

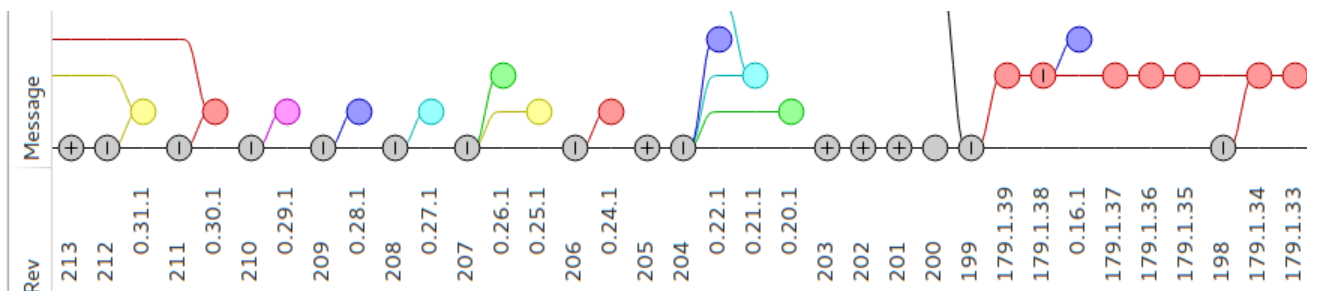


ITIS Amedeo Avogadro

A.S. 2013/2014

Tecnologie sostenibili per la Didattica

LTSP – L.I.M. - E-Register



Tesina di Valerio Bozzolan (5B Informatica)

Sommario

Premessa.....	3
LTSP (Linux Terminal Server project).....	4
Cos'è e dov'è.....	4
Analisi funzionale.....	5
Caratteristiche di una rete LTSP.....	5
Vista docente.....	6
Vista studente.....	7
Amministrazione.....	8
Analisi tecnica.....	9
Schemi di rete a confronto.....	9
Funzionamento dei client.....	10
Caratteristiche della rete.....	11
Caratteristiche del server.....	11
Caratteristiche thin client.....	11
Implementazione all'ITIS Avogadro.....	12
Situazione.....	12
Progettazione della VM.....	12
Installazione.....	13
Note.....	13
Lavagne Interattive Multimediali low-cost.....	14
Strumenti necessari.....	14
Schema di funzionamento.....	15
Note di implementazione.....	15
Un registro elettronico sostenibile.....	16
Note.....	17
Conclusioni.....	19
Riferimenti e fonti sitografiche.....	20

Premessa

L'oggetto della tesina è di portare a conoscenza di alcune tecnologie esistenti o in via di sviluppo che migliorerebbero il supporto alla didattica in maniera **low-cost** in molti Istituti scolastici.

Lo scopo è di ottenere questo risultato appoggiandosi al **software libero** e progetti **open source** anche in ottemperanza alla *Legge regionale n. 9 del 26 marzo 2009* contenente le *Norme in materia di pluralismo informatico, sull'adozione e la diffusione del software libero nella pubblica amministrazione*.^a

In modo particolare la tesina è incentrata sull'implementazione all'interno dell'ITIS Avogadro di Torino del progetto **Linux Terminal Server Project (LTSP)**, un progetto per i laboratori informatici. Altri due progetti saranno descritti brevemente poiché gemelli per simile importanza e sostenibilità:



Illustrazione 1: LTSP: Linux Terminal Server Project

- **Lavagne Interattive Multimediali (L.I.M.)**
Progetto *low-cost* per l'implementazione di lavagne interattive multimediali;
- **Registro Elettronico libero (ovvero Open Student)**
Un registro dei voti in via di sviluppo pubblicato sotto *licenza libera*.



Illustrazione 2: WiildOS: Sistema Operativo per le lavagne LIM nato in Italia

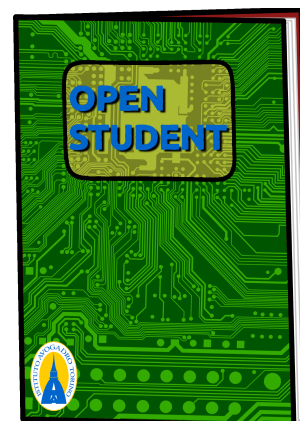


Illustrazione 3: Open Student: Registro dei voti in via di sviluppo

LTSP (Linux Terminal Server project)

Cos'è e dov'è

LTSP è un progetto nato originariamente per permettere a scuole meno abbienti di creare reti di computer laddove le risorse economiche fossero limitate. Computer obsoleti o minimali (thin client) possono essere riutilizzati in una rete LTSP, in quanto il carico di lavoro è delegato ad un server (o più). Il progetto può prevedere anche alcune varianti piene di potenzialità, come quella dei FAT client. *Grecia, Nepal, Indonesia, Uganda e Kenia* sono solo alcuni dei paesi noti per avere implementato per primi LTSP in molte delle loro scuole per creare dei laboratori professionali **low-cost**.



Illustrazione 4: LTSP nel mondo

Vi sono appena 700 implementazioni ufficiali di LTSP nel mondo delle quali solo 9 in Italia e, da quest'anno, nel nostro Istituto (Illustrazione 6). Inoltre sempre quest'anno un Istituto di Biella ha implementato LTSP proprio grazie al materiale di questa tesina.^b



Illustrazione 5: LTSP: Dettaglio in Europa

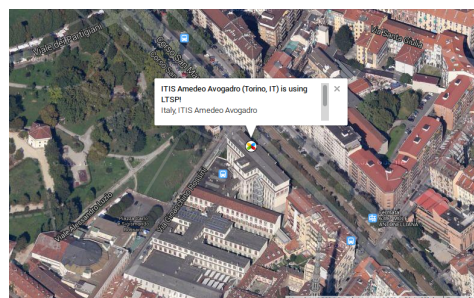


Illustrazione 6: LTSP: Dettaglio su Torino

Analisi funzionale

Caratteristiche di una rete LTSP

- **Ingente diminuzione dei costi per i nuovi laboratori**

Per un nuovo laboratorio sarà necessario acquistare soltanto *thin client* (speciali PC minimali che costano 20€-60€ a postazione) insieme ad un server (di costo simile a quello di un paio di nuovi PC di fascia media). Il tutto rende il progetto assolutamente **low-cost**;

- **Risparmio energetico**

I thin client sono computer privi di parti in movimento: Idealmente con nessun hard-disk, lettori CD/DVD e sovente persino senza ventole. Ciò significa che un thin client consuma all'incirca dai 7 ai 30 Watt, in confronto ad un comune PC medio-basso che consuma dai 60 a più di 150 Watt (un confronto nell'Illustrazione 7);^c

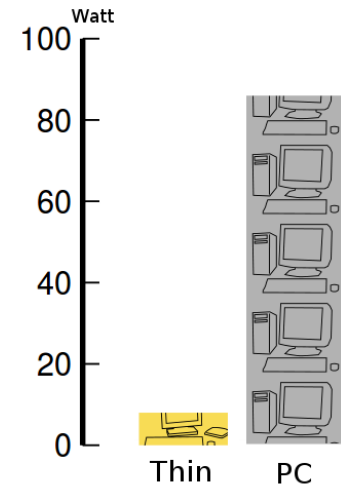


Illustrazione 7: Consumi energetici a confronto

- **Radicale centralizzazione del software**

Sui singoli client non vi è alcun software da mantenere poiché si avviano tramite rete. Tutto il software è installato soltanto sul server e dal server poi tutto è condiviso a tutte le postazioni di lavoro, all'istante.

Al contrario, nelle reti "usuali", un programma deve essere installato e mantenuto aggiornato su ogni singolo computer;

- **Alcuna necessità di "Reset"**

Alcuni Istituti (come il nostro) modificano i PC per far sì che si resettino ad ogni avvio, ciò principalmente per contrastare il *malware*.

Al contrario LTSP incoraggia ad avere sempre i propri documenti e le proprie impostazioni salvate: La solida architettura di GNU/Linux garantisce la sicurezza dei propri dati anche contro eventuali utenti malintenzionati.

È da sottolineare che il docente potrà **porre restrizioni** all'uso di Internet, aumentando la sicurezza della rete stessa;

- **Maggiore supporto alla didattica**

Il docente può diffondere il proprio monitor sugli schermi degli studenti (funzionalità molto utile in aule prive di proiettore);

- **Lezioni più dinamiche**

Il docente dalla sua postazione può intervenire direttamente sui computer degli studenti;

- **Azzeramento dei costi di licenza**

Ultimo ma non meno importante: Tutto il software necessario per il funzionamento del progetto è rigorosamente **software libero**^d.

Vista docente

Il docente in una rete LTSP potrà apprezzare un software educativo chiamato *Epopetes* la cui interfaccia è visibile nell'Illustrazione 8. Epopetes permette di gestire in maniera molto intuitiva i computer di un laboratorio.

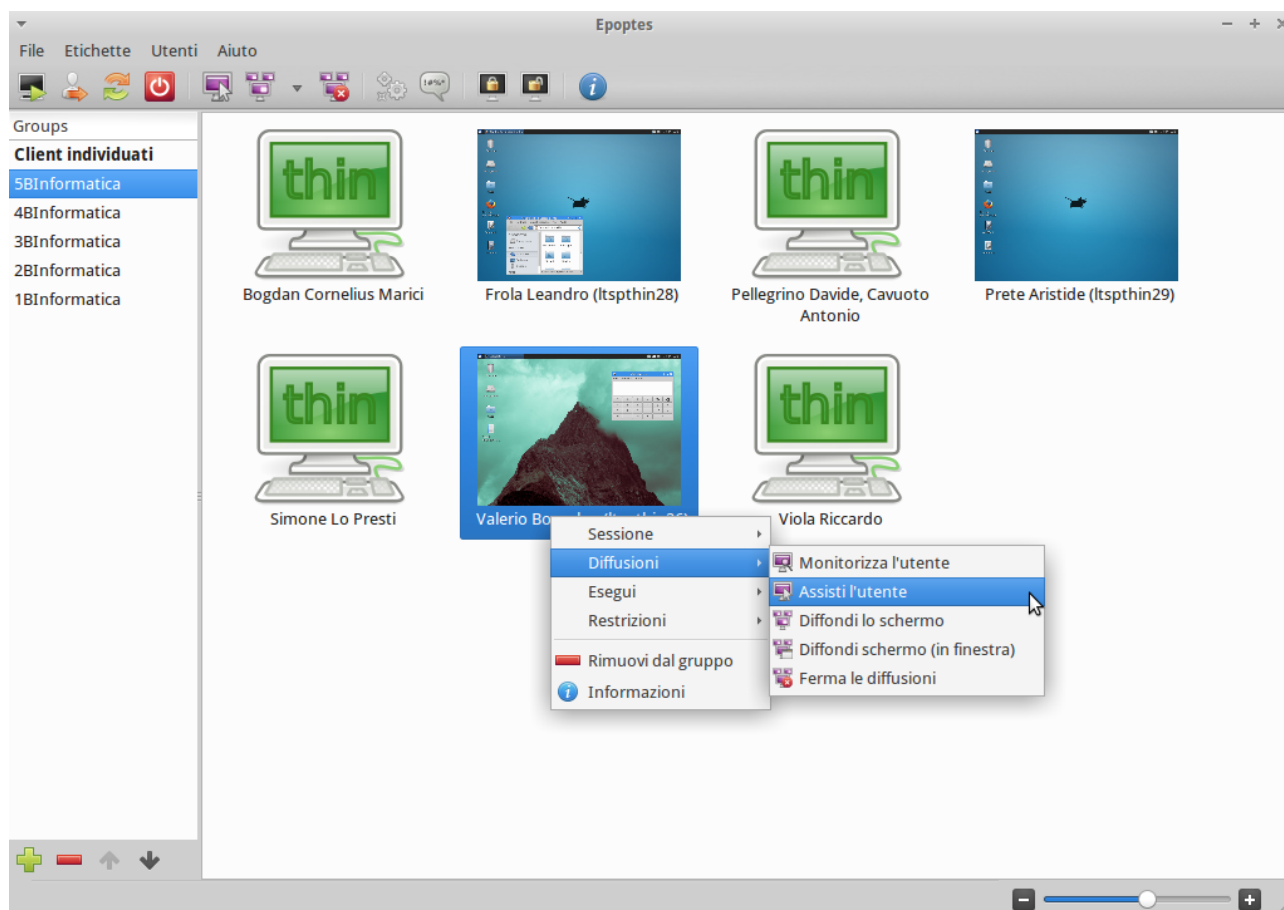


Illustrazione 8: Epopetes: Software per la didattica in una rete LTSP

Le opportunità offerte da questo strumento sono le più svariate in quanto non solo permette al docente di visualizzare i monitor dei client della rete ma anche di gestirli nella loro totalità. Alcuni esempi:

- Spegnere/riavviare i client;
- Bloccare/sbloccare gli schermi;
- Monitorare/assistere lo studente;
- Comunicare in modo diretto;
- Diffondere il proprio schermo su quello degli studenti (Questa funzionalità è particolarmente utile per effettuare presentazioni ove non vi sia un proiettore);
- Inviare comandi in background (come azzerare il volume audio di un PC)

Vista studente

Dopo l'accensione un thin client impiega meno di un minuto per essere operativo, giusto il tempo di allacciarsi al server LTSP e poter effettuare il login (Illustrazione 9).

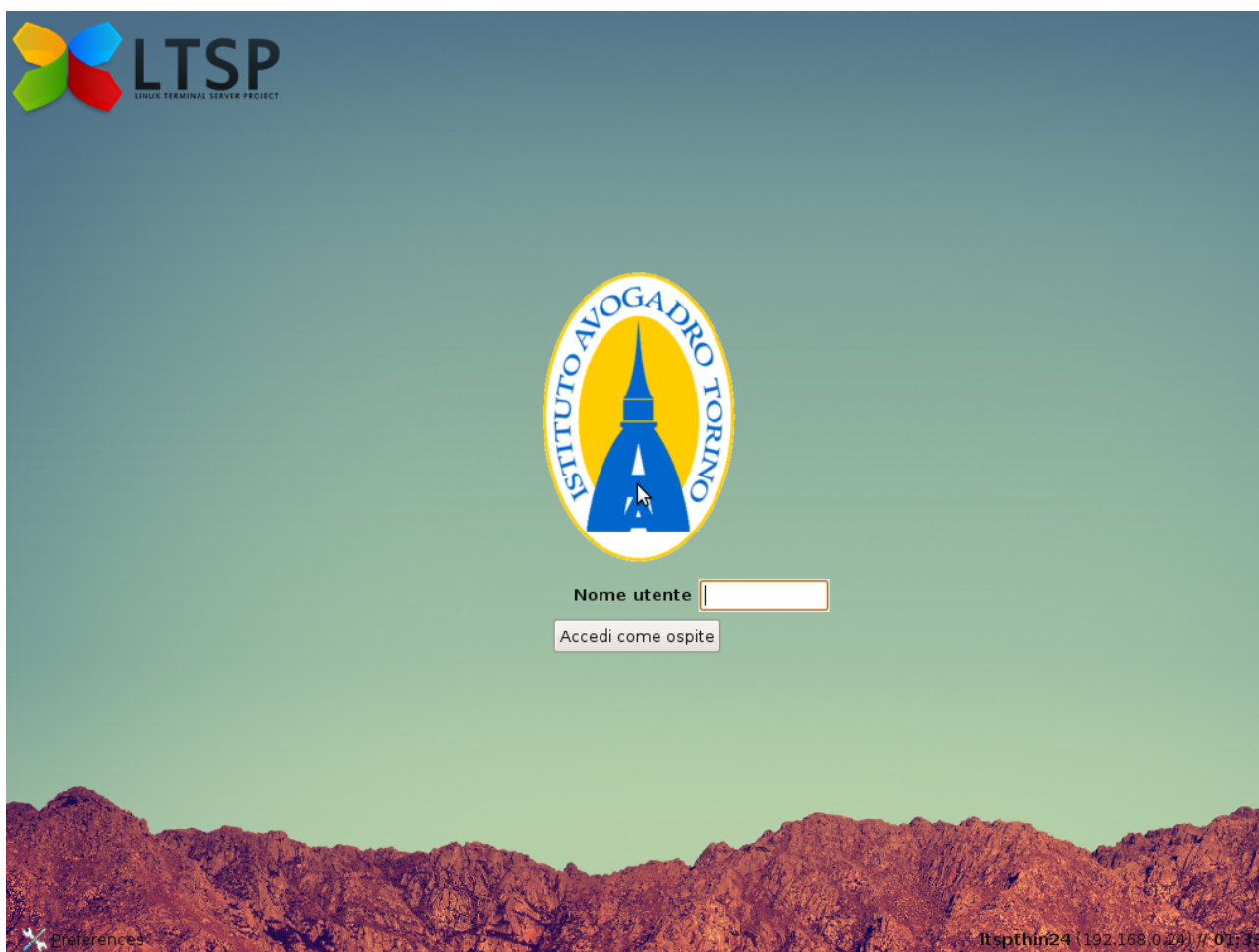


Illustrazione 9: Schermata di login di un client LTSP all'Avogadro

Si ha la possibilità di entrare come ospite oppure come utente registrato mantenendo i propri documenti e le proprie impostazioni direttamente salvate sul server.

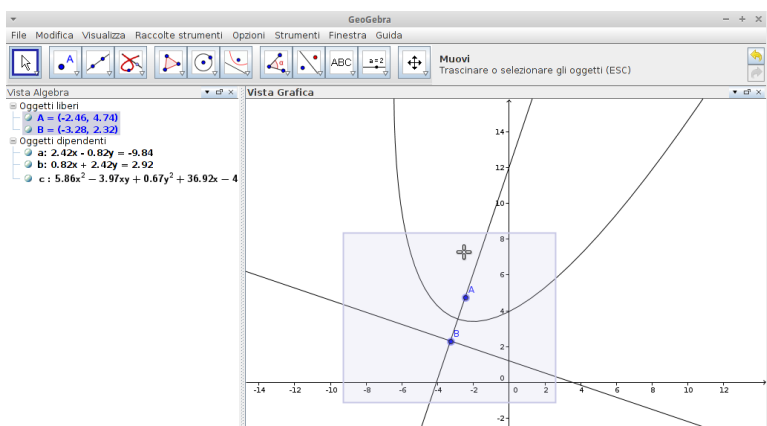


Illustrazione 10: Utilizzo di GeoGebra su un thin client

Dopo aver effettuato l'accesso il computer è subito pronto all'uso per l'utilizzo dei propri programmi didattici preferiti come **GeoGebra**, **NetBeans**, **Arduino IDE**, **LibreOffice**, **Firefox**, **Audacity**, **VLC** e **Gimp**; citati a puro titolo di esempio.

Amministrazione

Il tecnico di laboratorio potrà utilizzare un account di amministrazione per installare i programmi direttamente sul server e per renderli così disponibili a tutti i client. Potrà farlo anche attraverso l'**Ubuntu Software Center** (Illustrazione 11).

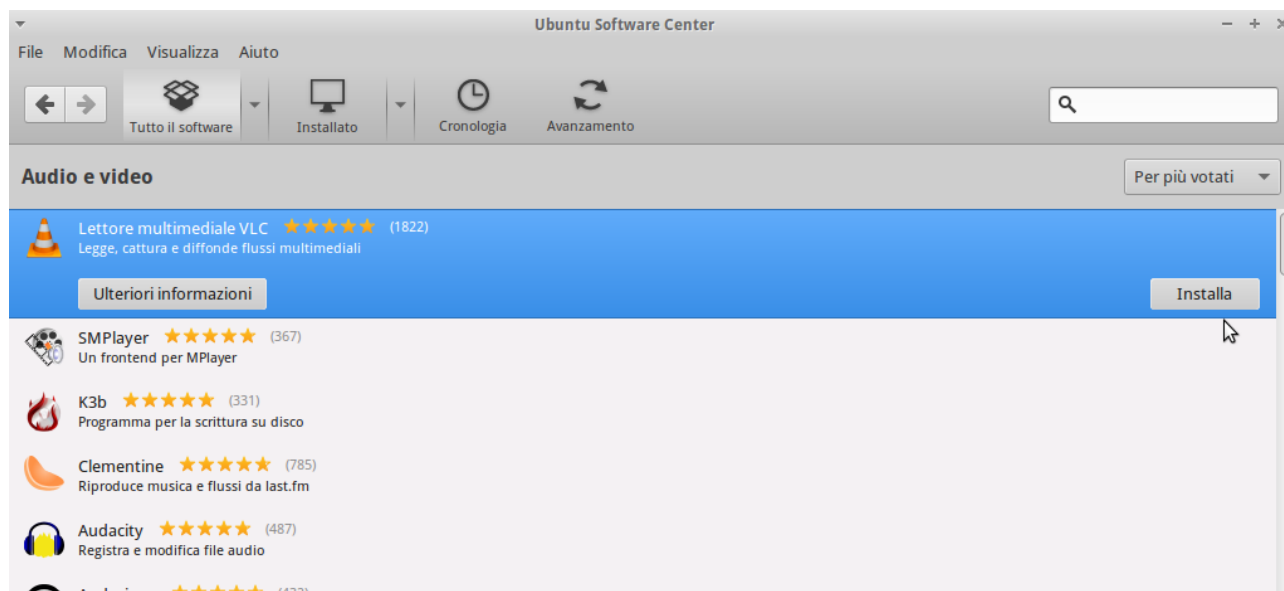


Illustrazione 11: Ubuntu Software Center

Inoltre anche la creazione e gestione degli utenti e dei gruppi è molto semplice utilizzando la relativa interfaccia **Impostazioni utenti**.

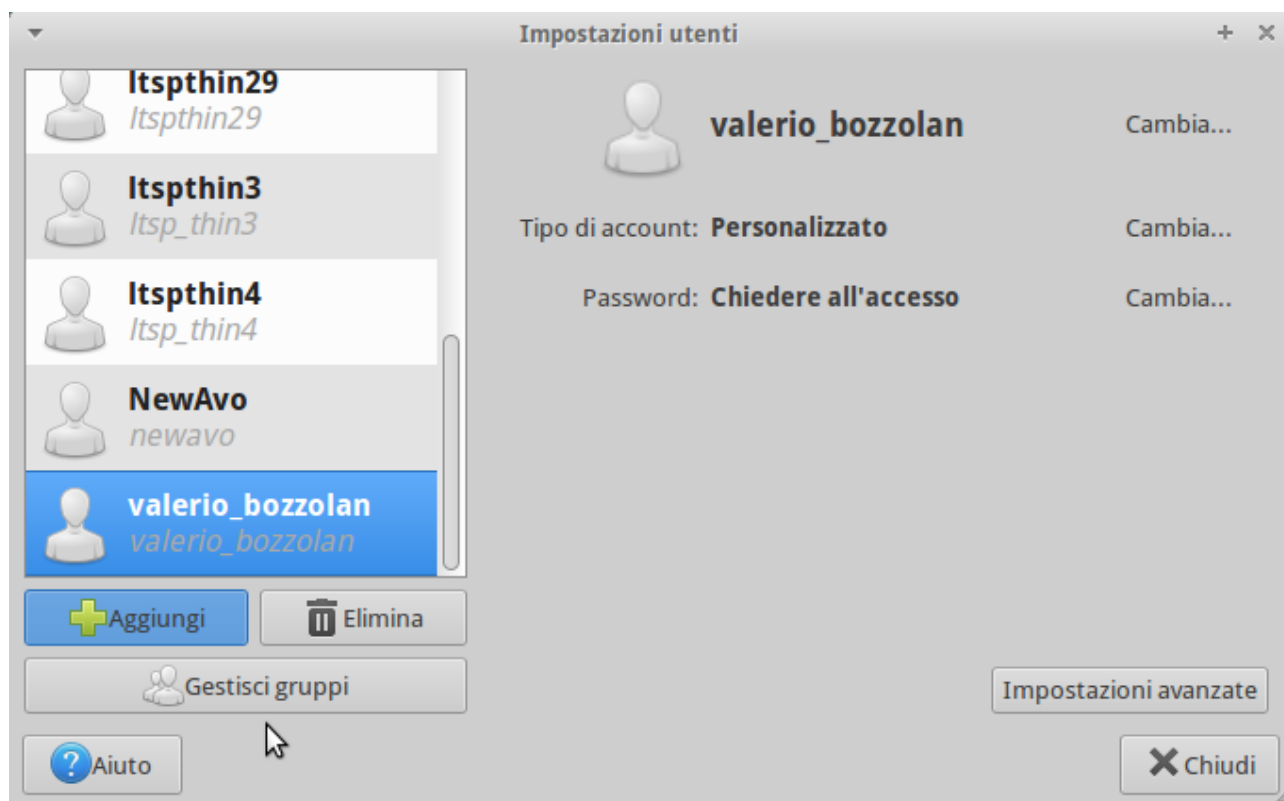


Illustrazione 12: Esempio di creazione utente "Valerio Bozzolan"

Analisi tecnica

Schemi di rete a confronto

Una rete "tradizionale" di un singolo laboratorio è solitamente rappresentabile con lo schema dell'illustrazione 13.

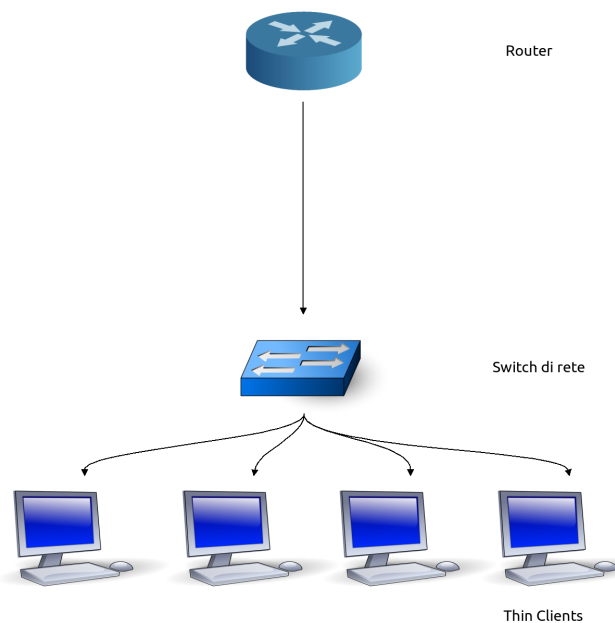


Illustrazione 13: Schema di rete tradizionale

Invece dallo schema di una rete LTSP (visibile sotto) si può notare come soltanto un nuovo elemento (il server) sia da inserire in una rete "tradizionale" per renderla LTSP.

Il server avrà ovviamente due schede di rete: una per comunicare con i client e farli partire via rete (e fare da server DHCP, TFTP, NDB, SSH, eccetera); l'altra per usare il router come *default gateway* e permettere ad i client di uscire in Internet.

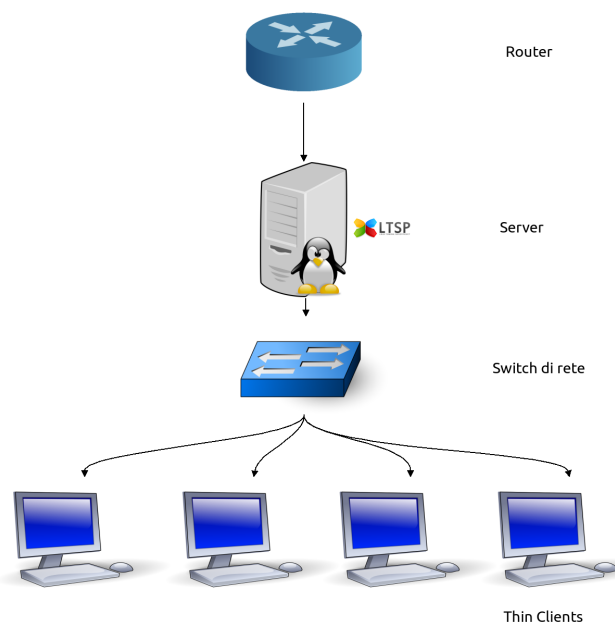


Illustrazione 14: Schema di rete LTSP

Funzionamento dei client

Si possono schematizzare le procedure di avvio di un client a seconda se si tratta di un thin client o di un fat client.

Thin client

Il thin client è un computer minimale privo idealmente di parti in movimento (hard disk, lettori CD, a volte persino senza ventole) che dipende direttamente dal server poiché ivi vi istanzia i propri processi e vi monta il proprio filesystem (via rete). Il suo avvio è così schematizzabile:

1. Wake on LAN

Il thin client può rimanere spento con la scheda madre in *Wake on LAN* affinché il docente possa, tramite un apposito software, accendere automaticamente i computer desiderati;

2. Network Boot

Una volta acceso, il client si avvia in rete locale (PXE) anziché via hard disk^e:

1. Il client richiede un indirizzo IP dal server tramite il protocollo DHCP;

2. Il client riceve le istruzioni per scaricare un sistema minimale (thin);

3. Il client scarica questo sistema minimale tramite TFTP e lo carica in RAM;

3. Il client può ora rilevare l'hardware a disposizione;

4. Il client monta un filesystem di rete NDB (Network Block Device) per avere documenti e file salvati direttamente sul server;

5. Il client apre una sessione grafica sul server tramite LDM e X-SSH



Illustrazione 15: Un thin client

Fat client

I fat client idealmente sono identici ai thin client, differendo soltanto nel fatto che sono muniti di caratteristiche RAM e CPU simili a quelle di un normale PC. Ciò permette loro di poter scaricare all'avvio un sistema complesso (fat) dal server LTSP. Questa immagine di sistema completa gli permette di rendersi indipendente a livello di processi dal server e in questo modo si sfruttano le caratteristiche di CPU e RAM del singolo client senza appesantire il carico del server.

Anche in ogni caso si ottiene una gestione centralizzata del laboratorio e con un significativo aumento delle prestazioni rispetto ad una rete di soli thin client.

Caratteristiche della rete

È consigliabile che, per il gran numero di connessioni, sia lo switch che per lo meno la scheda di rete che comunica con i client siano da 1Gbit/s (pagina 9, Illustrazione 14).

Caratteristiche del server

RAM

Per un server vi è un calcolo approssimativo per capire se la RAM è sufficiente:^f

$$\text{RAM del server in MB} = 1500 + (30 * \mathbf{N_FAT_CLIENT}) + (300 * \mathbf{N_THIN_CLIENT})$$

Un esempio in una rete di 20 thin client:

$$1500 + (30 * 0) + (300 * 20) = \text{all'incirca 6GB di RAM}$$

La RAM disponibile è basilare per cui per ottenere buone prestazioni è bene prendere in considerazione di aumentare la RAM consigliata da questo calcolo approssimativo. In questo caso due banchi da 4GB di RAM sono già una buona scelta. Tutto ciò per evitare la possibilità che il server sia costretto allo *swapping*.

CPU

Un server dual core con almeno 2.5Ghz è più che sufficiente per godere di discrete prestazioni su una decina di thin client. Per un numero superiore di thin client o per avere prestazioni superiori (e sempre in caso di disponibilità economica) sarebbe meglio prevedere un processore di famiglia Xeon almeno quad-core da 3Ghz.

Disco rigido

È consigliabile applicare qualche forma di RAID all'interno del server, come per esempio un RAID di tipo "1" formato da un paio di dischi SATA con tecnologia *Native Command Queuing*^g per migliorare le performance con costi veramente contenuti. Per budget maggiori si può optare per un RAID "10" con dischi 10,000 RPM o SSD per ottenere la massima velocità possibile e sufficiente sicurezza in caso di guasto di un disco.

Caratteristiche thin client

RAM

Non esistendo più banchi di RAM da 48MB (che è la RAM minima richiesta per un thin client), praticamente qualunque computer obsoleto è convertibile in thin client. In aggiunta un thin client raramente arriva ad utilizzare più di 256MB di RAM.

CPU

Un thin client con un single core da 533MHz ha già performance più che accettabili.

Implementazione all'ITIS Avogadro

Situazione

Raffaele Iacono, docente tecnico del laboratorio di Fisica 1 del nostro Istituto, ci ha messo al corrente di una situazione tipica per l'implementazione di una rete LTSP:

Nel suo laboratorio almeno una dozzina di computer giacciono inutilizzati poiché in stato di obsolescenza e, se convertiti in thin client per una rete LTSP, saranno in grado di essere operativi solo con l'aggiunta di un server. È stata fatta precisazione che nel laboratorio si utilizzerebbe molto il software **GeoGebra**.

Progettazione della VM

È stata così preparata una installazione di LTSP in *Virtual Machine* (attraverso **VirtualBox^h**). Questa scelta per poter far sì non solo che il sistema operativo per il server sia pronto per agire nel laboratorio di fisica ma anche in altri futuri. È stato scelto come ambiente grafico **XFCE4** data la sua grande velocità e semplicità anche su scarse risorse hardware.

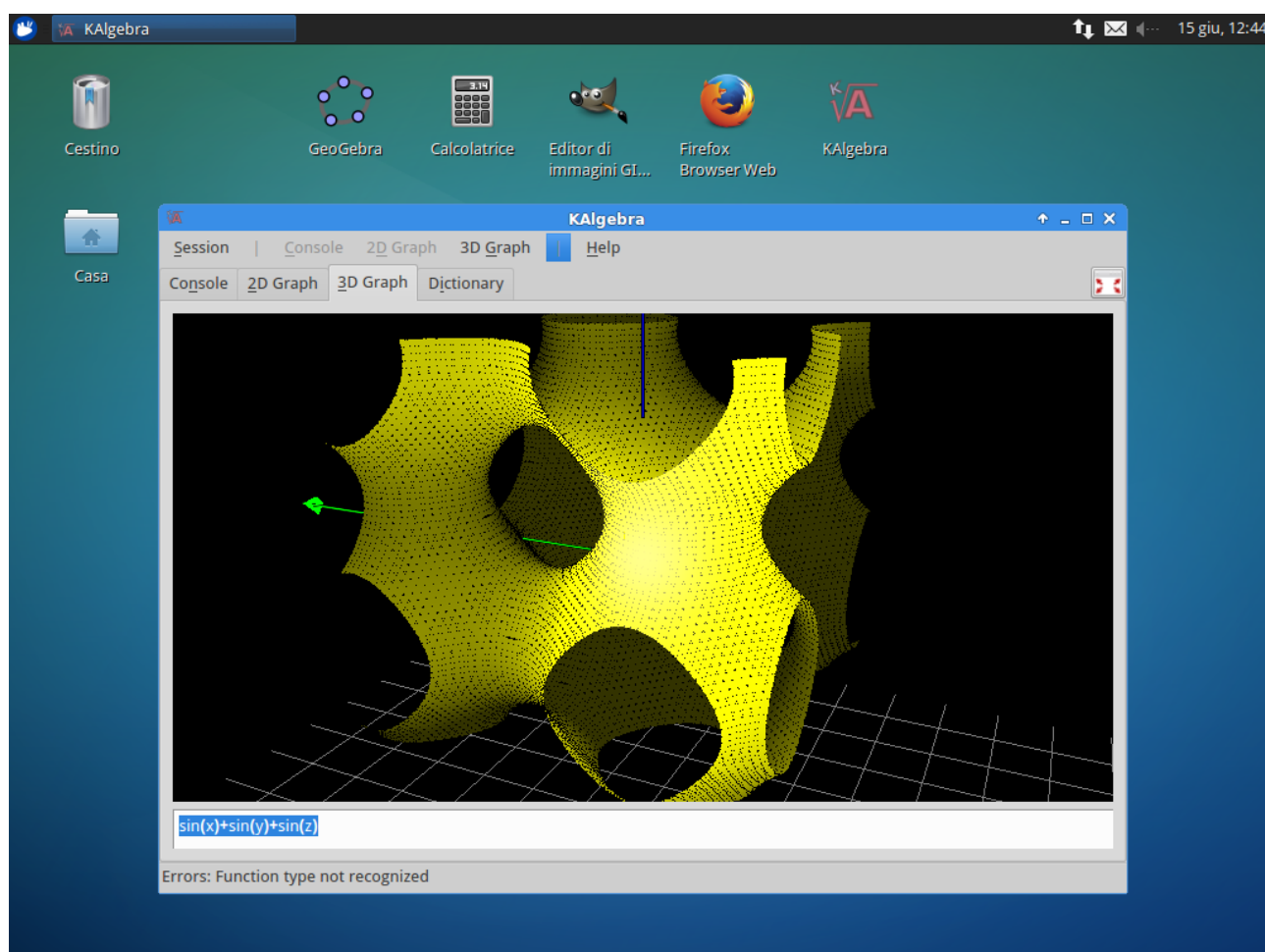


Illustrazione 16: Una simulazione dell'utilizzo di un programma algebrico chiamato **KAlgebra**, utilizzando un PC di Lab Fisica 1 avviato in rete LTSP

Installazione

Per l'installazione del Linux Terminal Server è sufficiente attenersi al Wiki della documentazione ufficiale per Ubuntu del progetto LTSP. In una nota vorrei sottolineare come abbia personalmente partecipato^j al miglioramento della documentazione ufficiali in alcuni passi:

<https://help.ubuntu.com/community/UbuntuLTSP>

Note



*Illustrazione 17: I **thin** client di Lab Fisica 1, ITIS Avogadro*

Grazie al **Linux Terminal Server Project** un laboratorio originariamente inutilizzato dal punto di vista multimediale può essere facilmente riconvertito in maniera **low-cost** in un laboratorio didattico d'avanguardia, centralizzato da ogni punto di vista e **privo di qualsiasi costi di licenze**. Le possibilità di miglioramento della rete così ottenuta sono pressoché illimitate proprio grazie alla modularità del progetto in sé e grazie ai tanti contributi da parte della comunità.

Lavagne Interattive Multimediali low-cost

Spesso in una rete LTSP accade che il server di rete per motivi di risparmio sia il computer del docente stesso e che esso sia collegato ad un **video proiettore**.

Si è voluta così arricchire la tesina per parlare di un modo per trasformare un video proiettore ed una lavagna (le comuni *whiteboard*) in una potente lavagna **touch-screen** attraverso un sistema assolutamente low-cost.

Il programma che gestirà il tutto oltre ad essere software libero è anche disponibile in altre versioni per Microsoft Windows (oltre naturalmente che per **Linux**).



Illustrazione 18: Una tipica whiteboard

Strumenti necessari

Oltre alla superficie di proiezione preferibilmente bianca, al proiettore a parete e a delle apposite tende o persiane (è necessario non ci siano forti fonti di luce diretta per non interferire nei prossimi strumenti necessari), occorrono:

- 1 WiiMote (Un telecomando noto ai possessori di una Nintendo Wii in grado di intercettare i raggi infrarossi)
Costo: ~30€;
- 1 Penna ad infrarossi (*IR Pen*)
Costo: ~20€ (o di realizzazione manuale^k);
- Software *python-whiteboard*^d e *Open Sankorè*^m
Disponibili per Linux e per altri sistemi operativi;

Ed eventualmente:

- WiildOS
Un sistema operativo basato su Lubuntu/Linux orientato alla lavagna.

È inoltre consigliato che nell'aula il proiettore sia posizionato sopra alla lavagna, in modo tale che una persona possa stare a qualche decina di centimetri di distanza dalla lavagna senza coprire col proprio corpo la superficie di proiezione.

Schema di funzionamento

Il concetto di funzionamento della LIM ottica è relativamente semplice.

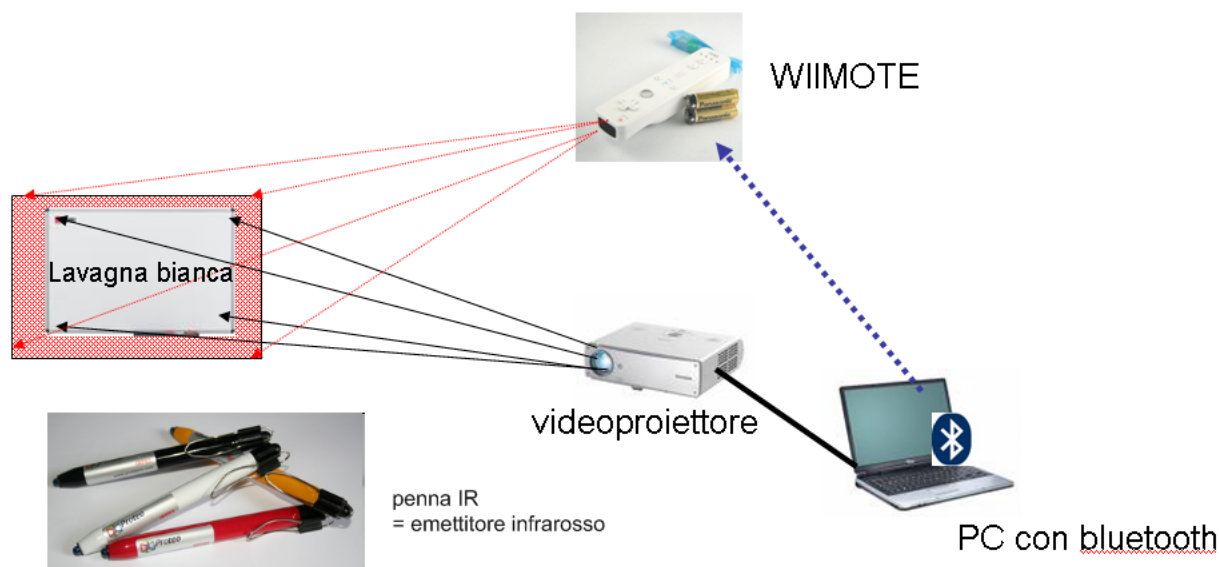


Illustrazione 19: Schema di funzionamento

1. Si collega il telecomando WiiMote al computer attraverso il Bluetooth;
2. Si posiziona il WiiMote in un punto dove possa sempre vedere direttamente la superficie di proiezione (ad esempio vicino al proiettore, o ad un lato della parete tramite un sostegno fisso);

Da questo momento il WiiMote (che percepisce gli infrarossi) è in grado di determinare se, usando la penna IR, stiamo toccando la lavagna. Toccando la lavagna con la penna IR accade che il WiiMote converte la posizione della penna in coordinate e il PC converte le coordinate in spostamenti del mouse **creando di fatto una lavagna touch-screen**.

Note di implementazione

Questo sistema rivoluzionario è ottimo per tutti quei docenti che desidererebbero utilizzare una lavagna LIM anche laddove non vi è presente la lavagna touch-screen ma dove vi è un proiettore. È però da sottolineare che questo progetto di per sé non è consigliato per essere implementato in modo fisso in un laboratorio, a causa delle piccole parti di cui è formato il progetto che potrebbero indurre a degli spiacevoli furti o noiosi guasti tecnici. Tuttavia, creando anche solo un supporto per il telecomando WiiMote in ogni aula munita di video proiettore, qualsiasi docente potrà decidere di portarsi dietro il proprio "piccolo" telecomando WiiMote ed una propria penna IR per creare una lezione più interattiva nel momento in cui desidera: Puntando il WiiMote alla lavagna e iniziando una lezione touch-screen ed evitando così gesso e pennarello.

Un registro elettronico sostenibile

Questa tesina si conclude con un accenno ad **Open Student**[®].

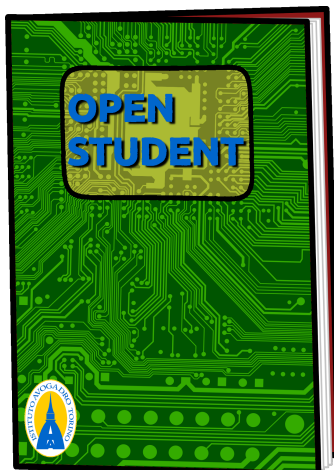


Illustrazione 20: Open Student: Registro elettronico sostenibile

Si tratta di un lavoro piuttosto chimerico che sto portando avanti dall'estate del 2013 insieme ad altri amici e che si prefigge di diventare uno dei primi (se non il primo) **registro elettronico libero** (sotto licenza GNU General Public License[®]).

Questo registro elettronico si predispone di avere tutte le caratteristiche che i "generici" registri elettronici lacuna no:

- **Essere intercompartimentale** fra dati di altri software di registri elettronici
Per lasciare totale libertà di interrompere l'utilizzo di Open Student e di optare per altre soluzioni;
- **Utilizzo online/offrine**
Quando manca la rete non deve essere un problema;
- **Open Data**
I dati inseriti devono essere esportabili dai diretti proprietari in qualunque formato: orari delle lezioni, avvisi, circolari, comunicazioni pubbliche, eventi in calendario (verifiche), eccetera. Ogni cosa deve essere esportabile tramite l'adeguata tecnologia (come per esempio il protocollo *Calda*[®] per sincronizzare gli eventi in calendario o gli orari di lezione con smarritone, tablet e PC).
- **Libretto elettronico**
Gli studenti potranno avere il libretto delle assenze e ritardi direttamente su questo registro elettronico permettendo all'Istituto di scegliere: utilizzare o meno il libretto cartaceo;
- **Design responsivo**
Il registro deve essere sempre adatto ad ogni browser mobile, tablet e desktop.

Note

Il progetto allo stato attuale non è ancora pronto per l'utilizzo.

Tuttavia è in continuo sviluppo e allo stato attuale è composto da più di **duemila righe di codice** che compongono un vero e proprio scheletro di un nuovo *Content Management System* realizzato totalmente da zero e che reggerà quello che ben presto (si auspica) diverrà **Open Student**.

Per gli interessati indico per cui i riferimenti per:

- **Schema del database:**
Vedere pagina 18
- **Codice sorgente:**
<https://launchpad.net/open-student>
- **Una pagina presentativa** sul sito dell'Hackaton Unito:
<http://www.hackunito.it/groups/open-student/>

```

v open_student.opens_1_option
e option_name : varchar(255)
e option_value : longtext
h autoload : tinyint(1)

```

```

v open_student.opens_school
e school_ID : bigint(20) unsigned
e guid : varchar(255)
h enabled : tinyint(1)
h maintenance : tinyint(1)

```

```

v open_student.opens_1_user
e user_ID : bigint(20) unsigned
e login : varchar(16)
e password : varchar(40)
e type : enum('admin','headmaster','secretary','teacher','student','family','other')
e email : varchar(64)
e name : varchar(32)
e surname : varchar(32)
e gender : enum('f','m')
e display_as : varchar(32)
e birthdate : date
e description : text
e last_log : datetime
e last_IP : varchar(64)
h login_attempt : tinyint(6) unsigned

```

```

v open_student.opens_1_family
e note : text
e user_family_ID : bigint(20) unsigned
e user_student_ID : bigint(20) unsigned

```

```

v open_student.opens_1_page
e page_ID : bigint(20) unsigned
e guid : varchar(255)
e title : varchar(255)
e status : enum('draft','published')
e content : longtext
h user_ID : bigint(20) unsigned

```

```

v open_student.opens_1_mark
e presence : enum('y','n')
e value : float(3,2) unsigned
e note : text
e test_ID : bigint(20) unsigned
e user_ID : bigint(20) unsigned

```

```

v open_student.opens_1_classroom_user
e participation : enum('y','n','expelled')
e classroom_ID : bigint(20) unsigned
e user_student_ID : bigint(20) unsigned

```

```

v open_student.opens_1_timetable
e timetable_ID : bigint(20) unsigned
e dow : tinyint(3) unsigned
e h : tinyint(3) unsigned
e classroom_ID : bigint(20) unsigned
e user_teacher_ID : bigint(20) unsigned
e subject_ID : bigint(20) unsigned

```

```

v open_student.opens_1_teacher_skill
e user_teacher_ID : bigint(20) unsigned
e subject_ID : bigint(20) unsigned

```

```

v open_student.opens_1_badnote
e badnote_ID : bigint(20) unsigned
e content : text
e date : date
h user_student_ID : bigint(20) unsigned
h subject_ID : bigint(20) unsigned

```

```

v open_student.opens_1_notice_ack
e read_date : date
e notice_ID : bigint(20) unsigned
h user_teacher_ID : bigint(20) unsigned
e classroom_ID : bigint(20) unsigned

```

```

v open_student.opens_1_teacher_presence
h user_teacher_ID : bigint(20) unsigned
e at : datetime
h timetable_ID : bigint(20) unsigned

```

```

v open_student.opens_1_student_entryexit
e student_entryexit_ID : bigint(20) unsigned
e date : date
e type : enum('entry','exit')
e h : tinyint(3) unsigned
e m : tinyint(3) unsigned
h user_student_ID : bigint(20) unsigned

```

```

v open_student.opens_1_student_absence
e student_absence_ID : bigint(20) unsigned
e from : date
e to : date
e justification : enum('none','personal','family','transport','medical','medical-certificate','other')
e note : text
h user_student_ID : bigint(20) unsigned

```

```

v open_student.opens_1_classroom
e classroom_ID : bigint(20) unsigned
e guid : varchar(128)
h year : int(1)
e name : varchar(32)
e description : text
h section_ID : bigint(20) unsigned

```

```

v open_student.opens_1_subject
e subject_ID : bigint(20) unsigned
e guid : varchar(32)
e name : varchar(32)
e description : text

```

```

v open_student.opens_1_notice
e notice_ID : bigint(20) unsigned
e guid : varchar(32)
e title : varchar(32)
e date : datetime
e content : text
e excerpt : text

```

```

v open_student.opens_1_test
e test_ID : bigint(20) unsigned
e guid : varchar(64)
e name : varchar(64)
e date : date
e type : enum('oral','written','lab')
e all : enum('y','n')
e content : text
h classroom_ID : bigint(20) unsigned
h subject_ID : bigint(20) unsigned

```

```

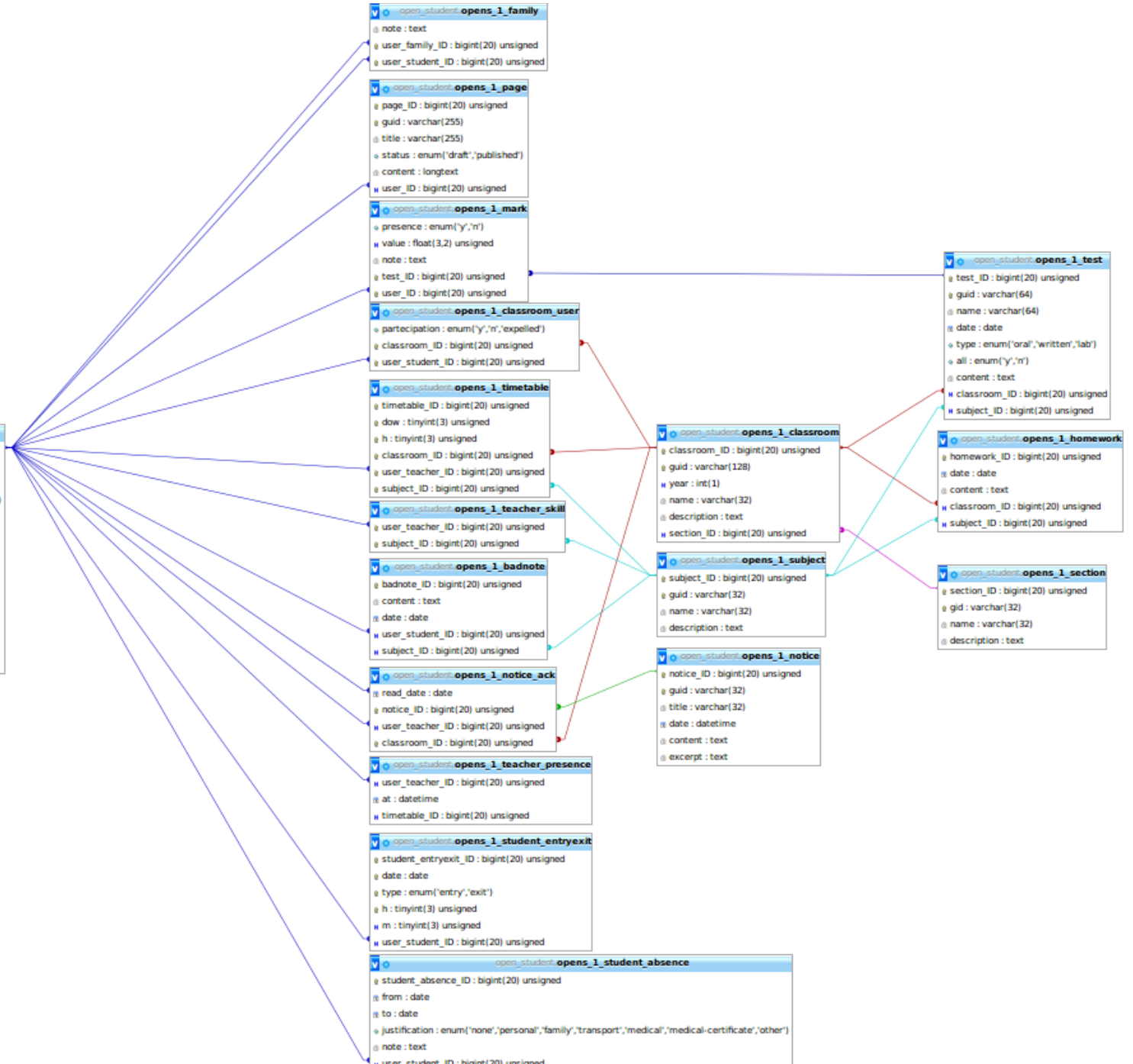
v open_student.opens_1_homework
e homework_ID : bigint(20) unsigned
e date : date
e content : text
h classroom_ID : bigint(20) unsigned
h subject_ID : bigint(20) unsigned

```

```

v open_student.opens_1_section
e section_ID : bigint(20) unsigned
e gid : varchar(32)
e name : varchar(32)
e description : text

```



Conclusioni

Grazie a diverse tecnologie open source, grazie al software libero e grazie ad un numero crescente di progetti ed iniziative sostenibili in quanto low-cost, è possibile incentivare l'utilizzo di un nuovo tipo di tecnologie più sostenibili e orientate alla didattica e ai primi utilizzatori: Docenti e studenti.

A tutti grazie per la visione di questa tesina.

Licenza

La seguente tesina è coperta da licenza *Creative Commons Attribuzione - Condividi allo stesso modo 3.0 Italia* (**CC BY-SA 3.0 IT**). Diritti e condizioni:

<http://creativecommons.org/licenses/by-sa/3.0/it/deed.it>

Per informazioni contatta l'autore, Valerio Bozzolan: <http://contatti.reyboz.it>

Riferimenti e fonti sitografiche

L'immagine di copertina è la rappresentazione grafica delle modifiche apportate ad LTSP da parte della comunità. Ogni cerchio colorato rappresenta una miglioria da parte di un utente o da una scuola che si è aggiunta al progetto principale (la retta) nel tempo.

- a Legge regionale n. 9 del 26 marzo 2009 contenente le *"Norme in materia di pluralismo informatico, sull'adozione e la diffusione del software libero e sulla portabilità dei documenti informatici nella pubblica amministrazione"*. Dall'articolo 7:
«[...] La Regione riconosce il particolare valore formativo del software libero e, nel rispetto dell'autonomia didattica delle istituzioni scolastiche, promuove forme di collaborazione per il recepimento nell'ordinamento scolastico e nei programmi didattici dei principi e del contenuto della presente legge, nell'ambito della progressiva informatizzazione dell'istruzione pubblica. [...]»
<http://arianna.consiglioregionale.piemonte.it/ariaint/TESTO%3FLAYOUT=PRESENTAZIONE&TIPODOC=LEGGI&LEGGE=9&LEGGEANNO=2009>
- b Un ringraziamento alla docente Carla Fava del Liceo Scientifico A. Avogadro di Biella che ha utilizzato un mio script per implementare più velocemente LTSP nella realtà del suo Istituto scolastico.
Profilo del docente:
<https://www.facebook.com/carla.fava.3>
Script in questione (scritto da me e rilasciato con licenza GNU GPL v3):
<https://launchpad.net/adduser-from-csv>
Installazione di LTSP nell'Istituto di Biella:
http://www.ltsp.org/stories/viewstory?story_id=556&secret=9abd13
- c Studio energetico fra thin client e PC:
http://www.athena.dk/files/userdir/documents/energy_study.pdf
- d Definizione di software libero:
<http://www.gnu.org/philosophy/free-sw.it.html>
- e Preboot Execution Environment:
https://en.wikipedia.org/wiki/Preboot_Execution_Environment
- f La formula per la stima dell'ammontare di RAM necessaria nel server mi è stata gentilmente suggerita da alcuni volontari appartenenti al team di sviluppo di LTSP, durante un dialogo personale nel loro canale IRC sul server pubblico *irc.freenode.net* nel canale *#ltsp*.
Siccome la stima non era corretta nel Wiki ufficiale, l'ho personalmente aggiornata:
<http://wiki.ltsp.org/wiki/Installation>
Link alla modifica effettuata:
<http://wiki.ltsp.org/mediawiki/index.php?title=Installation&diff=817&oldid=719>
- g Tecnologia NCQ solitamente presente nei nuovi dischi SATA:
<https://it.wikipedia.org/wiki/NCQ>
- h VirtualBox: Software open source per l'esecuzione di macchine virtuali
<https://www.virtualbox.org>
- i «XFCE4: Lightweight desktop environment for UNIX-like operating system; It aims to be fast and low on system resources, while still being visually appealing»
<http://www.xfce.org/?lang=it>
- j Contributi utente Valerio Bozzolan in documentazione LTSP
<http://wiki.ltsp.org/wiki/Special:Contributions/Valerio.bozzolan>
- k Realizzazione di una penna IR (infrarossi) utilizzando strumenti caserecci
<http://wiildos.wikispaces.com/Costruisci+una+penna+a+infrarossi+per+la+wiiLD>
- l Python-whiteboard software di gestione di lavagne touch-screen ad infrarossi
<http://wiildos.wikispaces.com/Python-whiteboard%3B+gestore+della+WiiLD>
- m Open Sankorè, The Free Interactive Whiteboard Software
<http://open-sankore.org/it>
- n Un ringraziamento a tutti i miei cari amici che stanno attualmente contribuendo per permettere al mio progetto Open Student di nascere, in particolare a:
 Marco Gagino (per il logo e per il lavoro di traduzione);
 Alberto Solaro (per idee e concepts);
 Florin Lungu(per il grande supporto in quanto membro del Direttorio del *GNU/Linux User Grup* di Napoli)
- o GNU General Public License, testo integrale
<http://www.gnu.org/copyleft/gpl.html>
- p Protocollo CalDav, specifiche RFC
<http://tools.ietf.org/html/rfc4791>