

# FermoLUG News

La Newsletter del FermoLUG

Novembre 2016 - Numero 12

## Indice

Contro il coding	1
Software Libero alla scuola "Pupilli" di Grottazzolina	3
Saldare è semplice, ecco come fare!	4

## "coding" contro "soluzioni"

E' più importante saper scrivere un programma, o saper affrontare i problemi, e quindi risolverli trovando sempre nuove soluzioni? L'articolo che apre questo numero vuole farci riflettere sul nostro futuro e su ciò che mina la base della nostra società odierna.

Pagina 1

## FermoLUG di nuovo a scuola

Eccovi il resoconto dei lavori svolti dal FermoLUG, con l'aiuto del professor Federico Tordelli, nei computer della scuola secondaria "Mario Pupilli" di Grottazzolina, per la messa in opera di un aula informatica completa, usando esclusivamente Software Libero.

Pagina 3

## Autori cercasi!

Se ti piace scrivere articoli e recensioni, FermoLUG News ti sta cercando! Invia il tuo materiale a:

[info@linuxfm.org](mailto:info@linuxfm.org)



Davvero il "coding" è il modo migliore di avvicinarsi all'informatica?

## Contro il coding

Tutti impazziti per il "coding", perché è *quella cosa quando fai sul serio col computer* e ti garantisce un lavoro per il futuro.

Partiamo dai fondamentali: "quella cosa quando fai sul serio col computer" un nome ce l'ha già: **si chiama capire il mondo**. Il codice esprime in un programma la misura della tua comprensione del mondo; quindi un programma che funziona è l'espressione del fatto che qualcuno ha compreso un pezzo di mondo.

Chi separa il codice dalla programmazione, e quindi dalla comprensione del mondo, lo fa per due motivi: ignoranza o malizia. La prima non è una colpa, ed è il motivo più diffuso; la seconda è una colpa, e grave.

**Primo caso: il coding degli ignoranti** - Chi parla di "coding" per ignoranza o per moda si basa su una conoscenza dell'**informatica da telefilm**: l'inevitabilmente giovane hacker bianco che pigiando tasti a raffica sventa la III guerra mondiale oppure penetra nella banca/CIA/NSA/centrale nucleare. Se è proprio un hacker moderno, riprogramma orbite di satelliti da

uno smartphone mentre cammina. Purtroppo questo è il livello di competenza digitale della nostra classe dirigente e della maggioranza dei docenti.

**L'informatica non è il codice, ma la comprensione del problema** che ne rende possibile la scrittura. Il codice, poi, ha difficoltà proprie (e per nulla banali), ma non esiste qualcosa come "scrivo un po' di codice e risolvo un problema". Prima di arrivare al codice un problema va compreso, analizzato, formalizzato, spesso con tecniche che richiedono anni di studio; dopodiché occorre cercare come quel problema è già stato risolto da altri, e se parte della loro soluzione va bene anche per noi, perché non esistono problemi nuovi e perché nel software copiare è cosa buona. Naturalmente occorre anche considerare tutti i vincoli ulteriori imposti alla nostra soluzione dal contesto in cui operiamo: tempi, costi, risorse, tecnologie disponibili, potenziali conflitti con altri programmi in uso. Poi la soluzione che vogliamo relizzare dovrà essere espressa in termini assolutamente precisi, perché è da

una macchina che cerchiamo di farci capire. E la macchina ha sempre ragione perché esegue i nostri ordini, non le nostre intenzioni.

A questo punto potremo finalmente iniziare a occuparci solo del codice, e della miriade di problemi che la sua scrittura comporta. A meno, naturalmente, che qualcuno sopra di noi nella gerarchia non abbia sentito la magica parolina “**agile**” e si senta in diritto di cambiare idea una volta a settimana, buttando alle ortiche metà del nostro lavoro.

Concentrarsi sul “codice” come se fosse un obiettivo e non un mezzo, come se fosse il punto di partenza e non quello di arrivo, è il modo peggiore di avvicinare i ragazzi all’informatica. Dà loro l’idea che le cose siano semplici quando non banali, addirittura (altro termine che odio) divertenti. Che non sia necessario concentrarsi, né informarsi, né imparare niente al di fuori di quello che le dita devono scrivere, quando chiunque abbia una minima esperienza di software sa che **dietro un minuto di digitazione c’è un’ora di studio**.

Questo **faciloneria tecnologica** è un insulto alla loro intelligenza, alla loro capacità di apprendere, all’apporto che possono dare in termini di creatività, e di invenzione, perché se i problemi sono sempre gli stessi le soluzioni vivaddio evolvono.

Suggerire che si possa scrivere codice senza sapere né curarsi di altro significa affidarsi alla tecnologia senza avere compreso che la tecnologia è meno importante del modo in cui viene applicata. Significa non voler capire che non ha senso produrre finte soluzioni che però “suonano” bene (ah, lo storytelling!) perché tanto “poi ai dettagli ci pensa chi è più tecnico”.

**Non esiste più un “livello tecnico” che possa venire tralasciato o affidato ciecamente a qualcun altro**, viviamo in una società troppo complessa.

Occorre dirci un’altra cosa: **il pigiatisti che non vede al di là del computer è l’immagine speculare del dirigente ignorante** che parla per frasi fatte e si cura esclusivamente del proprio personale potere scaricando le responsabilità sulle spalle degli altri. Entrambi i personaggi hanno fatto abbastanza danni, e di sicuro hanno fatto il loro tem-

po.

La programmazione (non la sola sottoattività di *coding*) è **il quarto pilastro dell’istruzione** dopo la lettura, la scrittura e la matematica.

**Secondo caso: il coding dei bastardi** - Dicevamo che l’ignoranza non è una colpa, ma la malizia sì. L’insistenza a parlare di “coding” come se la cultura del digitale si riducesse alla manualità dello scrivere codice a quello è anche propria di chi vuole produrre pigiatisti a cottimo, lavoratori a bassa specializzazione e basso costo, intercambiabili. Un proletariato del XXI secolo così impegnato a lavorare per vivere da non potersi accorgere che **il proprio lavoro è tenuto artificialmente in vita solo come pacificatore sociale**.

Il che ci porta al lavoro del futuro: siamo di fronte a un cambiamento a cui non siamo pronti: **la fine del lavoro** o, meglio, la fine della cultura del lavoro come obbligo per la sopravvivenza. Riconoscerlo significherebbe mettere in discussione i sacri assiomi del nostro modello economico, e questo non si può fare.

Ogni rivoluzione tecnologica ha modificato profondamente il carattere del lavoro, ma fino ad oggi per ogni compito superato se ne creava uno nuovo a quasi parità di competenze. Quindi se l’altro ieri lavoravi nei campi, ieri potevi lavorare in fabbrica senza particolari necessità di riconvertirti. Scrivevi lettere commerciali, tenevi la partita doppia a mano e poi imparicchiavi a fare lo stesso con Word e Excel o un gestionale. Poca roba, nessuno era veramente escluso.

Ma ora il cambiamento è tangibile. **I robot possono fare la maggior parte dei lavori manuali o a bassa specializzazione, in modo economicamente vantaggioso, oggi**. Per quanto riguarda i lavori intellettuali la sommarizzazione (<http://www.theverge.com/2015/1/29/7939067/a-p-journalism-automation-robots-financial-reporting>) e la produzione di notizie (<http://www.bbc.com/news/technology-26614051>), la consulenza legale (<http://www.rossintelligence.com/>), buona parte della diagnosi (<http://www.ibm.com/smarterplanet/us/en/ibmwatson/clinical-trial-matching.html>) clinica ([plain\) in medicina, l’investimento \(<http://www.techeconomy.it/2016/03/16/www.nytimes.com/2016/02/28/magazine/the-robots-are-coming-for-wall-street.html>\) in Borsa \(<http://www.bbc.com/news/technology-27426942>\) sono già da tempo terra di conquista per il software. E quindi anche molti lavori come giornalista, avvocato medico generico e trader sono morti che camminano](http://www.mghlcs.org/projects/dx</a></p></div><div data-bbox=)

(<http://www.theguardian.com/technology/2014/jun/15/robot-doctors-online-lawyers-automated-architects-future-professions-jobs-technology>) già adesso. E il futuro arriva ogni mattina fresco come una rosa.

Per contro, le **competenze** del digitale richiedono **anni di formazione** e apprendistato, un aggiornamento continuo e non sono per tutti. Non certamente per chi aveva un lavoro prevalentemente manuale a bassa specializzazione o per chi vivacchiava usando il computer come macchina per scrivere. Non certamente per chi crede che la formazione sia quella cosa che a sedici anni puoi dire di averne abbastanza. Non basta: anche i lavori del digitale stanno diminuendo vertiginosamente, perché nessun settore è immune dall’automazione. L’intera infrastruttura IT di **Google** per il Nord America richiede poco più di una **ventina** di tecnici. Foxconn rimpiazza operai cinesi (famosi per il costo della manodopera) con un **milione** di robot che costeranno ancora meno e lavoreranno ogni secondo della loro (lunghissima) vita meccanica. Amazon, il grande magazzino del mondo, assume in un colpo solo **diecimila** magazzinieri. Ma sono tutti robot (<http://www.tomshw.it/mobile/cont/news/amazon-assume-10-000-lavoratori-ma-sono-tutti-robot/56519/1.html>).

La cosa problematica non è tanto che il lavoro sta sparendo, ma che l’economia preme per eliminare anche tutti quei lavori che sono sopravvissuti per ignoranza, inerzia, lassismo, perché tutto sommato era politicamente e aziendalmente più facile mantenere le persone nell’illusione che ricevere uno stipendio per qualcosa che una macchina poteva fare meglio potesse costituire un ragionevole progetto di vita. Non c’è nessun futuro come barista ([Pagina 2](http://www.popularmecha-</a></p></div><div data-bbox=)

nics.com/technology/robots/a15242/robot-baristas-cornell-coffee/), casiere (<http://www.theguardian.com/world/2015/feb/04/japanese-bank-introduces-robot-workers-to-deal-with-customers-in-branches>), cameriere ([http://www.chinadaily.com.cn/business/2014-11/26/content\\_18978177.htm](http://www.chinadaily.com.cn/business/2014-11/26/content_18978177.htm)), tassisti (<http://qz.com/538758/japanis-building-a-robot-taxi-service-with-thousands-planned-for-the-2020-olympics/>), camionista (<http://www.informationweek.com/mobile/mobile-business/first-automated-truck-licensed-to-operate-on-public-roads/d/d-id/1320311>). E

sto parlando della situazione oggi. Vedrete fra due anni.

Oggi, i robot fanno il magazziniere, il meccanico, l'assemblatore di circuiti. L'intelligenza artificiale pilota aerei e treni, cura la logistica, decide le politiche dei prezzi, gestisce fondi speculativi e il servizio clienti online del tuo venditore online di fiducia. E in tutto questo c'è chi viene a parlare di "coding" come se spataccare codice a casaccio fosse la Via per chissà quali magnifici destini.

Abbiamo voluto la tecnologia perché ci liberasse prima dalla fame e poi dalla fatica. Oggi sta per liberarci dall'idea del lavoro come bi-

sogno universale per la sopravvivenza. Dobbiamo decidere cosa fare del nostro tempo, perché ne avremo sempre di più, mentre di lavoro ce ne sarà sempre meno, e per sempre meno persone. Ma la nostra cultura si basa ancora sul bisogno del bisogno. È tempo di ripensarla. L'economia di mercato come l'abbiamo conosciuta finora ci ha portato fino qui, ma non può portarci oltre.

Walter Vannini - (foto Rainer Stropek, CC-BY 2.0) - Fonte: <http://www.techeconomy.it/2016/03/16/contro-il-coding>

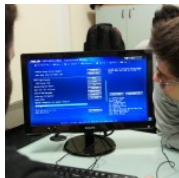
## Software Libero alla scuola "Pupilli" di Grottazzolina

A cura di Marco Alici (<https://twitter.com/MarcoAlici>)

Gli ingredienti sono semplici: una scuola con un'aula computer quasi inutilizzabile; docenti consapevoli delle opportunità didattiche derivanti dall'utilizzo delle nuove tecnologie digitali e pronti a mettersi in gioco; un LUG ([https://it.wikipedia.org/wiki/Linux\\_User\\_Group](https://it.wikipedia.org/wiki/Linux_User_Group)) (Gruppo di Utenti Linux) attivo e disponibile a dare una mano. Il risultato è un'aula informatica finalmente fruibile, di facile manutenzione e di facile utilizzo per insegnanti e ragazzi. A costo zero.

La scuola è la secondaria "Mario Pupilli" di Grottazzolina (<http://www.iscmontegiorgio.it/it/scuola-secondaria-di-1-grado-pupilli-di-grottazzolina/>). Nella sua aula computer convivono alcuni portatili acquistati di recente insieme a computer da scrivania vecchi anche di dieci anni, alcuni dei quali mal funzionanti, dotati di versioni diverse (in alcuni casi obsolete) sia del sistema operativo Microsoft Windows che dei programmi di produttività individuale (la suite Microsoft Office, per intenderci), alcuni dei quali in condizioni di licenza difficili da verificare. Una situazione troppo eterogenea, poco funzionale ed eticamente poco sostenibile per una scuola: praticamente inutilizzabile. Un peccato, perché il corpo docente è invece convinto dell'utilità delle tecnologie digitali nella didattica, della necessità di rispettare le leggi sull'utilizzo del software e della difficoltà di reperire risorse per migliorare la situazione.

In questo caso sono risultate fondamentali le intuizioni di un insegnante, il professor Federico Tordelli. Prima intuizione: considerare il mondo del software libero, che può essere scaricato ed installato liberamente su qualsiasi computer, come opzione. L'idea è di eliminare le diverse versioni di Windows ed installare un sistema operativo Linux su tutti i computer, per avere un parco macchine omogeneo, performante e virtualmente immune da virus, e di poter usare un'unica versione dell'ottima suite LibreOffice, che insegnanti e studenti possono eventualmente installare anche sui propri computer, indipendentemente dal sistema operativo, insieme a decine di altri ottimi programmi open source utili per la didattica. Seconda intuizione: chiedere aiuto all'associazione FermoLUG, che riunisce gli utenti di Linux del fermano, svolge da anni attività di volontariato nell'ambito della promozione del software libero ed ha già lavorato su progetti analoghi in altre scuole della zona.



**FASE 1** - Dopo un sopralluogo iniziale, per capire la situazione e pianificare l'intervento, Tordelli e alcuni soci del LUG si sono messi subito al lavoro. Le macchine vecchie o rotte sono state eliminate, recuperando hardware (soprattutto schede di memoria RAM) da utilizzare per aumentare le prestazioni delle altre. Attingendo alle risorse del progetto "Trashware", con il quale il FermoLUG

recupera macchine dismesse da aziende e enti pubblici e le dona a chi ne fa richiesta dopo averci installato Linux, un paio di computer rotti sono stati sostituiti con altri simili. Quindi si è proceduto ad installare Linux, precisamente la distribuzione Xubuntu (<http://www.ubuntu-it.org/derivate/xubuntu>), con alcune personalizzazioni che ne migliorano la facilità di utilizzo e l'estetica, su tutti i computer dell'aula e anche su quello della lavagna interattiva multimediale (LIM). Xubuntu ha il vantaggio di essere abbastanza completa per un'esperienza utente facile e soddisfacente, e nello stesso tempo sufficientemente leggera da poter girare anche su computer con risorse hardware limitate: un ottimo compromesso, dunque, per il parco macchine della scuola "Pupilli".

Sul computer del docente si è installato l'ottimo Epopotes (<http://www.epoptes.org/>), programma libero che permette all'insegnante di controllare i computer degli studenti dalla propria postazione e di interagire con essi in maniera semplice e intuitiva.

Nel giro di qualche serata – tutto si è svolto, necessariamente, fuori dall'orario scolastico e dall'orario di lavoro delle persone impegnate nell'operazione – si è riusciti a rimettere in piedi la struttura, che oggi conta una dozzina di computer identici dal punto di vista della dotazione software, connessi in rete, facili da usare, ricchi di programmi di ottimo livello, sicuri sia per quanto riguarda i virus (per sua na-

tura un sistema Linux è molto sicuro e robusto), sia per quanto riguarda la navigazione sul web: infatti con l'occasione si è pensato di dotare la rete di un sistema di controllo – anch'esso libero – che impedisce l'accesso a siti web pericolosi o non adatti alla fascia di età degli utenti. Non sfugga il fatto che, trattandosi di software libero, anche i ragazzi e i docenti potrebbero installare – legalmente e senza costi di licenza – lo stesso sistema operativo sui propri computer, o al limite molti dei programmi presenti (ad esempio LibreOffice), che sono multi-piattaforma e quindi installabili anche su altri sistemi operativi, ed avere quindi a casa un ambiente di lavoro identico a quello che trovano a scuola.

**FASE 2** - Naturalmente gli strumenti da soli non bastano: occorre formare le persone che quegli strumenti devono

## Saldare è semplice, ecco come fare!

Vuoi imparare a saldare?  
Vuoi realizzare delle ottime saldature?  
Vuoi insegnare ad altri come saldare?  
Sono felice di annunciare che ho realizzato la traduzione in italiano di:  
Soldering is Easy (Saldare è semplice) ([http://www.linuxfm.org/wp-content/uploads/2016/05/FullSolderComic\\_IT.pdf](http://www.linuxfm.org/wp-content/uploads/2016/05/FullSolderComic_IT.pdf))  
(seguite il link per prelevare la versione in italiano)  
un fumetto che insegnerà a chiunque le basi della saldatura. Sette pagine che spiegano in dettaglio come fare una buona saldatura anche a chi non ha mai saldato! Il lavoro originale in lingua inglese è stato curato da:  
- narrazione di Mitch Altman (<http://cornfieldelectronics.com/>)  
- adattamento fumetti Andie Nord-

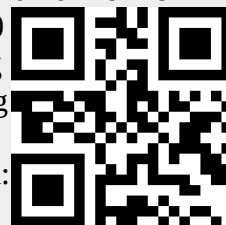
gren (http://log.andie.se/) - a cura di Jeff Keyzer (<http://mightyohm.com/>)  
Pagina di riferimento: <http://mightyohm.com/soldercomic>  
Il fumetto in inglese (ed altri argomenti molto interessanti) sarà incluso nel libro: **How to Make Cool Things with Microcontrollers (For People Who Know Nothing)** di prossima pubblicazione. Il libro è stato realizzato da Mitch Altman e Jeff Keyzer edito da No Starch Press.  
Il fumetto è rilasciato sotto licenza Creative Commons (Attribution-ShareAlike), quindi si è liberi di insegnare con questo fumetto, tradurlo, usarlo, diffonderlo, colorarlo ed è fondamentalemente fare tutto ciò che vi pare!

Antonio Faccioli - Fonte: <http://www.libreitalia.it/software-libero-alla-scuola-pupilli-di-grot-tazzolina/>



Michele Maffucci - Fonte: <http://www.maffucci.it/2011/12/28/saldare-e-semplce-ecco-come-fare/>

Associazione Culturale  
Fermo Linux Users Group  
Gruppo Utenti Linux di Fermo  
C.F.90037220440  
[www.linuxfm.org](http://www.linuxfm.org)  
[info@linuxfm.org](mailto:info@linuxfm.org)



Gruppo Telegram:  
[bit.ly/fermolug](https://bit.ly/fermolug)

Mailinglist pubblica:

<http://liste.linuxfm.org/mailman/listinfo/discussioni>

Il FermoLUG nasce nel 2003 da un gruppo di amici con la voglia di condividere le proprie scoperte in ambito informatico.

Lo scopo principale dell'Associazione è quello di promuovere e diffondere il Software Libero facendo corsi di formazione, eventi aperti a tutti e tenendo attiva e legata la propria comunità di soci e simpatizzanti.

Se hai voglia di condividere idee, trucchi e soluzioni nell'uso quotidiano di GNU/Linux, inserisciti nella Mailing List: è un sistema facile e veloce per entrare direttamente in contatto con i membri del LUG!

Se desideri aiutarci attivamente nella nostra missione, iscrivendoti ufficialmente alla nostra associazione, clicca su "Diventa Socio" dal nostro sito web [www.linuxfm.org](http://www.linuxfm.org).

Il costo dell'iscrizione è di 10€.

Licenza applicata a questo numero:  
Attribuzione - Condividi allo stesso modo 3.0 Italia (CC BY-SA 3.0 IT) salvo ove indicato  
<http://creativecommons.org/licenses/by-sa/3.0/it/>