

Proposte per migliorare la didattica nei Laboratori

Indice

1. Problematiche dei Laboratori	2
1.1. I Laboratori d'Informatica	2
1.2. Il Future Lab	3
2. Requisiti dell'ambiente laboratoriale di Informatica	4
2.1. Applicazioni	5
3. Architettura ipotizzata	7
3.1. Transizione	9
3.2. Servizi	9
3.3. Pro e contro della soluzione proposta	10
3.3.1. Pro	10
3.3.2. Contro	10
4. Laboratori in mobilità	10
4.1. Pro e contro	11
4.1.1. Pro	11
4.1.2. Contro	11
5. Migrazione al Software Libero e art. 68 CAD	11

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**

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

1. Problematiche dei Laboratori

1.1. I 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.

Monitoraggio

Una problematica consiste nell'uso, da parte degli studenti, delle tecnologie per fini non didattici che l'insegnante, mentre è impegnato nella presentazione della teoria o nell'illustrazione delle tecniche e dei metodi, non riesce a rilevare. Capita, infatti, che nelle classi numerose gli studenti aprano video-giochi piuttosto che guardino pagine non inerenti l'attività didattica.

Condivisione di videate

Un'altra problematica è la condivisione delle videate che, sebbene supportata dal proiettore, resta problematica specie per gli studenti delle ultime file.

Stesse versioni del software

Alle volte il software richiesto per l'attività didattica risulta non disponibile in alcuni PC o installato in versioni diverse e incompatibili.

Aggiornamento

L'aggiornamento di un software richiede tempo in quanto ogni postazione deve essere aggiornata.

Controllo fine della rete

Per alcune verifiche è fatto divieto di accedere a risorse esterne al proprio PC oltre a quelle del servizio di erogazione della prova e allo stato attuale il docente non ha controllo sul filtraggio di rete.

Blocco delle pubblicità

Per l'accesso ad alcune risorse capita che vi siano pubblicità inserite nelle pagine web e queste causano distrazione oltre a poter presentare contenuti non adatti all'ambito scolastico.

Reti cablate e senza fili unificate

Le reti cablate e senza fili sono separate e non vi è un bridge che le renda interoperabili. Per un docente è impossibile raccogliere le prove di uno studente che opera con il suo dispositivo senza

passare tramite lo scambio, insicuro, di dispositivi di storage o l'uso di piattaforme esterne.

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

L'attuale scelta dei sistemi inoltre vincola all'utilizzo di ogni dispositivo come stand-alone e solo con un sistema operativo.

Sarebbe auspicabile poter disporre di sistemi operativi aperti, con la possibilità di scegliere il sistema operativo da avviare, software e risorse di calcolo condivise.

Accesso limitato a social network e siti di streaming

Sarebbe auspicabile filtrare l'accesso ai siti di social network e streaming consentendone solo esplicitamente l'accesso.

Sistemi a codice sorgente aperto

Per la didattica del linguaggio C++ (classi terze ITE SIA e Liceo Scientifico sia Scienze Applicate che opzione Informatica), dei sistemi operativi (tutte le classi), delle reti di calcolatori (tutte le classi), delle tematiche di cybersicurezza (tutte le classi) sarebbe opportuno lavorare con Sistemi Operativi liberi quali GNU/Linux o BSD.

Clustering

Per la sperimentazione di alcuni algoritmi particolarmente onerosi potrebbe richiedere l'utilizzo di più PC come se questi fossero un'unica potente workstation. L'architettura hardware e software deve permettere la creazione di cluster.

Cloud privato

Per riprendere il controllo dei dati ed evitare le preoccupazioni causate dall'uso di servizi cloud gestiti da fornitori esteri, sarebbe interessante da valutare l'introduzione di una piattaforma di condivisione gestita internamente.

1.2. Il Future Lab

Separazione dall'atrio dell'Aula Magna

Attualmente il Future Lab non è separato dall'ingresso dell'aula magna e questo causa distrazioni e confusione.

Disponibilità di punti di punti luce

Attualmente la rete elettrica non dispone di punti luce tranne che sulle colonne e sulle postazioni fisse. Andrebbero create delle strutture per ospitare delle prese elettriche funzionali all'ambiente.

Disponibilità di punti prese di rete

Attualmente la rete dati dispone di prese ethernet solo nelle postazioni fisse. Nelle strutture presso le quali ospitare le prese elettriche sarebbero da integrare prese per la rete cablata e/o switch.

Disponibilità di PC/notebook

Non vi sono PC o notebook e quindi si deve confidare nei dispositivi degli studenti. Si potrebbero

utilizzare o i PC dei laboratori mobili, fra cui i circa 30 presenti nel magazzino del Benincasa e non ancora utilizzati, o anche acquistare dei sistemi economici quali le [Raspberry Pi 400](#).

Altre problematiche

Le altre problematiche sono le stesse dei laboratori

2. Requisiti dell'ambiente laboratoriale di Informatica

Per facilitare l'insegnamento in ambienti di apprendimento digitale devono essere soddisfatte le seguenti specifiche:

1. Il docente deve poter gestire 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 Linux, reale o virtuale, per creare e rimuovere utenti, applicazioni e servizi
 - b. La macchina linux si collega al servizio LDAP oppure mantiene l'elenco degli utenti
 - c. Il docente deve poter installare pacchetti sulla macchina Linux
 - d. Il docente deve poter avviare e terminare servizi sulla macchina Linux
 - e. I PC devono poter avviare le applicazioni sulla macchina Linux
 - i. I PC devono supportare lo X11 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 sul Drive e nelle cartelle condivise, 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

2.1. Applicazioni

Le applicazioni che dovrebbero essere installate sono:

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](#)
 - b. Software per l'editoria
 - i. [TeX Live](#)
 - ii. [Asciidoctor](#)
 - A. [Asciidoctor Diagram](#)
 - B. [Asciidoctor PDF](#)
 - C. [Asciidoctor EPUB3](#)
 - D. [Asciidoctor reveal.js](#)
 - E. [Ruby rouge](#)
 - F. [Ruby coderay](#)
 - G. [Ruby pygments.rb](#)
 - c. 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. [Valgrind](#)
- xii. [ghidra](#)
- xiii. [CMake](#)
- xiv. [GNU Make](#)
- d. 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](#)
- e. Strumenti di versionamento
 - i. [git](#)
- f. Accesso remoto
 - i. [ssh](#)
 - ii. [netcat](#)
 - iii. [openVPN client](#)
- g. Audio e video
 - i. [Audacity](#)
 - ii. [FFMPEG](#)
 - iii. [Shotcut](#)
- h. Browser
 - i. [Firefox](#)
 - ii. [Chrome](#)
- i. CAD
 - i. [FreeCad](#)
- j. Grafica
 - i. [GNU Image Manipulation Program](#)
 - ii. [Inkscape](#)
 - iii. [GraphViz](#)
 - iv. [GnuPlot](#)
 - v. [imagemagick](#)
 - vi. [PlantUML](#)
 - vii. [Kroki](#)

- k. Calcolo scientifico e ingegneristico
 - i. [GNU Octave \(octave-forge\)](#)
 - ii. [Scilab](#)
 - iii. [GeoGebra](#)
 - iv. [R](#)
 - v. [miniconda](#)
 - l. MindMapping
 - i. [FreePlane](#)
 - ii. [PlantUML](#)
 - m. Software per l'analisi della rete
 - i. [WireShark](#)
 - ii. [ping](#)
 - n. 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. Consegna dei compiti di programmazione
 - i. [CMS](#)
 - ii. [CMSSocial](#)
 - d. Piattaforma per il Cloud
 - i. [Nextcloud](#)
 - e. Tutela della Privacy
 - i. [GNU GPG](#)
 - ii. [Kleopatra](#)

3. Architettura ipotizzata

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.

In alternativa a FUSS potrebbe essere valido il software del progetto [SoDiLinux](#).

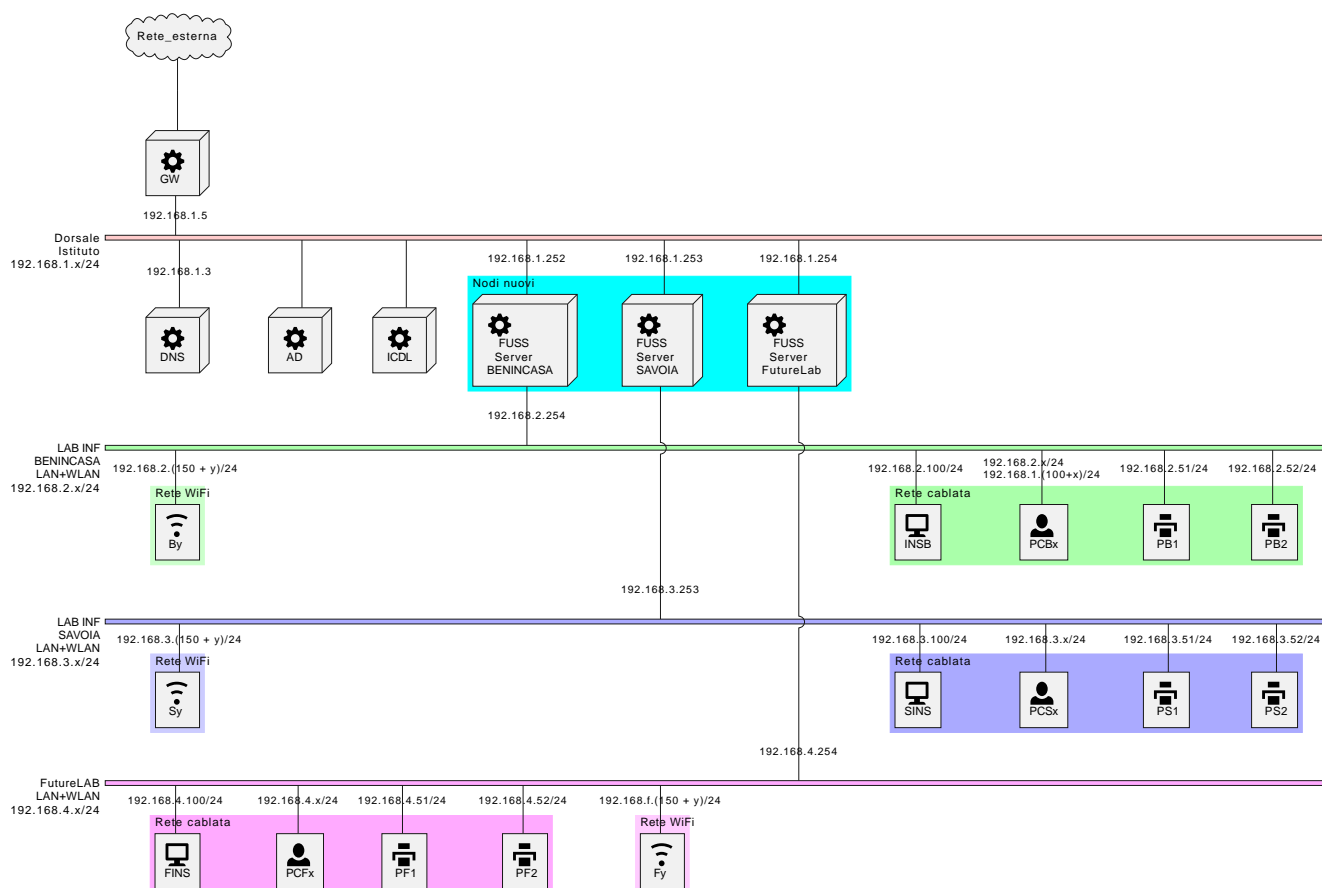


Figura 1. Architettura di rete

I [servizi](#) dovrebbero seguire lo schema seguente.

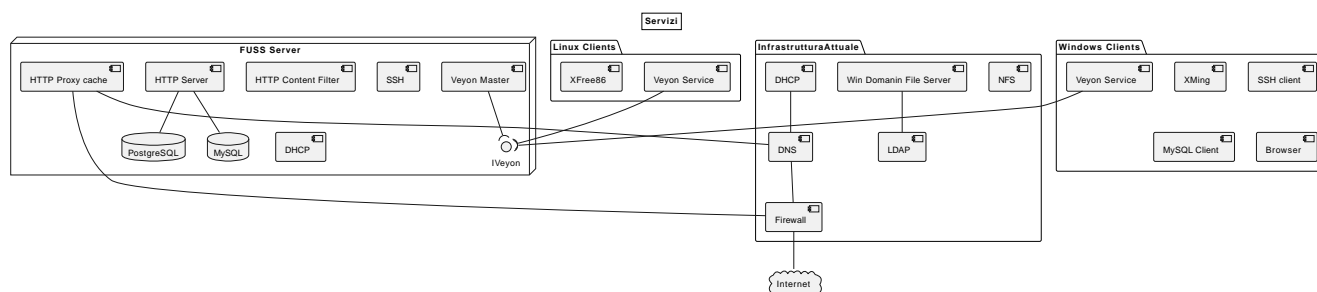


Figura 2. Servizi

3.1. 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.

3.2. 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.

3.3. Pro e contro della soluzione proposta

3.3.1. Pro

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

3.3.2. 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. Laboratori in mobilità

Una soluzione adottata da molti Istituti è quella delle di fornire un ambiente di calcolo, detto *classe virtuale*, che esegue il software su un server dell'Istituto ed 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

4.1. Pro e contro

4.1.1. 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 mantengono l'accesso ai loro dati;
- si possono controllare gli accessi 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.

4.1.2. Contro

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

5. Migrazione al Software Libero e art. 68 CAD

Per ridurre, se non azzerare, i costi di licenza del software in uso nei laboratori si potrebbe installare Linux, oppure BSD, come sistema operativo.

Questa opzione è da considerarsi in ottemperanza all'art. 68 del Codice per l'Amministrazione Digitale:

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.

Il rimando normativo è alle [Linee guida su acquisizione e riuso di software per le pubbliche amministrazioni](#).

Si evidenziano:

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[...]

A favore di questa opzione si citano le argomentazioni della Free Software Foundation:

- [Perché la scuola deve usare esclusivamente software libero](#) di Richard Stallman
- [Domande ricorrenti su software libero ed istruzione](#)

dei alcuni dei più illustri professori italiani di Informatica:

- [Scuola e università: perché preferire il software libero](#) di Angelo Raffale Meo
- [Lettera alla ministra Azzolina](#) di Angelo Raffale Meo
- [Importanza del Free and Open Source Software \(FOSS\) per l'istruzione.](#) di Andreas Formiconi
- [L'informatica nella Scuola: strumenti e metodi, scienza e tecnologia. La libertà di usare il Software Libero](#) di Renzo Davoli

