di Informatica

Scritto con Software Libero

Editor: VS Codium

PDF generator: `asciidoctor-pdf -v -r asciidoctor-bibtex -r asciidoctor-diagram`