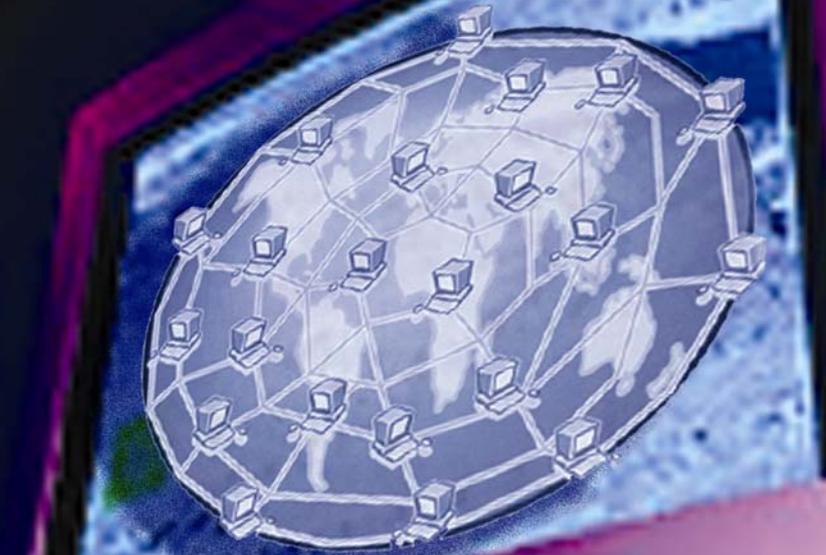


Università
degli Studi di **Palermo**



Facoltà di Scienze MM. FF. NN.

Corso di Laurea in: Informatica



**On-Lab: un ambiente
web per lo sviluppo di
laboratori on line**

**Università
degli Studi di Palermo**



Facoltà di Scienze MM. FF. NN.

Corso di Laurea in Informatica

On-Lab: un ambiente web per lo sviluppo di laboratori on line

Tesi di:
Rocco Versace

Matricola:
0401532

Relatore:
Prof. Marco Alfano

Correlatore:
Prof. Biagio Lenzitti

INTRODUZIONE	3
1. LABORATORIO E DIDATTICA	6
1.1 RAPPORTO TRA DIDATTICA E LABORATORIO	6
Specificità del metodo.....	6
I limiti dell'insegnamento tradizionale	7
Rapporto insegnamento/apprendimento.....	8
Didattica di Laboratorio: una definizione.....	8
L'apprendimento significativo.....	9
1.2 LA DIDATTICA ON LINE	9
1.3 IL LABORATORIO E LA DIDATTICA ON LINE	12
1.4 ESEMPI DI LABORATORI ON LINE	13
Esempi di laboratori on line per la gestione di apparecchiature fisiche e virtuali.....	13
Esempi di esercitazioni di laboratorio	18
2. STRUMENTI PER LA CREAZIONE DI LABORATORI ED ESERCITAZIONI ON LINE. 21	
2.1 ESEMPI DI STRUMENTI PER LA CREAZIONE DI LABORATORI ON-LINE	21
2.2 ESEMPI DI STRUMENTI PER LA CREAZIONE DI ESERCITAZIONI ON-LINE	23
2.3 VANTAGGI E SVANTAGGI DEGLI STRUMENTI ANALIZZATI	37
3. ON-LAB: UN AMBIENTE WEB PER LO SVILUPPO DI LABORATORI ON LINE..... 40	
3.1 OBIETTIVI E CARATTERISTICHE DI BASE	40
3.2 AMBIENTE DI SVILUPPO.....	41
3.3 PROGETTAZIONE ED IMPLEMENTAZIONE DI ON-LAB	47
Architettura	48
Flusso dinamico	49
Autenticazione	56
4. ESEMPI DI UTILIZZO DI ON-LAB..... 58	
CONCLUSIONI..... 73	
APPENDICE – CODICE JAVA	76
GENERATOREQUESTIONARI.JAVA.....	76
FILEIO.JAVA	82
QUESTIONARI.JAVA	86
BIBLIOGRAFIA..... 88	

Introduzione

Internet e le reti telematiche sono sempre più presenti nei vari ambiti sociali e professionali ed è quindi naturale cominciare a pensare a metodologie didattiche e formative che possano sfruttare al meglio questi strumenti nonché alle nuove modalità di insegnamento e apprendimento che tali strumenti suggeriscono e rendono più facilmente attuabili.

Allo stesso tempo la didattica tradizionale, basata principalmente su insegnamenti teorici, non è più sufficiente a fornire allo studente una formazione completa e tutti gli stimoli all'apprendimento di cui necessita. È bene, quindi, affiancarla a metodi più moderni di apprendimento (didattica laboratoriale) per rendere le forme di insegnamento sempre più vicine alle modalità e ai ritmi di apprendimento degli allievi, in special modo in questo momento in cui il baricentro dell'azione educativa e didattica della scuola si sposta dall'insegnamento all'apprendimento, dal "sapere" al "saper fare" dall' "apprendere" al "far apprendere ad apprendere" e la mediazione didattica deve intrecciarsi con l'operatività degli allievi.

In rete si trovano milioni di siti informativi che offrono dispense, manuali, guide su ogni tipo di argomento. Questi siti non fanno altro che mettere informazioni, normalmente consultabili su supporti cartacei, a disposizione di qualunque navigatore da qualsiasi parte del mondo, garantendo l'accesso alla stessa risorsa a centinaia di persone contemporaneamente. Questo è il modo più semplice per rendersi conto di quanto possa essere utile questa grande rete mondiale per la didattica e l'informazione.

Esistono centinaia di siti che mettono a disposizione corsi on line, alcuni affiancati da un tutor che interagisce con i corsisti, consentendo di creare un' "aula virtuale" frequentata da persone residenti in diverse parti del mondo. Alcuni siti consentono inoltre di interagire con strumenti ed apparecchiature, quali, ad esempio, oscilloscopi, robot e computer. Questi strumenti sono interamente controllabili da un computer remoto tramite la rete e spesso sono anche monitorati da webcam, per rendersi

effettivamente conto degli effetti prodotti dai comandi digitati. Da quest'ultimo esempio nasce il concetto di "Laboratorio On Line".

Nell'ambito dell'insegnamento, il "laboratorio" non deve essere considerato unicamente come un luogo in cui si elaborano e sperimentano saperi, ma anche un insieme di opportunità formative per produrre nuove conoscenze e sviluppare nuove competenze, un luogo di confronto, relazione, amicizia, condivisione, che permette di dare risposta ai bisogni formativi ed esperienziali degli studenti. Infatti, la didattica laboratoriale integra i saperi disciplinari con le competenze che ciascun allievo acquisisce per effetto dell'esperienza di apprendimento in laboratorio, spinge gli alunni a mettere le loro potenzialità personali al servizio di tanti, li aiuta a fare del lavoro di gruppo un momento di crescita, coesiste con la didattica tradizionale, presuppone un costante controllo e feedback. Essa si rivela un itinerario di lavoro significativo per l'allievo, capace di arricchire i suoi orizzonti, senza peraltro trascurare l'insegnamento delle conoscenze e delle abilità disciplinari dovute [1].

In un laboratorio è quindi possibile costruire situazioni di apprendimento in cui l'attività operativa dell'alunno (ad esempio, le risposte alle domande dell'esercitazione) serve per costruire competenze, conoscenze, abilità richieste, oppure per consolidare competenze, concetti, schemi, ecc. Infatti, l'operatività laboratoriale conduce l'allievo ad usare le varie conoscenze, competenze, abilità disciplinari in contesti spesso diversi da quelli in cui li ha appresi e, molto probabilmente, da quelli in cui li riutilizzerà. Questa attività laboratoriale, inoltre, non è del tutto staccata da quella teorica, alla quale, anzi, è strettamente legata, nel senso che la verifica dell'apprendimento di alcuni moduli, disciplinari e/o interdisciplinari, può essere organizzata sotto forma di laboratorio, proprio per la funzionalità che tale attività formativa ha nello sviluppo e nel consolidamento di conoscenze e competenze utili al cittadino e al professionista di oggi.

Nell'ambito del lavoro descritto qui di seguito si sono considerati numerosi esempi di laboratori on-line (reali o virtuali) e si sono analizzati gli strumenti disponibili per la realizzazione di tali laboratori. Da tale analisi è risultato che non esistono attualmente strumenti per la creazione di questi laboratori che da un lato permettano di interfacciarsi alle apparecchiature (reali o virtuali) per eseguire comandi su di esse e riceverne (e gestirne) i relativi output e dall'altro consentano al docente di creare

facilmente delle esercitazioni di laboratorio su queste apparecchiature e allo studente di poterle eseguire on-line creando, al contempo, percorsi sperimentali personalizzati.

Per un docente che non possiede specifiche competenze tecniche, risulta praticamente impossibile creare un laboratorio del genere e, quindi, per un'istituzione scolastica mettere a disposizione dei docenti questo genere di servizi, comporterebbe da un lato andare incontro a ingenti spese e dall'altro acquisire strumenti comunque difficili da usare e anche poco versatili.

A fronte delle problematiche discusse precedentemente, il lavoro che viene presentato qui propone una soluzione semplice, immediata e alla portata di tutti che consente ai docenti di realizzare senza grande fatica questo tipo di laboratori e agli studenti di avere accesso a questi laboratori sempre e dovunque nonché di creare i propri percorsi sperimentali. La soluzione presentata prende il nome di On-Lab ed è una piattaforma (sviluppata in Java) che consente di creare in pochi passi un'interfaccia di strumento (reale o virtuale), con la quale verranno poi create delle esercitazioni on-line da proporre allo studente.

Il lavoro qui descritto qui di seguito è strutturato come segue:

Nel primo capitolo, si discutono i rapporti che legano la didattica al laboratorio, analizzando, in primo luogo, le problematiche legate all'insegnamento teorico, per poi osservarne le differenze quando si passa all'utilizzazione di ambienti "on line".

Nel secondo capitolo si analizzano alcuni strumenti che consentono la creazione di laboratori ed esercitazioni on line, evidenziandone le caratteristiche e analizzandone i vantaggi e gli svantaggi.

Nel terzo capitolo viene descritta la soluzione On-lab, discutendo in particolare gli obiettivi, l'architettura e gli aspetti implementativi.

Nel quarto capitolo, infine, vengono presentati due esempi concreti di utilizzo di On-Lab, allo scopo di dimostrarne le potenzialità e facilità d'uso. Viene mostrato come On-Lab si comporta nella creazione delle interfacce (un router fisico ed un'interfaccia di funzioni matematiche), nell'impostazione di un'esercitazione e nella sua esecuzione.

Si noti che nomi e marchi citati nel testo sono generalmente depositati o registrati dalle rispettive case produttrici.

1. Laboratorio e Didattica

Il laboratorio è un luogo, uno spazio fisico, dove si tocca, si manipola, si prova e si riprova, si discute e ci si confronta, si vive insomma un'esperienza reale, concreta, coinvolgente, arricchente. Il laboratorio, trasferito in un contesto di didattica assistita dalla telematica, trova posto non più in uno spazio reale, ma in uno spazio virtuale riprodotto dalle tecnologie di rete. Il laboratorio diventa così un laboratorio virtuale.

In un laboratorio virtuale lo studente non interagisce con la realtà ma con un modello della realtà, cioè interagisce con una rappresentazione astratta della realtà che diventa il suo ambiente di apprendimento. Un ambiente di apprendimento, proprio perché strumento di apprendimento, deve orientare il discente ad una continua assegnazione di significato alla sua esperienza, facendo sì che la navigazione proceda di pari passo con la crescita della competenza e della consapevolezza di essa da parte del soggetto che apprende. L'ambiente deve mettere tra i suoi obiettivi principali lo sviluppo di una sensibilità all'autovalutazione delle proprie azioni e competenze, e all'autoverifica del raggiungimento degli obiettivi formativi che si prefigge. In molti ambienti di apprendimento oggi disponibili sul Web, siano essi basati su architetture ipertestuali o su ambienti tridimensionali, questo requisito viene ignorato o rimane del tutto implicito: la possibilità di spostarsi dappertutto anche senza capire nulla di ciò che si sta facendo contraddistingue molta dell'offerta corrente.

1.1 Rapporto tra didattica e laboratorio

Specificità del metodo

Il modello della didattica dei laboratori è rintracciabile soprattutto (ma non certo esclusivamente) nella pedagogia dell'attivismo di **Dewey e Kilpatrick**.

In particolare sono tre i principi che tale modello coltiva e che risultano di particolare interesse:

- gli scopi dell'educazione vanno fondati sui bisogni intrinseci del soggetto che apprende;
- la cooperazione contribuisce efficacemente a “liberare e organizzare” le capacità di chi apprende e a trasformarle in competenze;
- la valenza educativa delle attività sta nelle connessioni e nella flessibilità di percorsi riconosciuti dall'alunno come significativi per sé e spendibili nel compito di intervenire sulla realtà.

Il punto di partenza dell'azione educativa è lo sviluppo della persona nelle sue capacità individuali e sociali per metterla in condizione di pensare bene ed agire con autonomia di giudizio, permettendo positivi e fecondi rapporti di collaborazione e di identificazione con gli altri. Partendo dagli interessi della persona e dalla sua esperienza è più agevole realizzare questo processo perché ogni individuo impara più facilmente ciò che vive in una condizione di collaborazione con gli altri e di accettazione dell'ambiente. In questa situazione educativa, le attività (pratiche, tecniche, intellettuali ed affettive) sono intimamente congiunte. La Didattica dei Laboratori assume una funzione importante nella scuola come attività intenzionale per promuovere gli apprendimenti in libera cooperazione con gli altri individui. La sua particolarità sta nel proposito di dar vita ad una strategia o ad un piano da concretizzare attraverso azioni organizzate. Dalla situazione problematica scaturisce un processo dinamico e costruttivo in cui l'alunno viene sostenuto dall'insegnante che lo indirizza, lo sollecita alla scoperta dei percorsi possibili, lo sostiene nella fatica di affrontare le difficoltà. Il progetto è una delle modalità possibili per realizzare il Laboratorio [2].

I limiti dell'insegnamento tradizionale

Molti docenti rilevano come il maggior limite dell'insegnamento tradizionale sia l'essere ancorato ad una sola ottica, ad un solo “metodo” d'insegnamento basato

prevalentemente sui libri di testo, ad una sola modalità di relazione docente/alunno. Altro limite forte è mantenere come unico ground organizzativo la classe, costituita da tanti individui con capacità, conoscenze, abilità, interessi e stili di apprendimento diversi. La presentazione dei contenuti solitamente avviene secondo una **programmazione** didattica sistematica e strutturata, tutta compiuta a priori dal docente. L'insegnante spiega un argomento o una regola ritenuta da lui di grande importanza, lo studente spesso non riesce a cogliere il nesso con la propria esperienza, per cui la scuola rimane staccata dalla vita e l'attenzione per la materia, quando c'è, è dettata dal senso del dovere.

Rapporto insegnamento/apprendimento

Sotto la spinta di un vivo interesse, gli alunni sono indotti a superare le difficoltà, a ricercare e a trovare un nesso tra le conoscenze, le abilità disciplinari e la vita, il loro modo di essere e di vivere. L'aula assume l'aspetto di un Laboratorio dove si pone l'accento non solo sulle conoscenze dichiarative (ciò che si sa), ma anche su quelle procedurali (come si fa) e metacognitive (riflettere sulla propria esperienza cognitiva). I docenti sono alla continua ricerca e sperimentazione di nuove strategie didattiche ed educative che contemplino un rapporto insegnamento/apprendimento interattivo, interdisciplinare e cooperativo. L'allievo nella scuola laboratoriale è **costruttore** del proprio apprendimento: associando attività pratiche e intellettuali, è incoraggiato dal docente a conquistare un atteggiamento autonomo e responsabile.

Didattica di Laboratorio: una definizione

Laboratorio e progetto sono due termini che si riferiscono ad un modo attivo ed operativo di affrontare l'insegnamento/apprendimento. Tuttavia il primo ingloba anche il secondo nel senso che una attività laboratoriale può essere svolta come progetto disciplinare, interdisciplinare, transdisciplinare, come ricerca, come sperimentazione e osservazione diretta di fenomeni su cui poi si innesta una fase di riflessione metacognitiva. La Didattica dei Laboratori viene ipotizzata nella fase di previsione, sulla base di scenari, ma spesso è necessario modificarla in itinere per rispondere in maniera concreta alle esigenze di adattamento alla situazione reale e

alle esigenze dei discenti, per accogliere con apertura e sensibilità quanto proviene dal territorio inteso come risorsa in una continua operazione di interscambio con la scuola.

L'apprendimento significativo

Nella didattica, il termine Laboratorio, viene usato per indicare qualsiasi attività intenzionale tesa a raggiungere un risultato definito e concreto, attraverso una serie di procedure e di attività specifiche controllate dall'allievo e per lui significative [2]. Gli alunni si trovano a ragionare, a confrontarsi su compiti reali. Anche quando sono necessari percorsi didattici volti a dare sistematicità disciplinare agli apprendimenti o ad esercitare specifiche abilità, queste vanno inserite in un contesto che porti l'alunno a "dare senso" a quello che sta facendo perché ne sperimenta le possibilità applicative in un contesto vitale.

Anche in questo caso l'insegnante può attivare l'interesse degli alunni costruendo insieme a loro un percorso che preveda un prodotto finale, sintesi di varie fasi di lavoro.

È compito del docente stabilire i criteri e le prove di valutazione sulla base dei risultati attesi. In questo caso un metro di giudizio adeguato ad un "lavoro autentico" non può essere rappresentato solo dalle prove tradizionali. Un tipo di valutazione che consideri sia il processo che il prodotto finale di un percorso, fornisce informazioni sui progressi conseguiti dallo studente, su ciò che ha imparato e sui motivi che rendono rilevante l'apprendimento.

1.2 La didattica on line

È sufficiente eseguire una ricerca sul web per comprendere quanto in fretta si stanno moltiplicando i siti che offrono corsi on line. Dalle università, che vanno sempre più proponendo soluzioni alternative agli studenti fuori sede, alle associazioni di formatori che cercano di ampliare il proprio mercato, alle società di servizi su Internet, che estendono la propria offerta alla formazione, al Ministero dell'Istruzione dell'Università e della Ricerca (MIUR) che ha formato con strumenti di Formazione a Distanza (FaD) oltre 200.000 insegnanti, dirigenti e personale non docente. Qualche

tentativo volto alla creazione di modelli di Formazione On Line (FOL) in Italia si è rivelato a volte deludente. I cosiddetti "corsi on line" sono spesso semplici raccolte di dispense in formato DOC, PPT o PDF da scaricare, stampare e leggere esattamente come un libro, insieme a test a scelta multipla da eseguire. Spesso l'intero corso on line potrebbe essere visto semplicemente come un corso per corrispondenza, con l'unica differenza che viene trasmesso via Internet. Oppure vengono presentate come piattaforme (sistemi per la gestione di più servizi) per la FaD raccolte di strumenti di comunicazione (Forum, Chat, messaggeria), che permettono sicuramente una collaborazione a distanza tra studenti, ma i cui prodotti raramente vengono rimessi in circolo come documenti del corso. [3]

Un ambiente digitale per l'apprendimento non deve essere (solo) una scuola on line, ma è necessario che sia:

- Un luogo dove vengono raccolte le conoscenze di un gruppo in modo che possano essere filtrate, trovate, analizzate da chiunque sia interessato.
- Un luogo di (simulazione) di richieste e fornitura di supporto tra colleghi, in cui sia possibile mettere in comune le esperienze delle diverse attività educative e didattiche.
- Un luogo di costruzione di nuove conoscenze che permetta di scoprire in maniera cooperativa la complessità del reale non separando teoria, tecnica e pratica .

Una formazione online significativa deve soddisfare ad alcune caratteristiche di base, cioè essere **reticolare, dinamica, interattiva e cooperativa**. [4]

RETICOLARE

Per ottimizzare la produzione, occorre che le stesse informazioni siano fruibili da utenti diversi. D'altra parte, per personalizzare il corso, deve essere possibile l'attivazione di link diversi da uno stesso nodo¹. La struttura deve quindi essere composta da unità collegate con link non statici. Questo vuol dire che il percorso può cambiare ad ogni consultazione, sia che si tratti di utenti diversi, sia che a navigare sia lo stesso utente.

¹ Nodo: Viene così definito un singolo computer o terminale facente parte di un network. L'insieme di più nodi costituisce una rete di computer.

DINAMICA

Un corso è un sottoinsieme della rete che si crea dinamicamente in base alle caratteristiche dello studente. Ogni nodo è marcato con varie proprietà (livello di difficoltà, argomento generale, tipo di corso, etc). La richiesta di un nodo da parte di uno studente (esecuzione di una ricerca, seguire un link) si traduce nella generazione da parte del database di una pagina che traduce i contenuti del nodo in base alle proprietà dello studente stesso.

INTERATTIVA

Sarebbe auspicabile che ogni nodo presenti almeno una possibile sorgente di interazione, ma questo non deve tradursi in una forzatura che allontana chi è semplicemente "in visita" e non ha ancora deciso di impegnarsi.

Nei corsi dovrebbero essere proposti almeno tre livelli di interazione, proporzionali non alla difficoltà, ma al grado di impegno e di attenzione che l'allievo decide di dedicare:

1. *navigazione*: è la modalità più semplice, attiva anche per chi vuole dare un'occhiata al corso senza iscriversi;
2. *interazione immediata*: dalle pagine con "vota la risposta migliore", che permettono di avere un feedback in termini di appartenenza ad una percentuale, alle esercitazioni con valutazione automatica immediata
3. *interazione complessa*: invio di elaborati, aggiunta diretta di materiali ai nodi (forum), chat.

In base al livello scelto si potrebbero ottenere crediti diversi, corrispondenti ai diversi livelli di apprendimento.

COOPERATIVA

La visibilità di uno studente è fondamentale. Se un utente visitando un nodo vede una lista di altri utenti che lo stanno visitando, o che lo hanno visitato, deve poter interpellare uno o più utenti (attraverso chat o mail) per chiarimenti e/o confronti sull'argomento trattato dal nodo.

Collegato al corso vero e proprio, un forum dovrebbe permettere ad ogni utente di aggiungere un commento o di commentare il commento altrui.

1.3 Il laboratorio e la didattica on line

Alla didattica on line è opportuno affiancare un laboratorio anch'esso on line. La sperimentazione effettuata in un laboratorio che utilizza le tecnologie on line è molto flessibile, modulare e poco onerosa dal punto di vista economico.

Con una minima dotazione strumentale, il laboratorio on line consente di integrare efficacemente la didattica tradizionale. Inoltre, è relativamente semplice ed economico allargare ulteriormente il ventaglio delle applicazioni.

Le possibilità di realizzare lavori multimediali e acquisire su internet, con software specifico, dati sperimentali su temi multidisciplinari, completano il quadro delle caratteristiche che rendono il laboratorio on line una risorsa notevole per la didattica che propone un modello innovativo di cultura scientifica e tecnologica.

Riassumendo, i vantaggi che offre un tale sistema on line sono i seguenti:

- **Accessibilità**

Questa caratteristica consente una formazione orientata agli impegni dello studente. Tale possibilità è dovuta alla disponibilità di accedere alle informazioni del corso o al laboratorio in linea di principio ad ogni ora e in qualunque posto esista un computer, un modem ed un accesso ad internet.

- **Riduzione dell'investimento riservato al training**

Per sua natura un sistema online non prevede spostamenti fisici da parte dei docenti e degli studenti con conseguente riduzione delle spese di viaggio e dei tempi di inattività.

- **Bassi costi di distribuzione e semplicità di aggiornamento**

Corsi e laboratori sono accessibili da qualunque computer in qualunque zona del mondo. Qualora si rivelassero necessari cambiamenti ai programmi formativi originari, è sufficiente agire sul server su cui è installata la piattaforma per renderli disponibili a tutti gli utenti istantaneamente.

- **Motivazione**

La tecnologia multimediale consente una buona interazione online tra docente e studente. Tale interazione necessita di un materiale didattico fruibile in rete che sia gradevole, coinvolgente ed interattivo, progettato ed implementato studiato appositamente per questo mezzo.

- **Interazione**

Massima interazione e possibilità di intervento in qualsiasi momento grazie a chat in tempo reale e forum di discussione

- **Sicurezza**

Possibilità di interazione e sperimentazione illimitata con strumenti di ogni genere senza danno per gli stessi.

1.4 Esempi di laboratori on line

Esempi di laboratori on line per la gestione di apparecchiature fisiche e virtuali

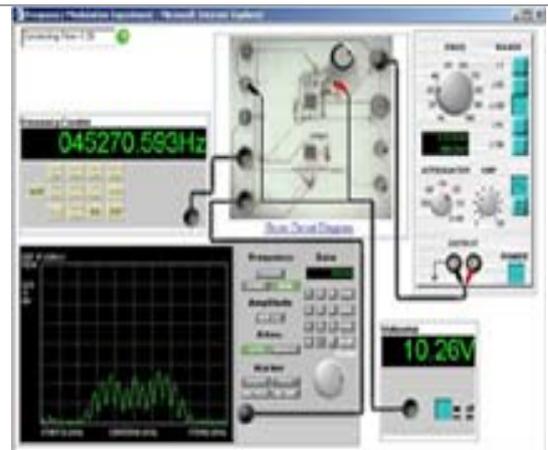
Esistono laboratori in rete in cui uno strumento fisico o virtuale viene messo on line, in modo da poter eseguire da remoto sia il controllo generico dello strumento che una o più esercitazioni guidate. Tra gli esempi più chiari e curati di laboratori on line che remotizzano strumenti fisici troviamo quello del NUS Internet Remote Experimentation. [5]

Per l'interazione con lo strumento viene utilizzata la nuova piattaforma Java 3D, che offre un output grafico di grande impatto e la possibilità di poter navigare virtualmente all'interno di un laboratorio.

Attualmente NUS offre i seguenti esperimenti:

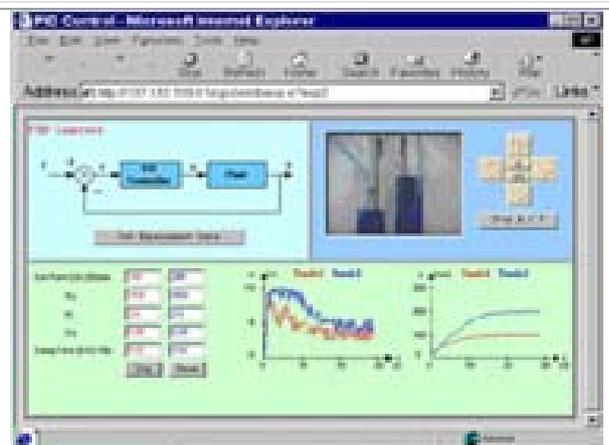
1. Frequency Modulation Experiment

Consente agli studenti di avere un'esperienza diretta di un analizzatore spettrografico, fornisce quindi uno strumento molto utile per la didattica on line. È consentito l'accesso a più persone contemporaneamente, solo l'utente principale potrà condurre l'esperimento, ma gli altri saranno liberi di guardare le azioni ed i risultati.



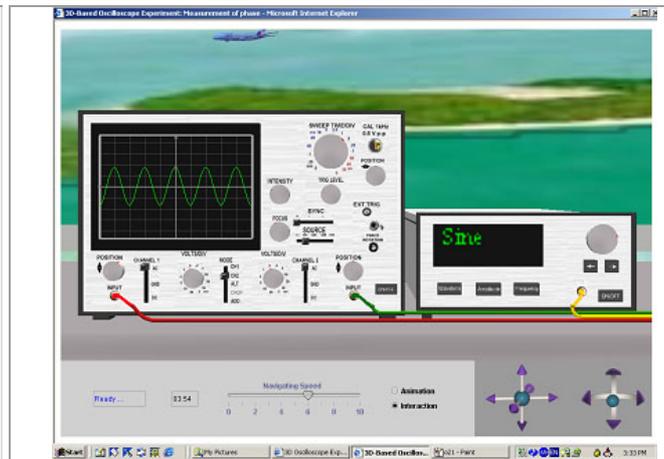
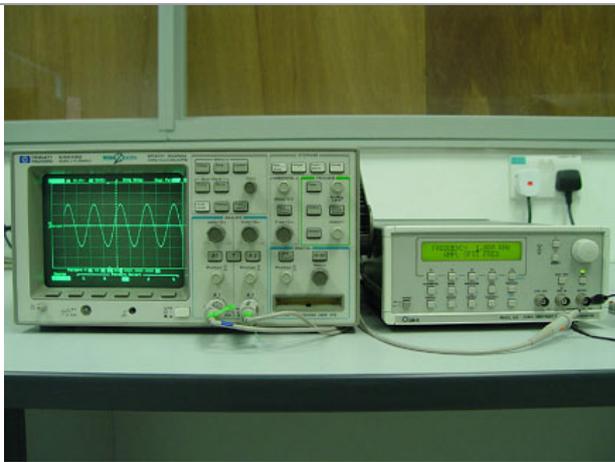
2. Coupled Tank Experiment

È un esperimento che consiste nel regolare il flusso di uscita dell'acqua da due vaschette tramite parametri. È possibile sia visualizzare il grafico delle funzione, che constatare, tramite webcam, che effettivamente i parametri inseriti vengono comunicati allo strumento.



3. 3D Oscilloscope Experiment

L'uso di Java 3D per questo esperimento dà un'esperienza di visualizzazione che è molto più realistica, attraente e naturale dei sistemi 2D esistenti. Tre moduli di controllo sulla navigazione, sulla rilevazione delle collisioni, e sul selezionamento consentono all'utente remoto di muoversi intorno e di eseguire un esperimento, come se si trovasse in un laboratorio reale. I comandi registrati sono convertiti in ordini per gli strumenti reali nel laboratorio fisico e il risultato dell'esperimento è trasmesso di nuovo al client che lo visualizzerà in un apposito pannello.



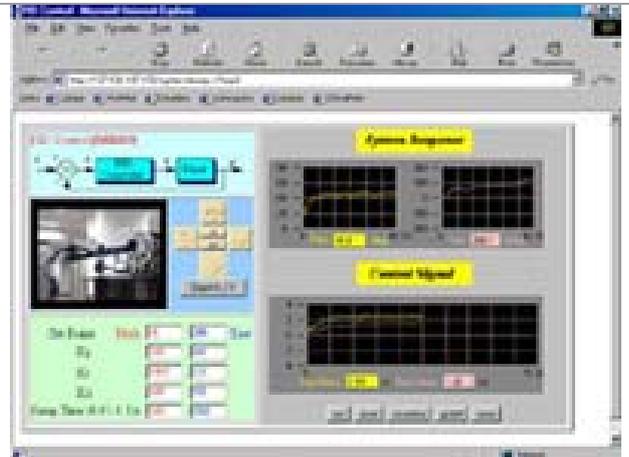
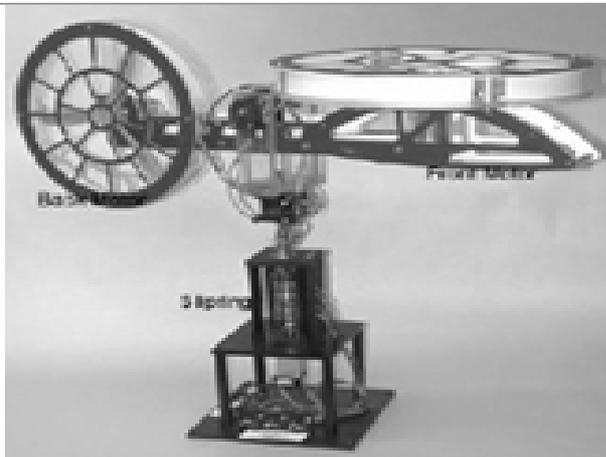
4. 2D Oscilloscope Experiment

Questo esperimento non visualizza a video nel client una simulazione del display dell'oscilloscopio, ma usa una webcam che, una volta spedito il comando e trasferito nello strumento, ne mostra in le immagini in tempo reale.



5. Helicopter Experiment

Consente di interagire col macchinario illustrato sulla sinistra e di controllare tramite webcam gli effetti dei comandi inviati.



6. Robotic Soccer Experiment

È una simulazione sperimentale che si occupa di studi sul moto dei robot; è presente anche in questo caso una webcam per vedere in tempo reale le reazioni dei robot ai comandi inviati.



È tuttavia più comune trovare in rete laboratori virtuali che si occupano di simulare strumenti realmente esistenti allo scopo di mostrarne le funzionalità. La maggior parte di questi esempi sono sviluppati in java (di solito attraverso servlet) per poter essere facilmente utilizzati attraverso la rete.

L'esempio riportato in Fig. 1.1, mostra, tramite un'applet java, graficamente, uno dei concetti principali del networking [6]: gli effetti che produce la distanza sulla trasmissione dei pacchetti. È possibile scegliere la dimensione del pacchetto, la larghezza di banda e la distanza fra l'origine e la destinazione, per potersi rendere conto di come questi parametri incidano sul ritardo di propagazione.

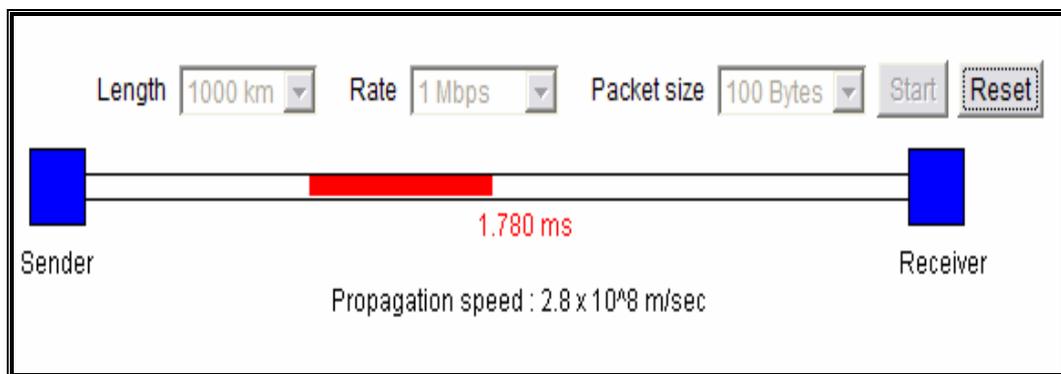


Fig. 1.1 – Ritardo di trasmissione vs. Ritardo di propagazione

L'esempio di Fig. 1.2 mostra invece il comportamento dei pacchetti da trasmettere. In particolare si nota che quando l'emissione è più veloce della trasmissione, la coda si riempie ed i pacchetti che saturano il buffer di trasmissione vengono scartati.

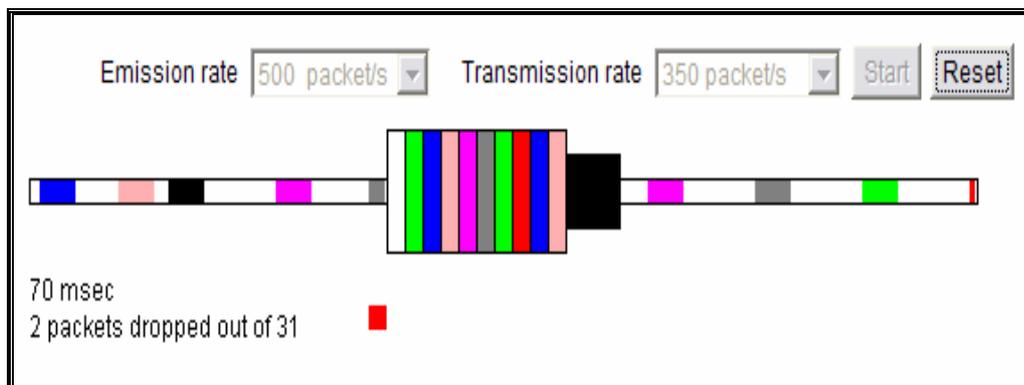


Fig. 1.2 – Code di pacchetti

Altri esempi (Fig. 1.3; Fig 1.3 a; Fig 1.3 b; Fig 1.3 c) forniscono applet che consentono di studiare in maniera visuale fenomeni fisici [7].

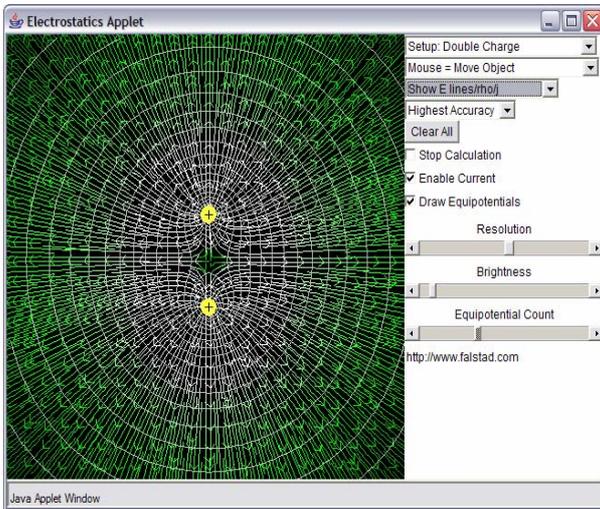


Fig. 1.3 – simulazione elettrostatica.

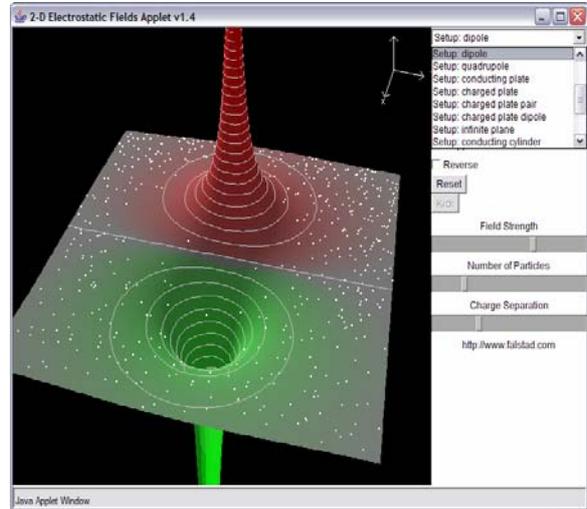


Fig. 1.3 a – simulazione elettrostatica 3D.

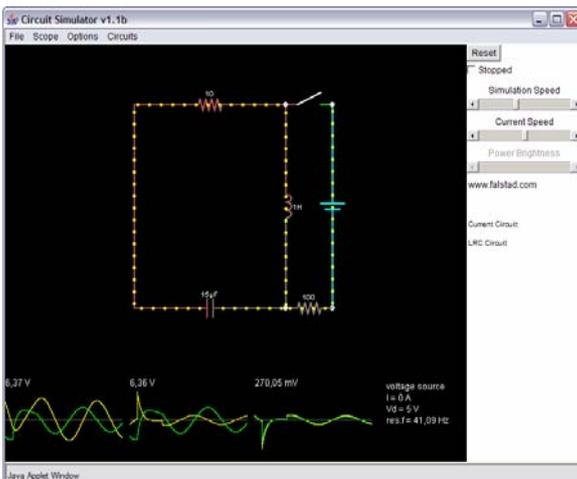


Fig. 1.3 b – simulazione di un circuito.

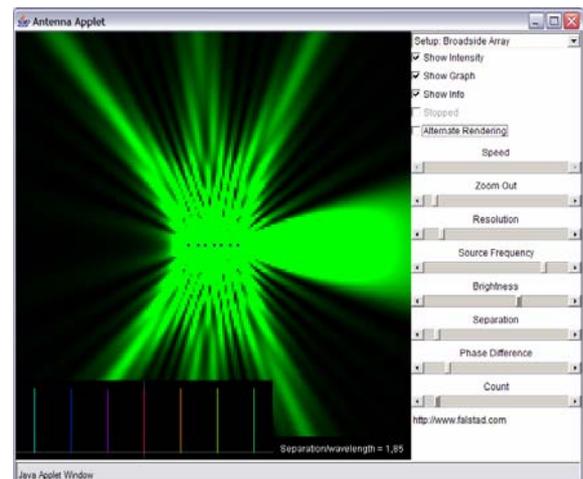


Fig. 1.3 c – Onde generate da un'antenna.

Esempi di esercitazioni di laboratorio

È molto comune trovare in rete esercitazioni di laboratorio generalmente sotto forma di questionari. Un primo esempio è possibile trovarlo nel sito dell'università di Bologna [8]. ExerciseLab genera automaticamente esercitazioni di "Basi di dati" e consente allo studente di testare la sua preparazione sull'argomento. L'iscrizione è

molto semplice in quanto è sufficiente indicare il proprio nome, cognome e matricola, e poi scegliere uno dei due compiti preparati disponibili (Fig 1.4).

Esercizio 9 (punti 2)

Quali delle seguenti dipendenze funzionali sono corrette?

- codMostra, dataMostra -> organizzatore
- sala -> organizzatore
- codMostra, codQuadro -> sala
- organizzatore, dataMostra -> codMostra
- codMostra, codQuadro -> autore
- codMostra, sala -> codQuadro

Fig. 1.4 – ExerciseLab. Esempio domanda

L'esercitazione creata consiste in una serie di domande, la maggior parte a risposta multipla, con la possibilità alla fine di visualizzare il punteggio ottenuto e la correzione delle eventuali risposte sbagliate (Fig. 1.5).

<p>Correzione del compito completata.</p> <p>Risultato:</p> <ul style="list-style-type: none">• Domanda 1 : punteggio 0/2• Domanda 2 : punteggio 0/1• Domanda 3 : punteggio 0/3• Domanda 4 : punteggio 0/2• Domanda 5 : punteggio 0/1• Domanda 6 : punteggio 0/1• Domanda 7 : punteggio 0/2• Domanda 8 : punteggio 0/1• Domanda 9 : punteggio 0/2 <p>Punteggio complessivo: 0/15</p> <p><input type="button" value="Visualizza"/></p>	<p>Esercizio 9 (voto 0/2)</p> <p>Quali delle seguenti dipendenze funzionali sono corrette?</p> <ul style="list-style-type: none"><input type="checkbox"/> organizzatore, dataMostra -> codMostra(errata) <input type="checkbox"/> codMostra, codQuadro -> autore<input type="checkbox"/> codMostra, dataMostra -> organizzatore<input type="checkbox"/> codMostra, sala -> codQuadro(errata) <input type="checkbox"/> codMostra, codQuadro -> sala<input type="checkbox"/> sala -> organizzatore
---	--

Fig. 1.5 – ExerciseLab. Esempio Correzione

Esiste anche un laboratorio di matematica on line, LabSubmittor [9]. È un laboratorio on line sempre basato su questionari di autovalutazione, costruito interamente con applet, anch'esso accessibile soltanto agli studenti registrati. Le domande risultano più "user friendly", rispetto al precedente laboratorio, e in più c'è la possibilità di testare l'effetto della risposta data, prima di confermare. Comunque, in entrambe le esercitazioni, lo scopo è sempre lo stesso: fornire allo studente un modo per verificare il proprio stato di conoscenza sugli argomenti del corso (Fig. 1.6).

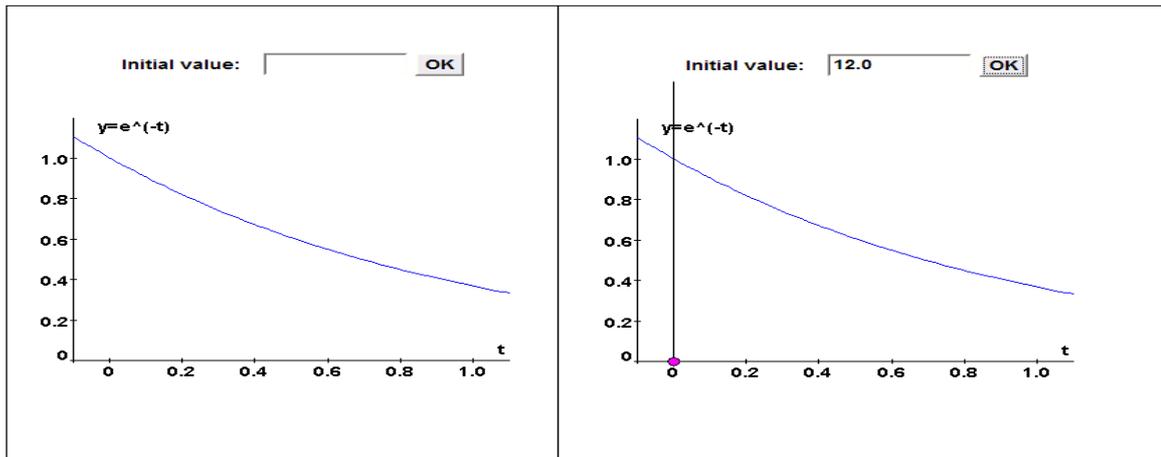


Fig. 1.6 – LabSubmittor. Esempio di Domanda

Un ulteriore esempio è offerto da "Scuola Virtuale" [10], che mette a disposizione tutor on line che, oltre a impartire le lezioni tramite un sistema di videochat, inviano in tempo reale agli studenti domande a risposta multipla. Scuola Virtuale fornisce una serie di corsi tematici che toccano diversi argomenti, costantemente aggiornati e rinnovati (Fig. 1.7).

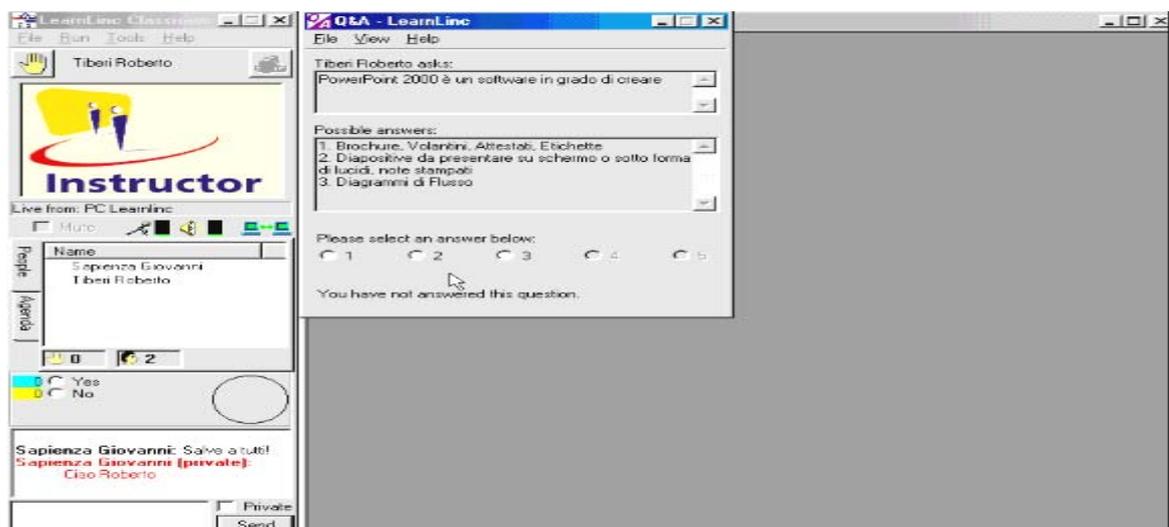


Fig. 1.7 – Scuolavirtuale. Esempio videochat

2. Strumenti per la creazione di laboratori ed esercitazioni On Line

2.1 Esempi di strumenti per la creazione di laboratori on-line

Per quanto riguarda la creazione di laboratori on-line, è stato analizzato un software prodotto dalla National Instrument, LABVIEW (Fig. 2.1) [11].

La sigla sta per "**L**aboratory **V**irtual **I**nstrument **E**ngineering **W**orkbench" (letteralmente "Piattaforma per lo Sviluppo di Strumenti Virtuali da Laboratorio").

È un linguaggio di programmazione grafico, concepito principalmente per la gestione dei processi di acquisizione dati e di controllo della strumentazione nell'automazione di laboratorio.

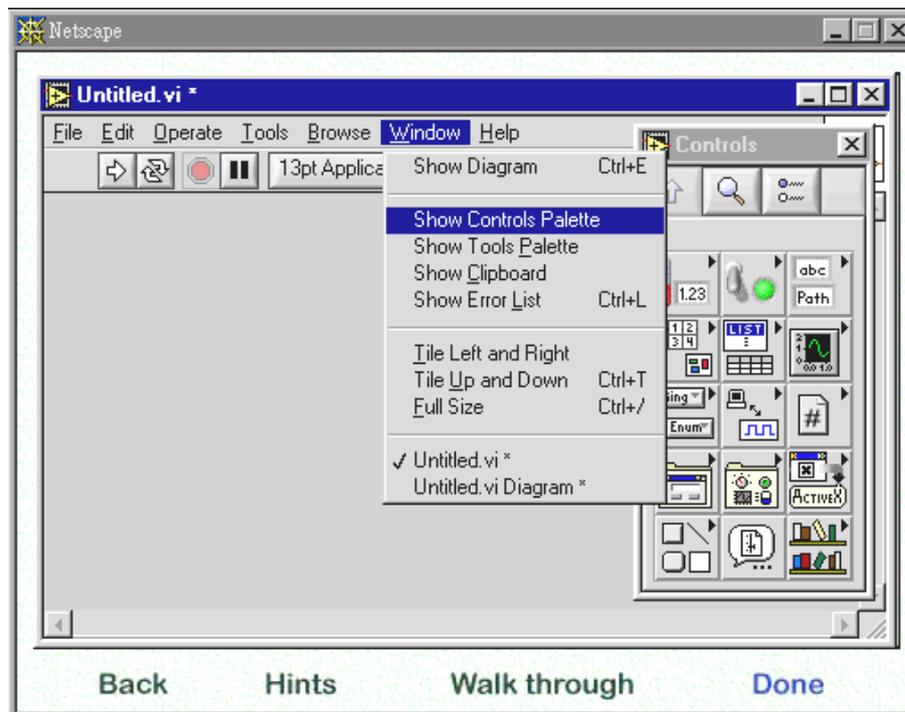


Fig. 2.1 – LABVIEW. Interfaccia principale.

I programmi realizzati con LabVIEW vengono detti semplicemente VI (Virtual Instrument) perché imitano gli strumenti reali: sono appunto Strumenti Virtuali. Gran parte della potenza e della convenienza di LabVIEW sono dovute alla sua modularità. È possibile creare un VI completo suddividendolo in moduli (sub-VI) che verranno realizzati uno per volta.

Aprendo un nuovo VI compaiono immediatamente due finestre:

Control Panel (Pannello di Controllo)

È una pagina grigia con quadretti. Questo foglio è utilizzato per creare l'interfaccia grafica dove l'utente può inserire valori, cambiare lo stato delle variabili, leggere il risultato delle operazioni svolte dal programma, analizzare grafici, eccetera.

Block Diagram (Diagramma a blocchi)

In questo spazio invece viene inserito il codice del programma (icone e connessioni) che utilizzerà gli elementi del front panel per eventualmente elaborarli e produrre dei risultati. Cliccando con il tasto destro del mouse comparirà la functions palette dalla quale è possibile scegliere la funzione o la struttura da utilizzare all'interno del codice [12]. Ad esempio si può realizzare un programma che si occupa della conversione di una temperatura inserita dall'utente da gradi centigradi a gradi fahrenheit [13].

Il programma legge un valore numerico inserito dall'utente rappresentante una temperatura in gradi centigradi e ne restituisce il valore in gradi fahrenheit.

Aprendo un nuovo VI si inseriscono due controlli numerici selezionandoli dal pannello controlli ed etichettandoli opportunamente (Fig. 2.2).

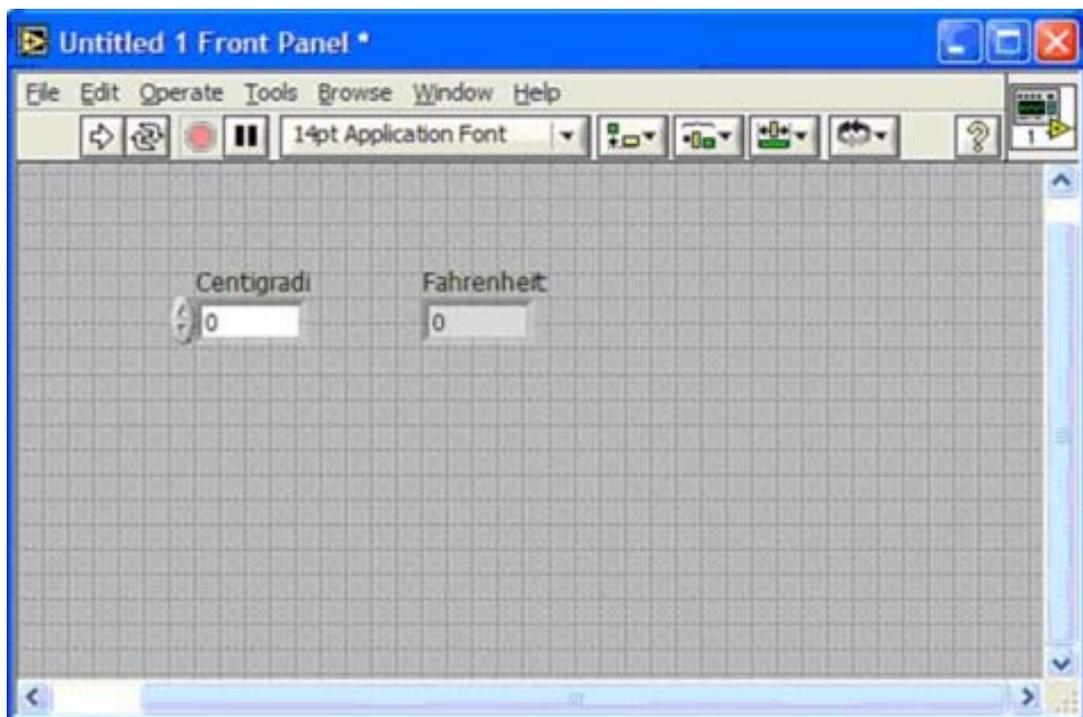


Fig. 2.2– LABVIEW. Esempio di utilizzo.

Spostandosi sul Block Diagram, con pochi click è possibile selezionare, inserire e collegare tra loro le funzioni necessarie per l'operazione (in questo caso ci serviranno una moltiplicazione ed un'addizione e le costanti) (Fig. 2.3).

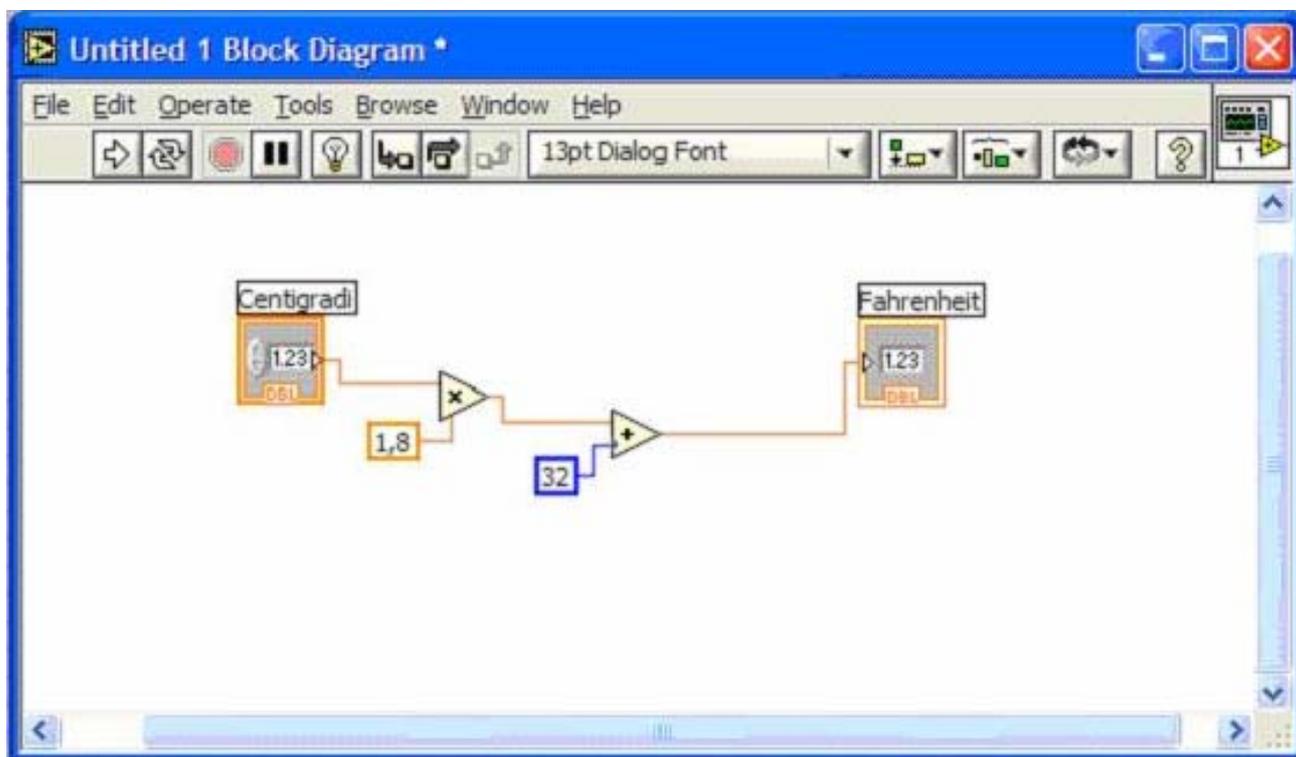


Fig. 2.3– LABVIEW. Esempio di utilizzo.

A questo punto il programma è pronto. In pochi minuti, e senza alcuna nozione di programmazione si è realizzato un convertitore visuale di temperatura da gradi centigradi a gradi fahrenheit.

2.2 Esempi di strumenti per la creazione di esercitazioni on-line

E' possibile trovare in rete numerosi strumenti software (web-based e non) che consentono di creare questionari di vario genere. Vengono di seguito descritti alcuni di questi strumenti che sono stati analizzati sia dal punto di vista delle caratteristiche principali di funzionamento che dell'efficienza e versatilità nel loro effettivo utilizzo.

GO-TEST

GO-TEST è offerto da gonetwork [14]. È un software che permette la creazione di questionari, quiz e test in formato HTML, pronti per essere svolti immediatamente a

video oppure pubblicati su Internet. Consente in modo agevole di progettare, realizzare e pubblicare questionari e test on line, la raccolta dei dati in tempo reale e la relativa analisi. Per ogni questionario generato è possibile stabilire le politiche di accesso, ovvero stabilire se l'accesso è riservato ai soli utenti registrati oppure è accessibile da tutti. La struttura può essere molto varia con tabelle, risposte chiuse, aperte, miste, ecc.

Il software offre le seguenti funzionalità:

- creazione di questionari
- test preliminare di un questionario esistente, prima della sua effettiva pubblicazione
- definizione delle politiche di accesso
- attivazione del questionario
- verifica dei risultati di un'indagine
- report dell'indagine
- esportazione dati

La gestione questionari di GO-TEST è effettuata tramite le sezioni mostrate in Fig. 2.4.

Interfaccia di Gestione		
Amministratore: admin		
Gestione Questionari	Funzioni di Amministrazione	Varie
<ul style="list-style-type: none"> > Crea un nuovo Questionario > Modifica un Questionario esistente > Collauda un Questionario > Copia un Questionario esistente > Cambia lo stato di un Questionario (attiva/fine/elimina) > Cambia accesso ad un Questionario (numero massimo degli Utenti.) > Vedi i risultati di un Questionario > Correlare i risultati di un Questionario > Vedi report di un Questionario > Esporta dati in formato CSV 	<ul style="list-style-type: none"> > Cambia la tua Password > Gestione Amministratori > Gestione Utenti > Gestione Gruppi 	<ul style="list-style-type: none"> > Guida dell' amministratore > Disconnetti

Fig. 2.4 GO-TEST. Pagina principale

Tramite la sezione “Crea un nuovo Questionario” è possibile creare un nuovo sondaggio.

Sono disponibili le seguenti sotto-sezioni:

- Generale

In questa sezione (Fig. 2.5) vengono definite le informazioni generali del questionario. Sono previste alcune informazioni obbligatorie, senza le quali non è possibile proseguire con la compilazione. In particolare deve essere definito il “proprietario” del questionario, ovvero il gruppo che amministra e gestisce il questionario stesso.

Creazione Questionario

Generale Domande Ordine Anteprima Fine

Le informazioni in questi pannelli si applicano all'intero sondaggio. Riempi questa pagina poi vai nel **Campotab** per correggere campi individuali.

Nome *Richiesta*
Nome sondaggio. Questo è usato per tutti i futuri accessi a questo sondaggio. (nessuno spazio, solo caratteri alfanumerici)

Proprietario admin / *Richiesta*
Gruppo che possiede questo sondaggio.

Titolo *Richiesta*
Titolo del sondaggio. Questo appare in testa ad ogni pagina del sondaggio. (libero; include spazi)

Sottotitolo
Sottotitolo del sondaggio. Appare sotto il titolo. (libero; include spazi)

Informazioni aggiuntive
Testo che apparirà nel sondaggio prima di ogni campo. (es. istruzioni, informazioni generiche...)

Pagina di conferma (URL)
L'URL alla quale un utente è reindirizzato dopo aver completato un sondaggio (es. pagina di ringraziamento...).

(testata)

(corpo)
Intestazione nella "pagina di conferma" (appare dopo che un utente completa un'indagine). Questo appare in grassetto. (l'URL, se presente, ha la precedenza sul testo di conferma.)

Email
Spedisci una copia di ogni risultato all'indirizzo (o lascia vuoto per non inviare nessuna email di backup).

Tema (css) da utilizzare con questo sondaggio.

crawlcrum
 default
 mystic
 template
 arancione

Clicca **cancello** per cancellare questo Questionario, o clicca **continua** per procedere alla prossima sezione.

Fig. 2.5 GO-TEST. Sezione “Generale”

- Domande

In questa sezione possono essere create le domande del questionario. È possibile selezionare una delle seguenti tipologie di risposta (Fig. 2.6):

- Yes/No: È possibile rispondere solamente “sì” oppure “no”.
- Text box: Casella di testo a linea singola. La domanda viene inserita, e come possibilità di risposta si presenta una casella testo.
- Essay Box: Testo libero. Il testo che viene digitato compare come testo della domanda. Compare una casella di testo a scorrimento in cui si può rispondere liberamente.
- Radio Button: È possibile selezionare solo una fra le alternative proposte.
- Chek Boxes: È possibile selezionare solo una fra le alternative proposte.
- Dropdown box: le opzioni inserite nelle linee di risposta vengono presentate all’utente come scelte possibili di un menu a tendina.
- Rate scale: Consente di esprimere un indice di gradimento.

Selezionando l’opzione “Richiesta = SI”, si può rendere obbligatoria la risposta ad una o più domande.

Creazione Questionario

Generale **Domande** Ordine Anteprima Fine

Correggi questo campo, o clicca il numero del campo che vorresti correggere:

Nuovo campo				
Nome del campo	Tipo	Lunghezza	Precisione	Richiesta?
<input type="text"/>	Text Box	<input type="text" value="0"/>	<input type="text" value="0"/>	No
Text	<div style="border: 1px solid black; padding: 2px;"> Yes/No Text Box Essay Box Radio Buttons Check Boxes Dropdown Box Rate (scale 1.5) Date Numeric Section Text </div>			
Introdurre le possibili risposte (se applicabile domanda. Tutte le linee vuote verranno eliminate).				
1.	<input type="text"/>			
2.	<input type="text"/>			
3.	<input type="text"/>			
4.	<input type="text"/>			
5.	<input type="text"/>			
6.	<input type="text"/>			
7.	<input type="text"/>			
8.	<input type="text"/>			
9.	<input type="text"/>			
10.	<input type="text"/>			

Clicca cancella per cancellare questo Questionario, o clicca continua per procedere alla prossima sezione.

Fig. 2.6 GO-TEST. Sezione "Domande"

- Ordina

In questa sezione è possibile stabilire l'ordine in cui saranno presentate le domande nel questionario finale.

- Anteprima

In questa sezione è possibile visualizzare in anteprima il questionario generato, in modo da poter effettuare eventuali modifiche e/o variazioni prima della sua effettiva pubblicazione.

- Fine

Cliccando su “Fine” viene terminata la fase di editing del questionario e viene comunicato l’indirizzo di attivazione su web del questionario stesso. È possibile comunque controllare nuovamente in ogni momento il questionario, completarlo, modificarlo prima della sua attivazione tramite la sezione “controlla un questionario esistente”.

GoTest è fornito come servizio ASP, ovvero è installato su un Server web gestito interamente da GoNetwork.

In alternativa il Cliente può acquistare la licenza d'uso ed installare GoTest su un proprio Server web con sistema operativo LINUX.

Tecchio.net

Un altro esempio di generatore di test è un’applicazione creata da Roberto Tecchio [15]. L’applicazione è sviluppata con Microsoft.NET, quindi può essere eseguita unicamente su server Microsoft. È divisa in due sezioni: Utente ed Amministratore.

All'utente che accede viene presentata una lista dei questionari (Fig. 2.7), unitamente all'indicazione del loro stato:

- *da fare*, se non sono mai stati esplorati;
- *iniziato*, se l'utente ha iniziato almeno a leggere le domande;
- *sospeso* se ha risposto solo in parte;
- *terminato* se ha risposto almeno a tutte le domande obbligatorie (questionario di reazione) o se ha concluso la prova di verifica (questionario di valutazione).

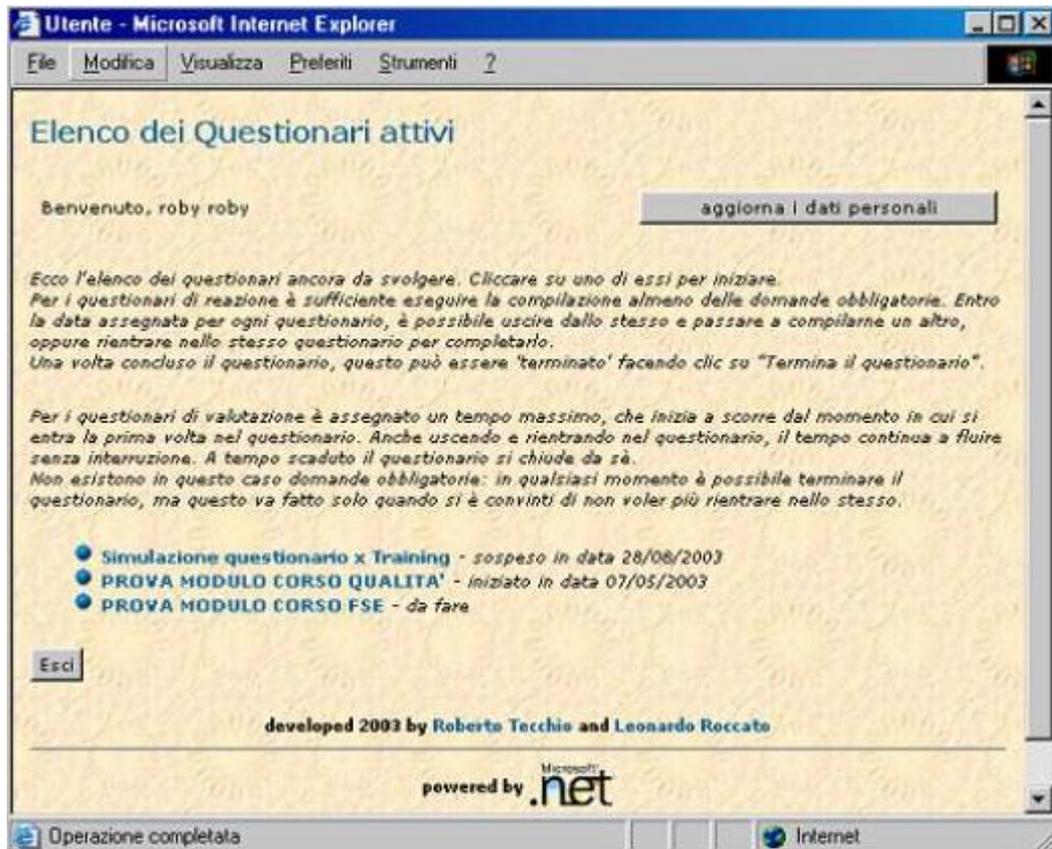


Fig. 2.7 – tecchio.net. HomePage Utente.

Se l'utente esegue un **questionario di reazione**, può interromperlo e riprenderlo quante volte vuole, anche in tempi diversi, entro una data massima di consegna assegnata dal tutor o docente. Il questionario si considera terminato quando è scaduta la data di consegna o quando l'utente ha risposto almeno alle domande segnate come obbligatorie.

Se l'utente esegue un **questionario di valutazione**, all'inizio dell'esecuzione dello stesso si avvia un timer che scandisce il tempo a disposizione, assegnato anch'esso dal docente o tutor. Al termine del tempo il questionario si chiude da sé e non è possibile riprenderlo, neanche da un'altra postazione.

La pagina di esecuzione del questionario si presenta con :

- Selezione delle domande da eseguire;
- Testo della domanda, ed eventuale commento o suggerimento aggiuntivo;

- Help opzionale: facendo clic sul punto interrogativo a destra del titolo si accede ad una finestra di aiuto (disponibile se l'autore della domanda ha ritenuto utile fornire tale aiuto);
- Scelte possibili, in una delle forme di seguito specificate;
- Pulsante 'termina il questionario, e, nel caso di *questionari di reazione*, anche pulsante 'torna all'elenco questionari', per sospendere momentaneamente l'esecuzione;
- Tempo rimanente, nel caso di *questionari di valutazione* (Fig. 2.8).

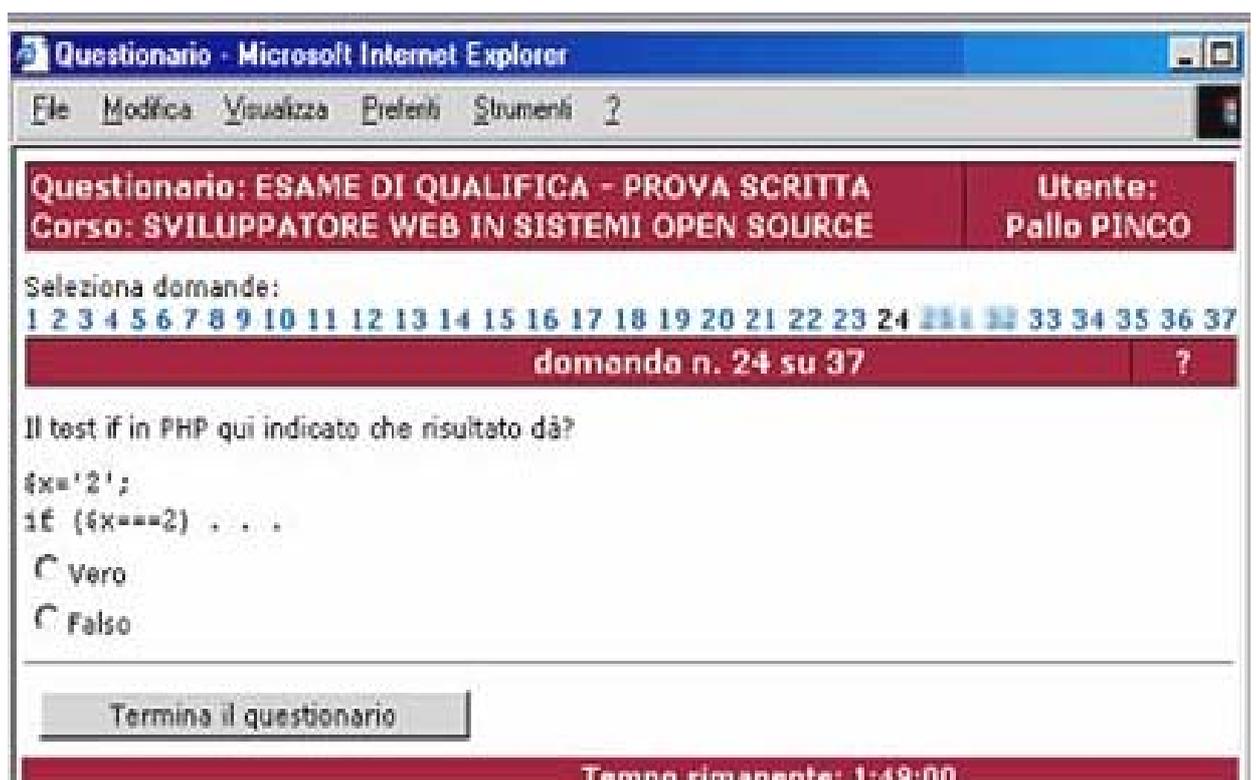


Fig. 2.8 – tecchio.net. Esempio Questionario.

SCHEMI DI DOMANDE POSSIBILI

<p>A risposta libera</p>	<p>Dire quale sarà l'output dell'esecuzione della seguente classe:</p> <pre>class A(public static void main(String args[]){ int vettore[]={3,24,1,-5,84,2}; int n = vettore[0]; for(int i=0; i<vettore.length; i++){ if(vettore[i]>n) n=vettore[i]; </pre>
<p>A risposta singola</p>	<p>Il test if in PHP qui indicato che risultato dà?</p> <pre>\$x='2'; if (\$x==2) . . .</pre> <p><input type="radio"/> Vero</p> <p><input type="radio"/> Falso</p>
<p>A risposta multipla</p>	<p>Il docente ha utilizzato più di un metodo didattico?</p> <p><input type="checkbox"/> lezione tradizionale <input type="checkbox"/> commento di lucidi, diapositive</p> <p><input type="checkbox"/> simulazioni <input type="checkbox"/> lavori di gruppo</p> <p><input type="checkbox"/> esposizioni e mostre</p>
<p>A elenchi di valori</p>	<p>Secondo te si sarebbe lavorato meglio dando più o meno spazio alle seguenti metodologie:</p> <p>lezioni teoriche <input type="text" value="-"/></p> <p>esercitazioni e simulazioni <input type="text" value="+"/></p> <p>analisi delle diverse esperienze di lavoro <input type="text" value="-"/></p>
<p>A elenchi da ordinare</p>	<p>Indicare la corretta sequenza delle clausole di un'istruzione SELECT in MySQL:</p> <div style="border: 1px solid gray; padding: 5px;"> <p>WHERE filtri_di_ricerca</p> <p>FROM tabella</p> <p>GROUP BY criteri_di_raggruppamento</p> <p>SELECT elenco_dei_campi</p> <p>[INNER - LEFT - RIGHT - NATURAL] JOIN elenco_relazioni</p> <p>LIMIT rigainiziale, numerorighe</p> <p>ORDER BY criteri_di_ordinamento</p> </div> <p style="text-align: right;"><input type="button" value="sposta in su"/></p> <p style="text-align: right;"><input type="button" value="sposta in giù"/></p>

Tramite la sezione amministratore è possibile amministrare le tabelle degli utenti (Fig. 2.9), dei corsi, delle aree tematiche, delle domande, delle risposte ed opzioni per le domande stesse, creare questionari tipo, creare edizioni specifiche dei questionari, controllarne lo stato ed ottenere la reportistica delle risposte.

The screenshot shows a web interface titled 'Corsi' with a search filter 'filtra per'. Below is a table with the following data:

		Titolo	Anno	Stage	
		Addetti/a alla contabilità	2002/2003	161	
		Addetti/e alle vendite e al visual merchandising	2002/2003	161	
		Corso FSE 1/01 - 1^B -	2002	0	
		Corso FSE 1/01 - 1^C -	2002	0	
		Corso simulazione 1	2002	0	
		Gestione Aziendale	2002	140	
		Nuovo Obbligo Formativo	2004	240	
		Spec Vendite	2002	340	

Fig. 2.9 – tecchio.net. Esempio Tabella.

Per gli Utenti è possibile inserire tutti i dati anagrafici, quelli relativi allo stage e all'eventuale occupazione, ed è possibile assegnare un utente ad uno o più corsi. Gli utenti possono essere abilitati o meno; solo quelli abilitati possono accedere ai questionari. È possibile definire più aree tematiche in cui raggruppare le domande, per comodità di reperimento; inoltre, ogni area può contenere una o più sottoaree. Al momento dell'inserimento di una domanda, se si tratta di domanda per un questionario di valutazione, si assegna anche un livello di difficoltà (molto facile, facile, media, difficile, molto difficile) che stabilirà un peso per la domanda stessa. È poi possibile definire un suggerimento o commento, che apparirà assieme al testo, ed un eventuale help per la finestra di aiuto. La preparazione dei questionari si distingue in due fasi: costruzione di un modello di questionario e pubblicazione di una edizione di un questionario. I modelli sono questionari tipo, studiati per specifiche situazioni, e messi a disposizione di tutti i tutor e docenti. Le edizioni sono copie di un

modello di questionario tipo, erogate per uno specifico corso in un dato momento. È, quindi, possibile erogare un modello di questionario a più corsi in vari momenti, creandone più edizioni distinte. Queste conterranno tutte le stesse domande, ma saranno erogate a utenti diversi, quindi con reportistiche diverse. Per creare una edizione di un questionario è sufficiente assegnare uno dei modelli di questionario tipo ad un corso e dargli un nome ed una data di scadenza; per i questionari di valutazione è necessario aggiungere anche la votazione massima ed il tempo assegnato per la prova. Anche le edizioni vanno attivate per erogarle. In qualunque momento è possibile esaminare lo stato dei questionari, ovvero vedere quanti ne sono stati completati, o solo iniziati, e in quali corsi, ed è anche possibile esaminare in dettaglio le risposte date da un corsista (ma non modificarle!), anche a questionario non ancora terminato.

Per i questionari di reazione vengono elaborate le statistiche delle risposte date (Fig. 2.10); per le risposte a testo libero, vengono semplicemente elencate le risposte date da ogni corsista.

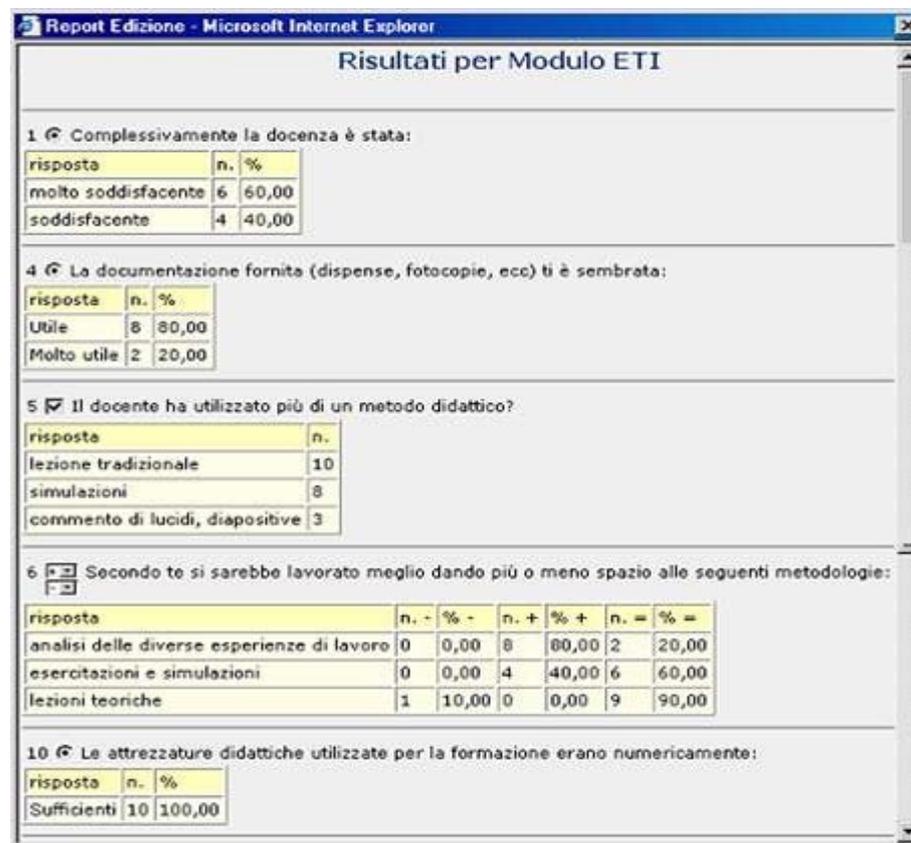


Fig. 2.10 – tecchio.net. Esempio Statistiche.

Per i questionari di valutazione viene esaminata ogni risposta data per ogni domanda; per ognuna di queste si calcola la percentuale di esattezza assegnata, e la si applica al peso assegnato alla domanda. Sommando così tutti i punteggi ottenuti si ottiene un punteggio complessivo per quell'utente, che viene rapportato al voto massimo stabilito dal docente o tutor per quel questionario. Il tutto viene riportato a stampa e/o video. (Fig. 2.11)

Report Edizione - Microsoft Internet Explorer

Valutazione ESAME DI QUALIFICA - PROVA SCRITTA

Punti questionario: 50 - Punti domande: 151

1 Per variare font, dimensioni e colori dei testi nelle pagine web si possono usare: **punti : 5**

richiesta	% ass.	scelta	risp.esatta	punti
i fogli stile CSS	35%	ottimale	ottimale	1,75
i tag , <I>, , 	15%	accettabile	errato	0
il tag <BODY TEXT=... LINK=...>	15%	accettabile	errato	0
il tag 	35%	sconsigliato	sconsigliato	1,75

2 Indicare quali delle seguenti situazioni si verificano quando un form HTML viene inviato con il metodo GET: **punti : 5**

risposta	% ass.	punti
I dati immessi vengono inviati assieme all'URL della pagina	25%	1,25
I dati sono inglobati in un'unica stringa nell'URL della pagina	25%	1,25
I dati sono descritti con coppie nome = valore	25%	1,25
E' possibile inviare direttamente il form ad un indirizzo di posta elettronica	0%	0
I dati inviati rimangono memorizzati tra i siti visitati di recente nel browser	25%	1,25

55 Come è necessario specificare il layout di una classe che estende JPanel per poter disporre gli oggetti all'interno di esso specificandone posizione e dimensione? **punti : 5**

risposta	risp.esatta	% ass.	punti
setLayout(new JButton());	setLayout(null);	0%	0

56 Per la compilazione di una classe Java da riga di comando deve essere utilizzato il comando: **punti : 2**

risposta	risp.esatta	% ass.	punti
javac	javac	100%	2

57 Dire quale sarà l'output dell'esecuzione della seguente classe: **punti : 5**

p.ti diff.	risposta	valore esatto	tolleranza +/-	punti
5	84	84	5	5

58 La parola chiave new è un comando che: **punti : 2**

risposta	risp.esatta	% ass.	punti
crea un nuovo oggetto in memoria	crea un nuovo oggetto in memoria	100%	2

59 Gli indici di un array Java di N posizioni vanno da **punti : 2**

risposta	risp.esatta	% ass.	punti
0 e N-1	0 e N-1	100%	2

Punteggio per **Punti totalizzati 112,75 su 151 -> Valutazione 37,34 su 50**

Fig. 2.11 – techio.net. Esempio valutazione questionario.

WebQuiz 2002

Un ulteriore esempio è offerto da SmartLite Software [16] che commercializza un programma base per la creazione di quiz on line, WebQuiz 2002.

WebQuiz 2002 è dotato di una interfaccia molto semplice, con tutte le funzionalità a portata "di mouse". Il prodotto è stato progettato per ottenere la massima facilità di utilizzo rendendo estremamente veloce la creazione dei quiz. L'utilizzo di WebQuiz 2002 non richiede alcuna conoscenza di HTML o Javascript, dopo avere inserito le domande, i questionari possono essere pubblicati automaticamente sul proprio sito Internet. Il programma, infatti, include un client FTP necessario per effettuare l'upload diretto dei file. Per una semplicità ancora maggiore, le domande possono essere pubblicate sul sito di WebQuiz on line. Sono solo 4 le fasi necessarie a WebQuiz 2002 per creare e pubblicare un quiz su Internet. Prima di tutto è possibile impostare le varie opzioni del quiz, dalla successione delle domande e delle risposte alla definizione di una valutazione. In particolare WebQuiz 2002 supporta le seguenti caratteristiche:

- Visualizzazione di un numero predefinito di domande
- Successione sequenziale o casuale delle domande e delle risposte
- Impostazione di una valutazione finale basata sul punteggio totale realizzato
- Inserimento di messaggi mostrati prima o dopo le domande
- Visualizzazione di tutte le domande in una sola pagina o di una domanda per pagina

Successivamente si inseriscono i testi delle varie domande e risposte, scegliendo tra: scelta multipla, risposta multipla, vero / falso, inserimento.

Ad ogni domanda è possibile associare un'immagine (in formato GIF, JPG, PNG) ed utilizzare formule matematiche. Durante la pubblicazione del quiz, le formule

WinAsk 2000

SmartLite Software offre anche uno strumento gratuito per la creazione di questionari: WinAsk 2000, un'applicazione windows che consente di creare questionari in formato PDF, (Fig. 2.13a) su video (Fig. 2.13b) oppure in formato HTML.

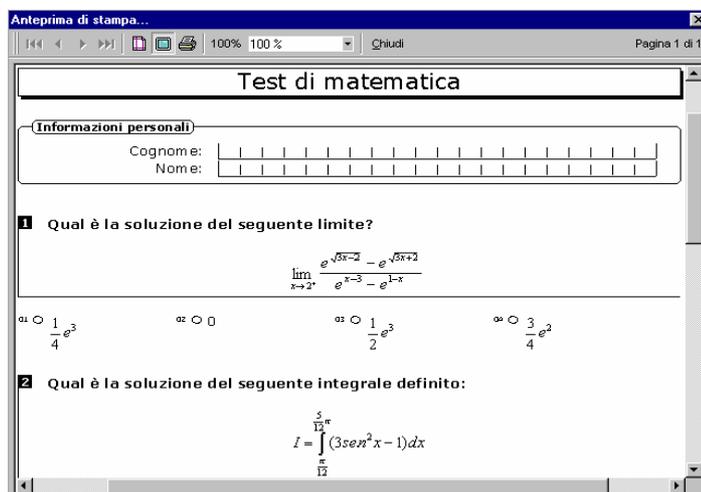


Fig. 2.13a. WinAsk 2000. Questionario PDF

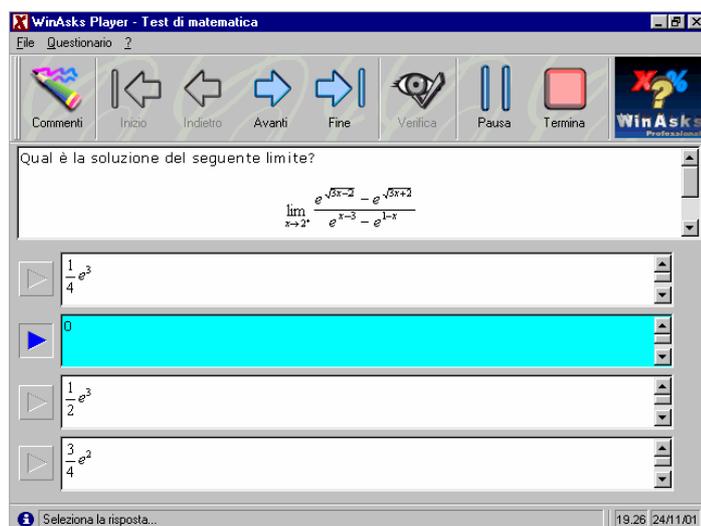


Fig. 2.13b. WinAsk 2000. Questionario su Video.

2.3 Vantaggi e svantaggi degli strumenti analizzati

Dopo aver analizzato le soluzioni precedenti, viene adesso riportata una tabella riassuntiva con le caratteristiche e i vantaggi/svantaggi dei vari strumenti:

PRODOTTO	CARATTERISTICHE	VANTAGGI	SVANTAGGI
LabView	<ul style="list-style-type: none"> • Off line • Interfaccia semplice • Possibilità di acquistare pacchetti di espansione con funzioni e componenti aggiuntivi 	<ul style="list-style-type: none"> • Personalizzazione degli strumenti da creare • Non richiede nessuna competenza di programmazione classica • Risultati veloci e soddisfacenti 	<ul style="list-style-type: none"> • Utile agli ingegneri elettrici • Non vi è la possibilità di creare quiz ed esercitazioni • Costo elevato • Disponibile solo in lingua inglese
GO-TEST	<ul style="list-style-type: none"> • On Line • Gestione degli utenti • Tipologia domande personalizzabile • Proprio server in affitto (su richiesta) 	<ul style="list-style-type: none"> • Ottima personalizzazione delle domande • Ottimo sistema di statistiche per le risposte • Buona gestione degli utenti 	<ul style="list-style-type: none"> • Adatto più per sondaggi che per questionari valutativi • Poco intuitivo • Non multimediale • Costo elevato • Manca la possibilità di creare strumenti o interagire con strumenti fisici
tecchio.net	<ul style="list-style-type: none"> • On Line • Orientamento professionale • Tipologia domande personalizzabile • Gestione gerarchica degli utenti • Manutenzione in tempo reale da controllo remoto 	<ul style="list-style-type: none"> • Buona personalizzazione delle domande • Ottimo sistema di statistiche e correzione per le risposte • Doppia tipologia di questionario selezionabile 	<ul style="list-style-type: none"> • Non sono chiari i termini di utilizzo • Non è multimediale • Vincolo ad utilizzare un server Microsoft • Manca la possibilità di creare o interagire con strumenti
WebQuiz	<ul style="list-style-type: none"> • Off line • Interfaccia Web • Layout questionario personalizzabile 	<ul style="list-style-type: none"> • Semplice ed intuitivo • Wizard di pubblicazione on line del questionario • Sito Web gratuito per la pubblicazione dei questionari 	<ul style="list-style-type: none"> • Poco personalizzabile • Non multimediale • Manca la possibilità di creare o interagire con strumenti
WinAsk 2000	<ul style="list-style-type: none"> • Off line • Interfaccia applicazione windows • Formato questionari personalizzabile 	<ul style="list-style-type: none"> • Gratuito • Molto semplice ed intuitivo 	<ul style="list-style-type: none"> • Poco personalizzabile • Non multimediale • Manca la possibilità di creare o interagire con strumenti

La tabella precedente riporta pregi e difetti degli strumenti analizzati nei paragrafi precedenti. Per quanto riguarda gli strumenti per la creazione di laboratori on-line è possibile trovare nella rete soluzioni che consentono di esplorare ed interagire con strumenti fisici o virtuali per mezzo di interfacce principalmente create in java, o anche software che consentono di creare strumenti virtuali. Non è stato trovato in rete alcuno strumento che consenta invece di creare in modo veloce e semplice interfacce di strumenti fisici. In ogni caso i software che consentono di creare interfacce di strumenti virtuali non offrono la possibilità al docente di creare delle esercitazioni basate sullo strumento in esame, e, viceversa, i software che consentono di creare questionari ed esercitazioni, non offrono la possibilità di interagire con strumenti fisici o virtuali. Alla luce di questo scenario nasce On-Lab, creato con lo scopo di poter avvicinare lo studente a strumenti che non sono solitamente accessibili (perché costosi e delicati o perché situati in luoghi non accessibili normalmente dagli studenti). Per consentire tale interazione viene offerta al docente la possibilità di creare delle esercitazioni basate su uno strumento fisico o virtuale che On-Lab consente di implementare. On-Lab nasce quindi con lo scopo di “integrare” entrambe le tipologie di soluzioni precedentemente analizzate. On-Lab viene illustrato in dettaglio nel capitolo seguente.

3. On-Lab: un ambiente web per lo sviluppo di laboratori on line

3.1 Obiettivi e caratteristiche di base

Come è stato detto in precedenza, i laboratori on line sono strumenti di supporto alla didattica molto utili per l'apprendimento, ma, soprattutto in Italia, sono poco diffusi. Il motivo principale potrebbe essere dovuto al fatto che sono necessarie alcune competenze tecniche per creare ad esempio un questionario interattivo ed inserirlo in rete. Il progetto On-Lab nasce anche per risolvere questo problema, oltre alla possibilità di creare interfacce di strumenti fisici o virtuali e a consentire in maniera semplice la creazione di un'esercitazione basata sullo strumento, consente in maniera trasparente di pubblicare on line le esercitazioni create, per metterle a disposizione degli studenti. On-Lab non è una simulazione, i comandi lanciati tramite l'esercitazione creata su un'interfaccia fisica, vengono eseguiti realmente sullo strumento. Chi esegue l'esercitazione quindi, si rende conto effettivamente di ciò che avviene lanciando determinati comandi su uno strumento, proprio come se effettivamente stesse lavorando sullo strumento stesso. L'interazione fra gli strumenti e l'esercitazione viene filtrata da un'interfaccia di strumento, creata dall'amministratore.

Il metodo di creazione delle interfacce è semplice, basta innanzitutto stabilire se l'interfaccia sarà fisica, cioè realmente esistente, oppure virtuale. Nel primo caso (interfaccia fisica) è sufficiente specificare la lista dei comandi accettati dallo strumento. Nel secondo caso (interfaccia virtuale) oltre ai comandi è necessario specificare l'output (in forma testuale e/o grafica) che il comando immesso dovrà produrre.

Il Docente che vuole creare un'esercitazione per i propri studenti, deve solamente selezionare l'interfaccia appropriata e scrivere le domande.

L'applicazione, per ovvi motivi di sicurezza, è divisa in tre livelli di autorizzazione: Amministratore, Docente e Studente.

Il compito di creare una nuova interfaccia spetta solamente all'amministratore, il docente invece può liberamente creare tutte le esercitazioni che vuole, mentre lo studente può solo eseguirle.

Gli obiettivi che si è pensato di conseguire con il lavoro qui di seguito descritto si possono riassumere nei seguenti punti:

- un ambiente facile da usare, sia da parte del docente sia da parte dello studente;
- possibilità per il docente di creare esercitazioni on line;
- interazione automatica con gli strumenti (il docente deve solo creare un'esercitazione basandosi su un'interfaccia di strumento creata dall'amministratore);
- possibilità per lo studente di sperimentare e testare comandi sugli strumenti e sulle apparecchiature studiati;
- possibilità per l'amministratore di creare in maniera veloce interfacce per qualsiasi strumento;

3.2 Ambiente di Sviluppo

Java e le Servlet

Per sviluppare On-Lab si è scelto il linguaggio Java e le tecnologie collegate quali le servlet. La motivazione della scelta del linguaggio Java è legata, fondamentalemente, alle caratteristiche di flessibilità e portabilità che contraddistinguono tale linguaggio di programmazione; ciò consente anche una maggiore diffusione dell'applicazione che su di esso si basa.

In particolare, la scelta di Java come supporto per la programmazione nell'implementazione del laboratorio remoto, si basa su valide motivazioni, prima tra tutte la sua semplicità di utilizzo rispetto a linguaggi quali C++, C ed altri. Inoltre Java è "machine-independent", nel senso che consente di creare programmi eseguibili su una grande varietà di computer che supportano differenti sistemi operativi, risolvendo il problema della portabilità; ancora, i programmi Java non sono eseguiti direttamente

dal sistema operativo, ma dalla JVM (Java Virtual Machine) risolvendo problemi di sicurezza ed accesso non autorizzato. Java, inoltre, si adatta bene alla realizzazione di applicazioni web interattive, basti pensare alla tecnologia delle "servlet", una delle punte di diamante della programmazione Java che è usata proprio nell'implementazione di On-Lab.

Le caratteristiche principali di Java

Indipendenza dalla piattaforma:

Le istruzioni binarie di Java indipendenti dalla piattaforma sono più comunemente conosciute come Bytecodes. Il Bytecodes di java è prodotto dal compilatore e necessita di uno strato di software, la Java Virtual Machine (JVM), per essere eseguito. (Fig. 3.1)

La JVM è un programma scritto mediante un qualunque linguaggio di programmazione dipendente dalla piattaforma e traduce le istruzioni Java, nella forma di Bytecodes, in istruzioni native del processore locale.

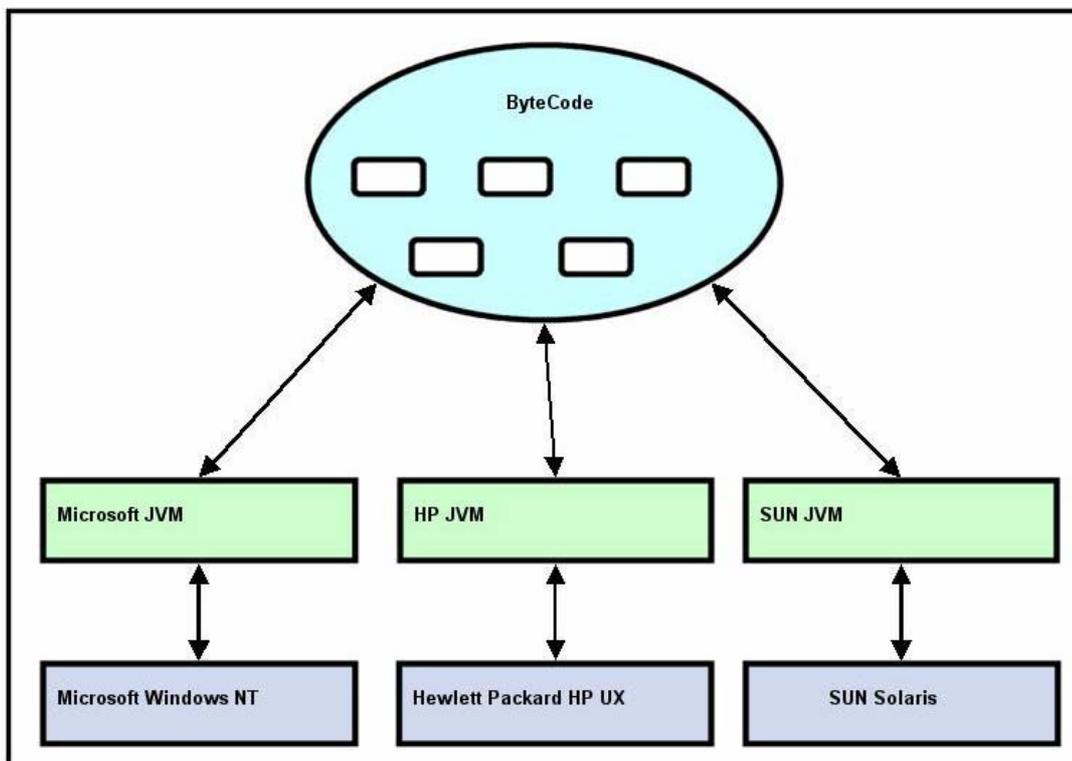


Fig. 3.1. Architettura di una applicazione Java

Non essendo il Bytecodes legato ad una particolare architettura Hardware, questo fa sì che per trasferire una applicazione Java da una piattaforma ad un'altra è necessario solamente che la nuova piattaforma sia dotata di una apposita JVM.

Servlet

La crescente richiesta di pagine web con contenuti dinamici e di servizi su web, ha portato allo sviluppo di una particolare tipologia di software che collabora con il server web per estenderne le funzionalità e poter interagire con basi di dati. Le tecniche per realizzare questo tipo di software sono tante; tra di esse le più conosciute sono sicuramente le estensioni CGI (Common Gateway Interface), realizzate principalmente in C o in Perl. Anche Java mette a disposizione una API, le Servlet, appunto, che permette di sviluppare applicazioni lato server. Le Servlet sono moduli software scritti in Java che vengono eseguiti in applicazioni lato server per esaudire le richieste dei client. Esse non sono legate ad un particolare protocollo per la comunicazione tra client e server, anche se più comunemente si utilizza il protocollo HTTP ed infatti si parla di http Servlet.

Le Servlet vengono più comunemente utilizzate per:

- elaborare e salvare dati provenienti da form HTML;
- provvedere alla creazione di pagine HTML dinamiche, ad esempio utilizzando i dati contenuti in un database, in seguito ad una richiesta di un client;
- gestire informazioni con stato, ad esempio la gestione di un sistema di commercio elettronico, dove un certo numero di clienti fa degli acquisti contemporaneamente e dove occorre quindi mantenere in modo corretto le informazioni sui clienti e sui loro acquisti (il classico carrello della spesa);

Servlet vs CGI

Le Servlet hanno diversi vantaggi rispetto al tradizionale CGI e precisamente sono più efficienti, più facili da utilizzare, più potenti, più portabili e più economiche. Vediamo più in dettaglio ciascun punto:

- *Efficienza:* Con una tradizionale applicazione CGI, viene generato un processo (istanza del programma o script CGI) per ogni richiesta che arriva al server, questo tipo di operazione può risultare abbastanza pesante in termini di risorse. Con le Servlet invece, la JVM (Java Virtual Machine) genera, per ogni richiesta da esaudire, un Thread Java, molto più leggero (lightweight) di un processo generato dal sistema operativo: con il CGI se ci sono N richieste contemporanee, esistono N immagini in memoria del programma CGI; con le Servlet invece, ci sono N Thread ma una sola copia della classe Servlet in memoria. Inoltre c'è da dire che l'efficienza è ancora superiore se si pensa che, ad esempio, un'applicazione CGI che si interfaccia con un database si connette ad esso ad ogni richiesta che arriva e si disconnette al termine: operazioni queste molto pesanti. Con le Servlet invece è possibile mantenere aperte una o più connessioni al database, utilizzando ad esempio una Connection Pool, e riutilizzarle da richiesta a richiesta.
- *Facilità d'uso:* Java mette a disposizione una API per la scrittura delle servlet che facilita molto le varie operazioni coinvolte nell'utilizzo di una Servlet e cioè: manipolazione dei dati provenienti dai form HTML, lettura e gestione delle intestazioni (headers) delle richieste http, gestione dei Cookie e delle sessioni e molte altre cose utili.
- *Potenza:* Con le servlet si possono fare tante cose che con i programmi CGI sono difficili o addirittura impossibili da realizzare: le servlet possono dialogare direttamente con il server web facilitando l'utilizzo e la condivisione di dati (testi, immagini ecc.). Inoltre le servlet facilitano operazioni come la session tracking, permettendo il mantenimento di informazioni tra una richiesta e un'altra, e di caching delle operazioni già eseguite in una richiesta precedente.
- *Portabilità:* Essendo scritte in Java mediante un'apposita API standard, le Servlet possono girare su qualsiasi piattaforma. Oggi le servlet sono supportate, direttamente o mediante plug-ins (servlet engines), dalla maggior parte dei server web in circolazione, commerciali e free.
- *Convenienza:* Molti server web che supportano le servlet sono free o comunque hanno un basso costo. Se si dispone già di un server web

commerciale che non supporta le servlet l'aggiornamento con appositi plug-ins per il supporto delle servlet è normalmente free.

Architettura di base di una Servlet

Una Servlet, nella sua forma più generale, è un'istanza della classe che implementa l'interfaccia `javax.servlet.Servlet`, e cioè `javax.servlet.GenericServlet`; ma come abbiamo già detto quella più utilizzata è quella che usa il protocollo HTTP e che si costruisce estendendo la classe `javax.servlet.http.HttpServlet`.

Struttura di una applicazione web

Una applicazione web è una gerarchia di file e directory disposti secondo uno schema standard.

/webapp. Una applicazione web ha una propria directory radice (*root*). Il nome della cartella corrisponde a ciò che nella terminologia delle servlet viene chiamato *context path* (nel seguito supporremo che tale nome di tale cartella sia *webapp*). La directory radice contiene tutti gli altri elementi che compongono l'applicazione web.

File HTML, JSP, ... La directory */webapp* contiene i documenti HTML, le pagine JSP, le immagini e le altre risorse che compongono l'applicazione web. Tali file possono essere strutturati in directory esattamente come si farebbe per un sito web statico.

/webapp/WEB-INF. All'interno della cartella *root* si trova una directory speciale, denominata *WEB-INF*. La funzione di questa cartella è quella di contenere file riservati, come ad esempio il file di configurazione dell'applicazione, *web.xml*. Per questo motivo le specifiche impongono che la cartella ed il suo contenuto debbano risultare inaccessibili per gli utenti dell'applicazione.

/webapp/WEB-INF/web.xml. Il file *web.xml* viene comunemente denominato *deployment descriptor*. Si tratta del file di configurazione dell'applicazione web; in esso, ad esempio, si definiscono gli *alias* delle servlet, i parametri di inizializzazione, le mappature dei percorsi e così via.

/webapp/WEB-INF/classes e **/webapp/WEB-INF/lib**. Queste due directory sono destinate a contenere le classi e gli archivi Jar di cui la nostra applicazione web necessita. Se dobbiamo utilizzare delle classi Java compilate (file con estensione .class andremo a copiarle all'interno di /webapp/WEB-INF/classes, secondo l'usuale struttura di directory Java (così se abbiamo una classe MiaServlet contenuta nel package sito.servlet. il percorso del file compilato sarà /webapp/WEB-INF/classes/sito/servlet/MiaServlet.class). Se dobbiamo utilizzare classi contenute in archivi Jar, invece, sarà sufficiente copiarli in /webapp/WEB-INF/lib.

Tomcat: il contenitore delle servlet Java

Tomcat contiene al suo interno tutte le funzionalità tipiche di un web server, ovvero - in sintesi - ha la capacità di interpretare una richiesta di una risorsa veicolata su protocollo HTTP, indirizzarla ad un opportuno gestore (o prenderla dal filesystem) e restituire poi il risultato (codice HTML o contenuto multimediale che sia). La caratteristica innovativa di Tomcat non è comunque quella di essere un web server, ma piuttosto quella di fornire allo sviluppatore un vero e proprio ambiente nel quale girano applicazioni Java (le servlet).

Quando un utente accede al web server accede in realtà ad una sezione specifica del web server (a un certo dominio virtuale o directory, ad esempio) identificata sostanzialmente dalla URL. Nel gergo di Tomcat la sezione viene chiamata contesto (*context*). All'interno di questo contenitore si troveranno quindi oggetti comuni a tutta la sezione, che saranno gli stessi per tutti gli utenti e tutte le applicazioni del contesto. Questo contenitore viene generato all'avvio di Tomcat e rimane attivo indipendentemente dal fatto che qualcuno lo stia interrogando in quel momento.

Il primo contenitore con caratteristiche fortemente orientate al web è quello associato alla sessione (*Session*). Come dice il nome questo contenitore viene associato ad un utente (o, meglio, ad un client) per tutto il tempo per il quale rimane connesso ad un contesto specifico. Al suo interno verranno memorizzati oggetti che saranno accessibili solo a quell'utente e ne personalizzeranno l'ambiente rispetto ad altri client che utilizzino risorse dello stesso contesto nello stesso momento.

Il contenitore con la vita più breve è la richiesta (*Request*) che nasce e muore nell'ambito della singola transazione web. Raramente questo contenitore viene riempito da applicazioni anche se svolge un ruolo fondamentale nel flusso dei dati che attraversano Tomcat. [17]

3.3 Progettazione ed implementazione di On-Lab

Nella progettazione di un laboratorio on line occorre tenere conto di tre punti chiave: modularità, riusabilità e gestione di interfaccia comune.

- Modularità: differenti moduli possono avere un insieme dedicato di agenti con un insieme definito di task, tali moduli possono essere facilmente sostituiti, aggiornati e aumentati;
- Riusabilità: possibilità di riutilizzare le definizioni di oggetti nel sistema in diversi modi: ereditarietà, aggregazione ecc.;
- Gestione di interfaccia comune: le definizioni di interfacce comuni che possono gestire virtualmente ogni tipo di risorsa che possa essere condivisa nel laboratorio virtuale.

On-Lab rispetta tutti i punti sopra descritti, infatti utilizza classi java indipendenti che possono essere aggiornate o sostituite in qualsiasi momento, organizzate gerarchicamente secondo la logica della programmazione ad oggetti. Sono presenti dunque classi specifiche che si occupano di un singolo aspetto implementativo, e classi comuni che mettono a disposizione le proprie procedure per tutta l'applicazione.

Per interagire con questi laboratori e consentire accesso a tutti gli studenti, deve essere necessariamente utilizzata Internet, che è una rete pubblica non protetta. In questo caso uno dei problemi è lo sviluppo di funzioni sicure e protette, per non compromettere la stabilità e la sicurezza dell'intero sistema. Nel caso specifico deve essere gestita la sicurezza di tutta l'applicazione, un filtro per la gestione dei comandi da eseguire e l'autenticazione per evitare accessi indesiderati dall'esterno.

Architettura

L'architettura di On-Lab ((Fig. 3.2)) consiste di tre componenti principali: L'interfaccia di strumento (IS), il Livello di Distribuzione (LD) ed il Client (CL).

L'interfaccia di strumento (IS)

Il compito dell'Interfaccia di strumento è quello di guidare uno strumento reale.

Nel caso di On-Lab è una classe java che si occupa di comunicare allo strumento il comando da eseguire, attendere l'output ed inglobarlo in una pagina web.

Il Livello di Distribuzione (LD)

Per quanto riguarda il livello di distribuzione, On-Lab è basato su un server http che serve da proxy per impartire comandi e ricevere risposte. Una volta che On-Lab riceve i comandi da un utente, esso usa connessioni dirette ai dispositivi, esegue quindi il comando specificato dall'utente sullo strumento e resta in attesa dell'output. Non appena riceve l'output, crea automaticamente il codice HTML, lo inserisce nella pagina di risposta e lo manda indietro al client.

Il Client (CL)

Il Client è costituito dall'utente che si connette con On-Lab tramite internet. Per connettersi con On-Lab è sufficiente utilizzare un normale web browser.

On-Lab è stato progettato per girare su una macchina con sistema operativo Unix-based, come, ad esempio, Linux. Può anche essere installato su una macchina windows, in quanto per l'implementazione si è scelto di utilizzare Java e le servlet, che, presentando un funzionamento basato su una macchina virtuale (Java Virtual Machine – JVM), sono indipendenti dalla piattaforma. La scelta di utilizzare Linux quale sistema operativo è sembrata naturale, dal momento che è il sistema operativo che offre maggiori opportunità didattiche nello studio di strumenti quale router, firewall (ad esempio con iptables od il vecchio ipchains), od anche semplici esercizi su comandi da shell.

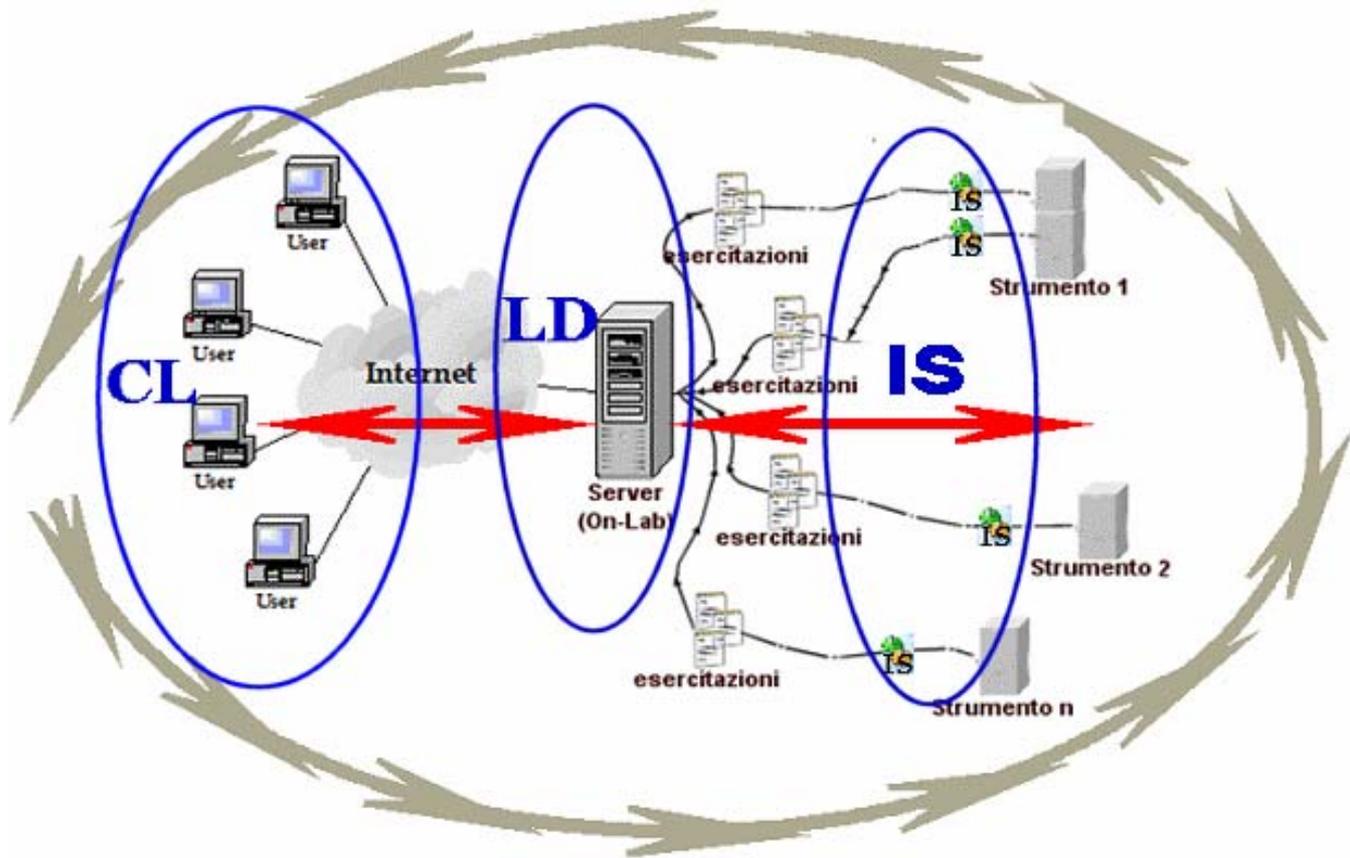


Fig. 3.2 On-Lab. Architettura generale

Flusso dinamico

On-Lab va installato sul server. Requisito necessario al suo funzionamento è l'installazione sul server di un servlet-container (Tomcat) e di Java. Il principio di funzionamento base è il seguente:

- Il client (CL) (tramite internet) si collega al server e, tramite la pagina di login, invia i dati di autenticazione.
- Il server (SRV) verifica i dati di autenticazione inviati, e, se corretti, rimanda al CL l'home page utente corretta.

- Il CL manda al SRV la richiesta di eseguire una particolare esercitazione (EX) ad esempio basata su uno strumento fisico o virtuale (S).
- Il SRV costruisce automaticamente la pagina HTML dell'EX leggendo i parametri dal file dell'EX e la rimanda al CL.
- Il CL esegue un comando e lo invia al SRV.
- Il SRV tramite l'Interfaccia di strumento (IS) controlla se il comando è valido e consentito, in caso affermativo, il comando viene inviato allo S e resta in attesa. Non appena lo S termina l'esecuzione del comando, il SRV richiama l'output e lo inserisce nella pagina HTML dell'EX e la rimanda al CL con il punteggio aggiornato. Nel caso in cui il comando non è corretto o non è permesso, il S manda invece la pagina al CL con un errore al posto dell'output del comando.

On-Lab dunque fa da filtro per consentire al client di comunicare direttamente con lo strumento.

Nella figura della pagina successiva (Fig. 3.3) viene schematizzato quanto detto sopra.

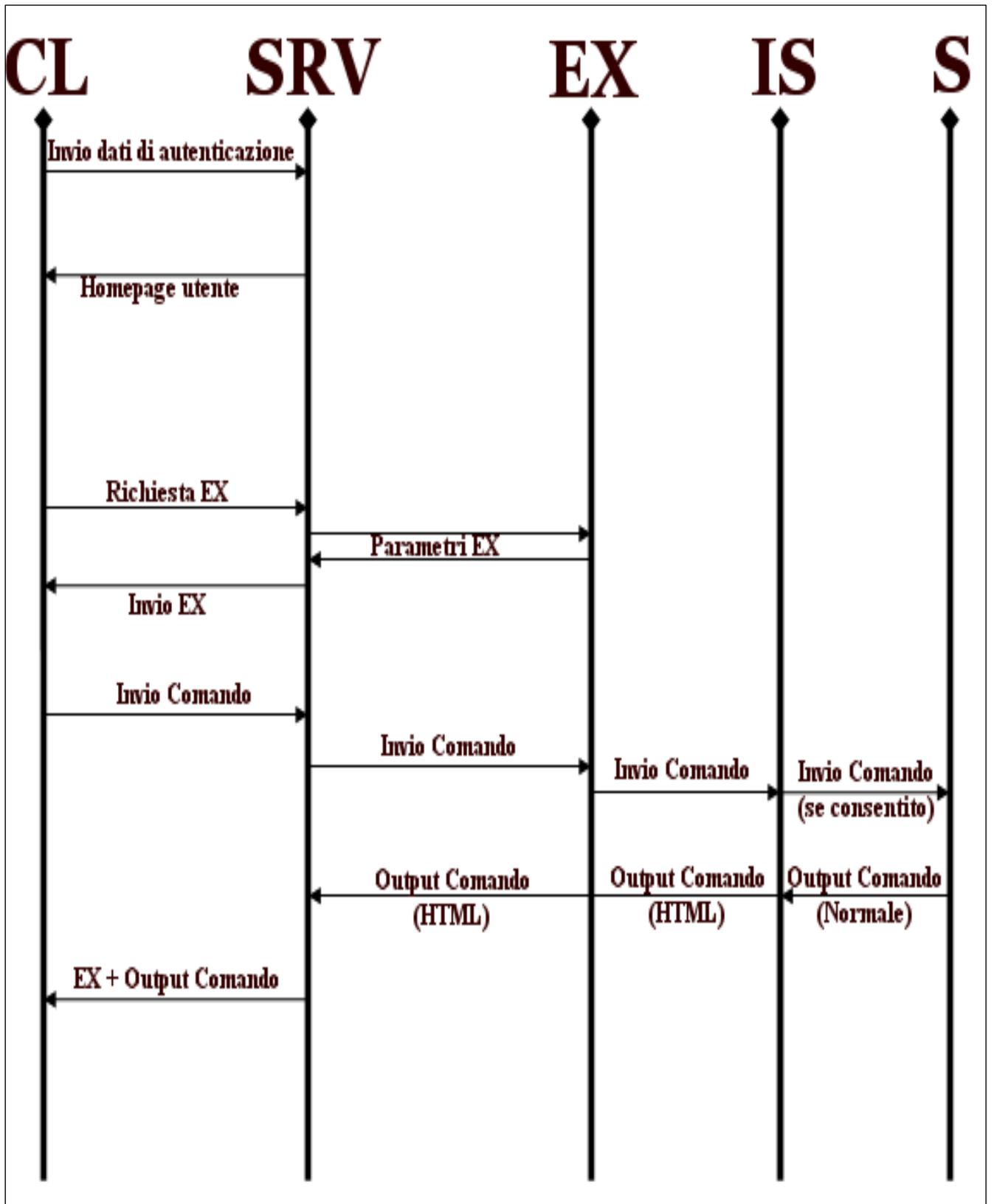


Fig. 3.3– Flusso Dinamico

Aspetti implementativi

L'architettura di base dei file di configurazione ed autenticazione di On-Lab è la seguente:

- */root/vlab:*
È la directory principale. Contiene le directory docente, studente ed amministratore, ed inoltre "executor", lo script che viene usato per eseguire fisicamente i comandi sulle interfacce.
- */root/vlab/Amministratore:*
Contiene il file password dell'amministratore ed una directory dove sono definite le varie interfacce.
- */root/vlab/Amministratore/ifs:*
Questa directory contiene la definizione delle varie interfacce. Contiene un'altra sottodirectory dove sono memorizzate le immagini eventualmente associate all'output dei comandi delle interfacce virtuali.
- */root/vlab/Docente:*
Contiene il file password dei docenti, e delle sottodirectory (una per ogni docente) che contengono le varie esercitazioni create.
- */root/vlab/Studente:*
Contiene il file password degli studenti, e delle sottodirectory (una per ogni studente) che contengono le varie esercitazioni che lo studente ha parzialmente completato.

L'intera applicazione è formata dalle seguenti classi:

- *Fileio.class*
Si occupa della memorizzazione o cancellazione dei file e dei cookie usati dall'applicazione. È una classe stand-alone utilizzata esclusivamente per

richiamare procedure e funzioni frequentemente utilizzate, non vi è l'implementazione all'interno dei metodi doGet() e doPost().

- **Cancella.class**

È utilizzata per dare un responso web ad alcune operazioni di cancellazione. Utilizza i metodi di fileio.class.

- **Errore.class**

Utilizzata per reindirizzare l'utente che compie operazioni non valide. Accetta come parametro semplicemente la descrizione dell'errore.

- **GeneratoreQuestionari.class**

È la classe principale. Fondamentalmente accetta come parametri i dati utente ed il puntatore al file di configurazione dell'esercitazione richiesta. In base ai parametri letti dal file costruisce adeguatamente l'esercitazione, con il layout scelto, le domande e le risposte corrette. Legge le risposte dinamicamente dal file, per evitare che lo studente possa leggerle prima di rispondere.

- **NuovoUtente.class**

È la classe utilizzata per la creazione di un nuovo docente o studente.

La creazione di un nuovo docente è consentita esclusivamente all'amministratore, la creazione di un nuovo studente è consentita esclusivamente al docente.

- **Login.class**

Classe che si occupa di passare i parametri di login alla classe che si occupa dell'autenticazione (auth.class).

- **Auth.class**

Come precedentemente accennato, questa classe si occupa dell'autenticazione degli studenti, dei docenti e degli amministratori. Data la scarsa necessità di sicurezza delle password, non è stato implementato alcun

algoritmo di cifratura delle password. Esse sono memorizzate in chiaro nel file contenente i dati utente.

- HomeDocente.class, HomeStudiante.class, admin.class
Home page rispettivamente di docenti, studenti e amministratore.
- PopUpPage.class
Viene richiamata da un javascript quando lo studente richiede un aiuto, legge la risposta dal file dell'esercitazione, estrae e mostra l'aiuto corretto, e decrementa opportunamente il punteggio conseguito.
- Punteggi.class
Utilizzata semplicemente per contare il punteggio degli studenti ed alleggerire così il carico di GeneratoreQuestionari.class
- Questionari.class, InterfacciaFisica.class, InterfacciaVirtuale.class
Tre classi utilizzate per creare rispettivamente un nuovo questionario, una nuova interfaccia fisica e una nuova interfaccia virtuale.

Si descrivono ora i dettagli delle classi più importanti.

GeneratoreQuestionari.class

Inizialmente questa classe esegue alcuni controlli per stabilire se è uno studente che sta accedendo all'esercitazione, oppure se è un docente che la sta visualizzando. Questo controllo è necessario per evitare che un docente possa rispondere a qualche domanda non compilata dallo studente ed è effettuato tramite i cookie di sessione. I cookie sono essenziali anche per evitare che qualche studente possa sostituirsi ad un altro utente. Sono stati scelti i cookie di sessione senza alcuna scadenza, la sessione utente quindi durerà fino alla chiusura del browser web. Tutta la struttura dell'esercitazione viene memorizzata in una variabile (Body).

La pagina Web sarà costruita leggendo le informazioni presenti in quella variabile. Non esistono riferimenti diretti alle risposte delle domande, ma solo puntatori al file, per evitare che lo studente possa “barare” e rispondere correttamente guardando il sorgente della pagina.

In questa classe è anche presente la routine che si occupa, dopo aver verificato che a rispondere è lo studente corretto, di aggiungere la risposta al file temporaneo, solamente se la stessa non è già presente (per evitare che lo studente possa “correggere” la risposta data premendo back nel browser).

Sempre nella stessa classe è implementata la routine di controllo che si occupa dell’esecuzione dei comandi lanciati o dal textbox presente in ogni esercitazione e che consente di testare un comando sull’interfaccia o da uno dei textbox usati per dare la risposta. Dopo un’attenta verifica sugli effettivi permessi di esecuzione del comando, esegue fisicamente il comando (nel caso di un questionario costruito su uno strumento fisico) ed attende l’output. Terminata l’esecuzione, l’output viene catturato ed inserito all’interno di una textarea. Dell’esecuzione e della cattura dell’output si occupa la classe **fileio.class**.

Il controllo sul comando viene eseguito in base ai comandi definiti al momento della creazione dell’interfaccia e ai comandi definiti nel questionario. Il filtro comandi è molto rigido, non vi è quindi alcun rischio per la sicurezza del sistema, tuttavia il docente al momento della creazione dell’esercitazione e, soprattutto, l’amministratore al momento della creazione dell’interfaccia, devono prestare molta attenzione a non definire comandi che possano essere potenzialmente pericolosi per la sicurezza del sistema.

Fileio.class

Questa classe si occupa non solamente della memorizzazione e cancellazione dei file, come il nome potrebbe indurre a pensare, ma si occupa anche della gestione dei cookie e dell’esecuzione dei comandi nonché della cattura del relativo output. Il comando da eseguire viene passato come parametro ad una routine della classe, che lo inserisce in un file script (/root/vlab/executor), lo esegue, attende la fine dell’esecuzione e ritorna il controllo alla procedura chiamante. Lo script si occupa di

reindirizzare l'output e l'errore (*stdout* ed *stderr*) in un file, che verrà poi inserito all'interno della textarea. Chiaramente la classe si occupa anche di tutte le operazioni di lettura e scrittura su file, ed inoltre legge e scrive tutti i cookie di sessione utilizzati.

Questionari.class

Questa classe è usata per creare le esercitazioni. Durante la costruzione dell'esercitazione, viene allocato un file temporaneo in modo da consentire di poter ultimare la costruzione dell'esercitazione in un secondo momento.

Autenticazione

On-Lab gestisce un sistema di autenticazione a tre livelli:

Amministratore

- Crea/Modifica/Elimina le interfacce
- Crea/Elimina i docenti

Docente

- Crea/Elimina Studenti
- Crea/Modifica/Elimina i questionari

Studente

- Esegue i questionari
- Visualizza i punteggi ottenuti

Per la gestione dell'autenticazione si è scelto di utilizzare un file password con i dati utente separati da un carattere ("§"). Il metodo di memorizzazione utenti risulta quindi simile al file */usr/passwd* di Unix. Poiché l'applicazione risiederà su una macchina dedicata, accessibile solamente da remoto con ssh (conoscendo ovviamente la password di root) e poiché comunque non è necessaria un'eccessiva sicurezza delle password utente, si è scelto di non criptare le password utente, le quali saranno

quindi memorizzate in chiaro nel file delle password. Nel caso di necessità particolari comunque, non sarà difficile aggiungere un modulo per criptare le password, poiché la procedura di autenticazione è stata creata in una classe a sé stante (**auth.class**), che come input riceve solamente i parametri di autenticazione. Sarà necessario quindi unicamente creare una nuova classe che si occupa della cifratura delle password e passare come parametro alla classe di autenticazione la password cifrata al posto di quella in chiaro.

In Appendice si riporta il codice delle principali classi implementate.

4. Esempi di utilizzo di On-Lab

L'utilizzo di On-lab è semplice ed immediato. Inizialmente sarà compito dell'amministratore (Login di default: user: admin, password: cambialasubito) definire e creare le interfacce che saranno alla base della creazione delle esercitazioni.

Per consentire una maggiore flessibilità, esiste la possibilità di creare due tipologie di interfacce: interfaccia virtuale e interfaccia fisica.

Creazione di un'interfaccia VIRTUALE

Si tratta di un'interfaccia totalmente libera ed eterogenea, che consente la creazione di questionari di qualsiasi argomento e materia (Fig. 4.1).

Per utilizzarla, occorre:

- specificare il comando da eseguire;
- inserire una breve descrizione (che verrà visualizzata assieme all'output);
- inserire l'output in modalità testuale.

È anche consentito associare un output visivo a quello testuale, eseguendo l'upload di un'immagine.

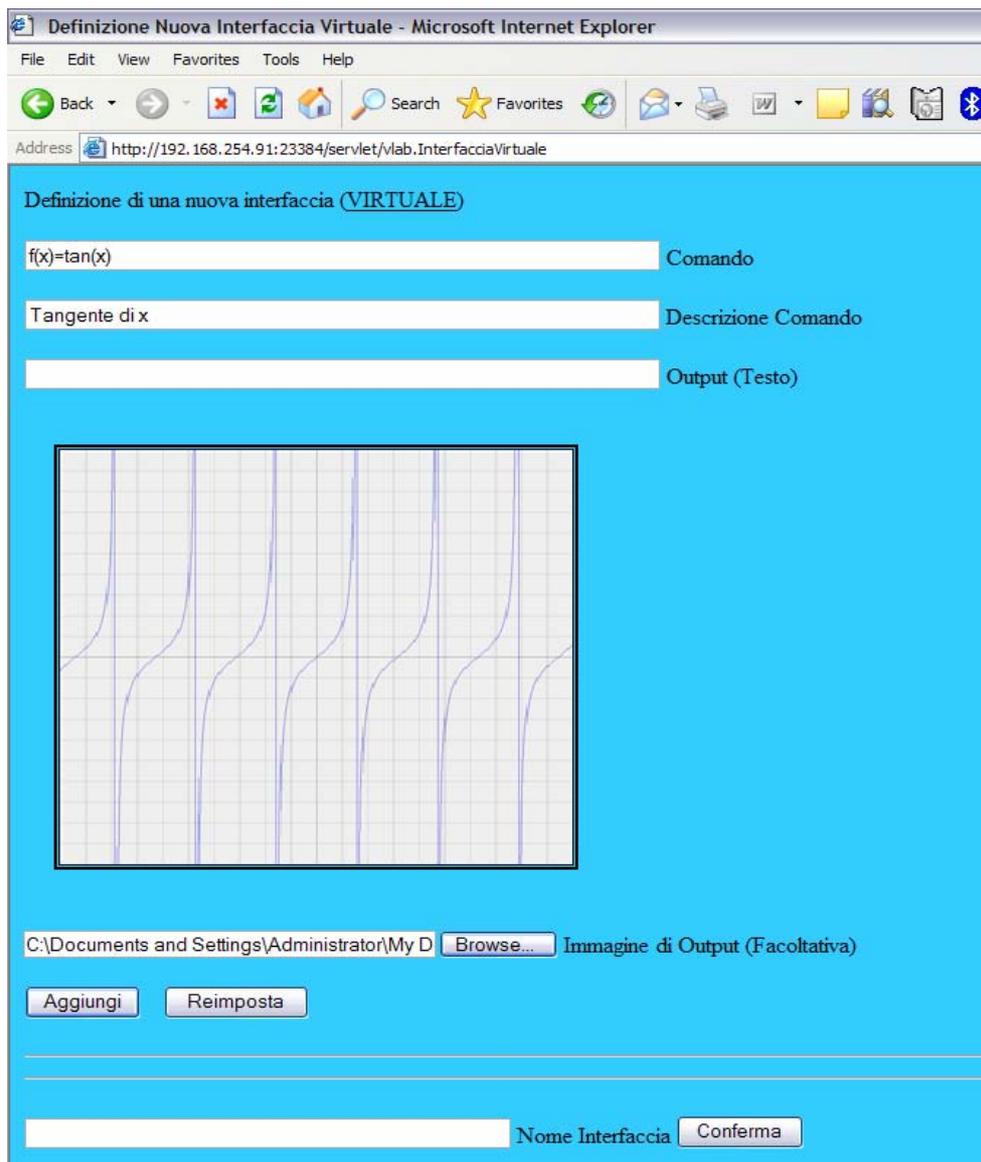


Fig. 4.1 – On-lab. Creazione di un'interfaccia virtuale.

Creazione di un'interfaccia FISICA

Si tratta di un'interfaccia ad un'apparecchiatura reale; per utilizzarla è necessario solamente specificare il comando da eseguire (Fig. 4.2).

Durante l'esecuzione del questionario, se lo studente fornisce la risposta corretta, il comando verrà realmente eseguito, attraverso l'interfaccia, sull'apparecchiatura e l'output verrà poi catturato e riportato nella pagina del questionario.

I comandi specificati al momento della creazione dell'interfaccia verranno rigorosamente filtrati, non sarà possibile ad esempio per lo studente aggiungere un parametro a sua scelta al comando, se questo parametro non è stato aggiunto dal docente al momento della creazione dell'esercitazione.



Fig. 4.2 – On-lab. Creazione di un'interfaccia fisica.

Creazione di un'esercitazione

Il docente troverà nella propria homepage tutte le interfacce definite dall'amministratore (Fig. 4.3) e sarà libero di creare tutte le esercitazioni di cui necessita, basate sulle interfacce definite. Egli dovrà solamente selezionare l'interfaccia interessata e cliccare su "crea" per iniziare la compilazione del proprio questionario personalizzato.



Fig. 4.3 – On-Lab. HomePage Docente.

La creazione dei questionari (Fig. 4.4 e 4.5) è molto semplice ed intuitiva. È consentito specificare la domanda, gli eventuali parametri (opzione non consentita per un questionario basato su interfaccia virtuale) e decidere se attivare o meno due tipi di aiuti (Aiuto 1 suggerisce il comando, Aiuto 2 suggerisce i parametri relativi al comando).

È possibile specificare il punteggio da assegnare alla domanda e il punteggio da detrarre in caso di un'eventuale richiesta di aiuto. È anche possibile variare il layout del questionario decidendo colore di sfondo e colore del testo ed assegnando un'intestazione (che poi diventerà anche il nome del questionario) e un piè di pagina. È infine possibile decidere se il questionario sarà di "autovalutazione" e, quindi, se lo studente potrà eseguirlo a suo piacimento tutte le volte che lo riterrà necessario, oppure bloccarlo in modo da consentire una sola esecuzione cosicché il docente possa rendersi conto del lavoro eseguito dallo studente.

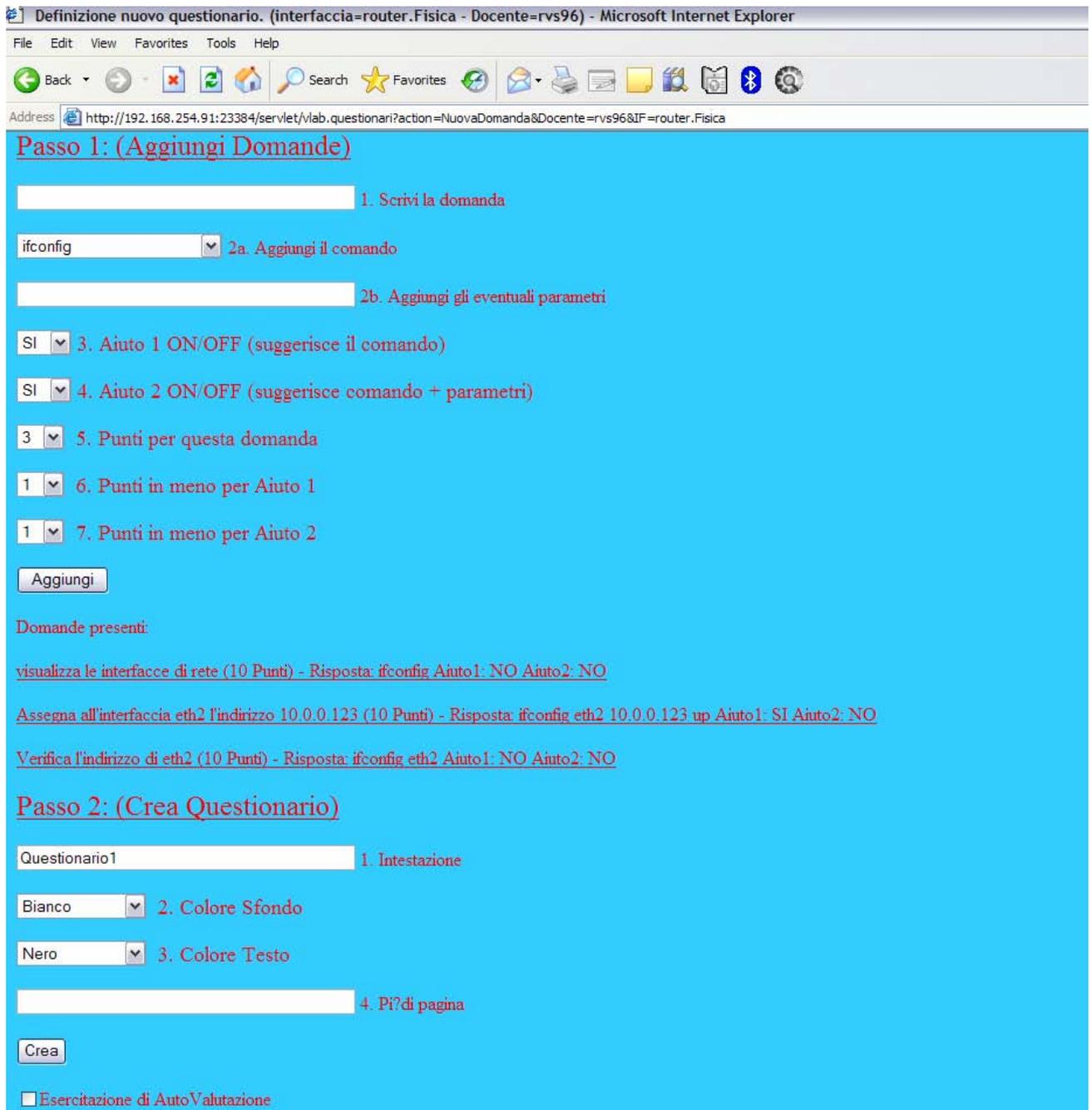


Fig. 4.4 – On-Lab. Creazione questionario, interfaccia fisica.

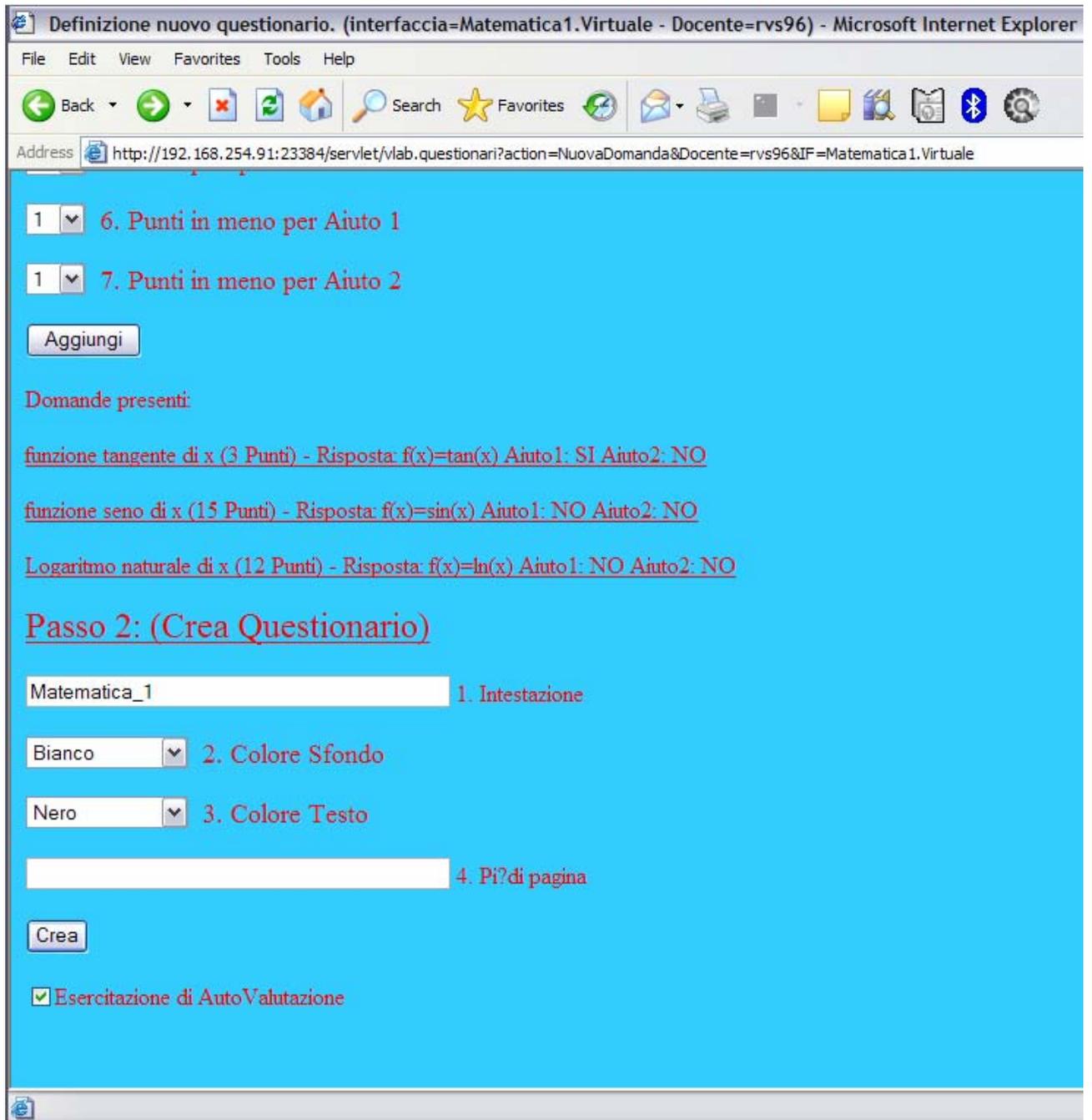


Fig. 4.5 – On-Lab. Creazione questionario, interfaccia Virtuale.

Esecuzione di un'esercitazione

Lo studente vedrà comparire automaticamente nella propria homepage tutte le esercitazioni create dal docente con cui si è registrato (Fig. 4.6)

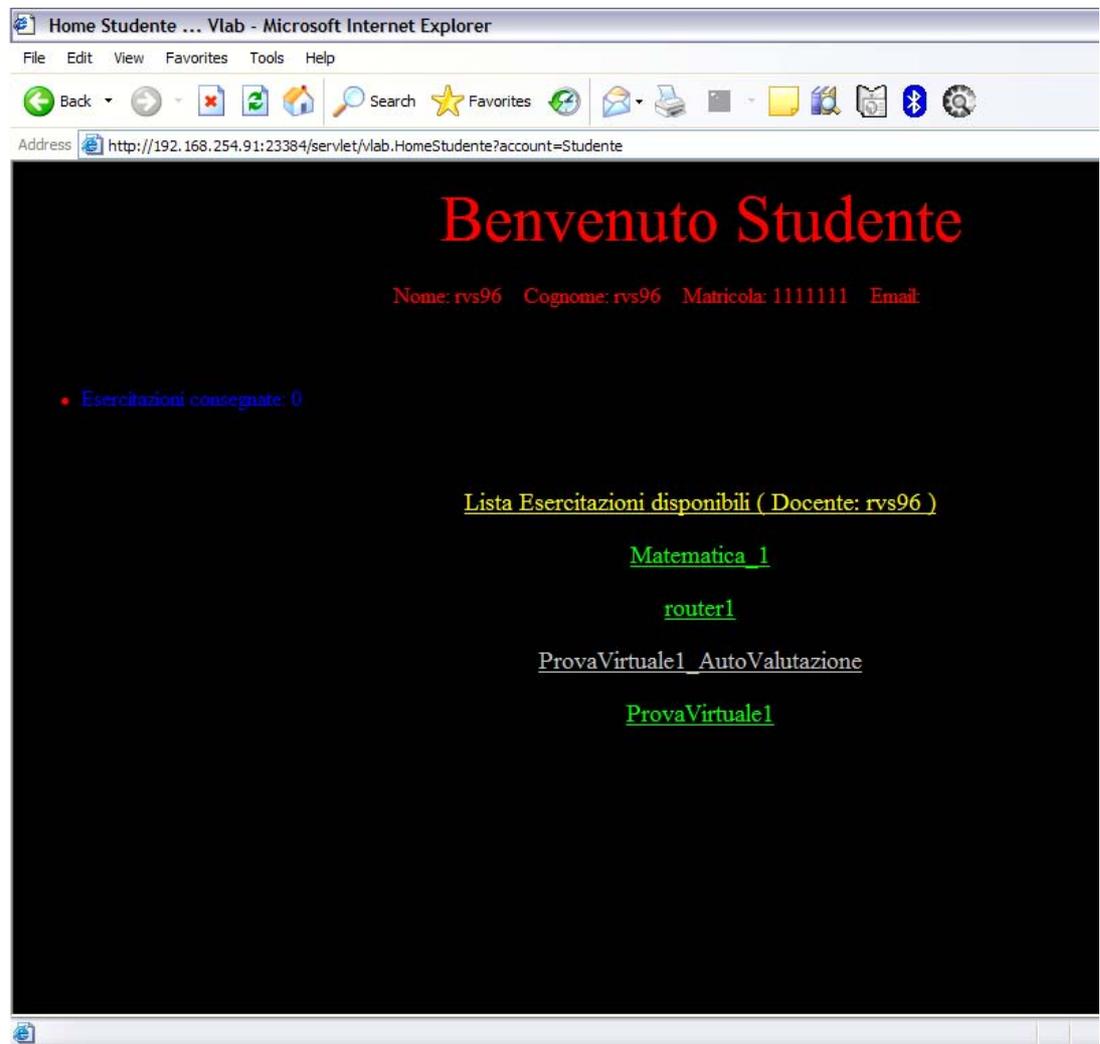


Fig. 4.6 – On-Lab. HomePage Studente.

In Fig. 4.7a e 4.7b si vede il risultato della creazione di due questionari di esempio. Il primo è stato creato su un'interfaccia virtuale come questionario di autovalutazione. Lo studente può eseguirlo tutte le volte che vuole, semplicemente premendo il pulsante "Resetta Questionario". Il docente non sarà mai in grado di vedere quante volte lo studente ha svolto l'esercitazione o il punteggio ottenuto.

Il secondo, creato su un'interfaccia fisica, è stato invece "bloccato". Lo studente ha, quindi, la possibilità di svolgerlo una volta soltanto e alla fine può "Consegnarlo" al docente, che lo vedrà comparire nella propria homepage (Fig. 4.8) e potrà così verificare il livello di apprendimento conseguito dallo studente.

Matematica_1 (Matematica1.Virtuale)

Punti Totalizzati: 0

Domanda 0

funzione tangente di x (?)

Ok Aiuto1 Aiuto2

Domanda 1

funzione seno di x (?)

Ok Aiuto1 Aiuto2

Domanda 2

Logaritmo naturale di x (?)

Ok Aiuto1 Aiuto2

Resetta Questionario

Fig. 4.7a. On-Lab. Esempio di questionario su interfaccia virtuale

router1 (router.Fisica)

Punti Totalizzati: 0

Domanda 0

visualizza le interfacce di rete (?)

Ok Aiuto1 Aiuto2

Domanda 1

Assegna all'interfaccia eth2 l'indirizzo 10.0.0.123 (?)

Ok Aiuto1 Aiuto2

Domanda 2

Verifica l'indirizzo di eth2 (?)

Ok Aiuto1 Aiuto2

Consegna Questionario

Testa Comando

Fig. 4.7b. On-Lab. Esempio di questionario su interfaccia fisica.



Fig. 4.8 On-Lab. Lista di questionari consegnati dagli studenti.

Lo studente, se il docente ha abilitato l'opzione al momento della creazione del questionario, ha la possibilità di richiedere un aiuto per una domanda della quale non è sicuro sulla risposta.

Una finestra popup fornirà le informazioni richieste (Fig. 4.9) e provvederà a penalizzare lo studente con i punti associati all'aiuto richiesto.

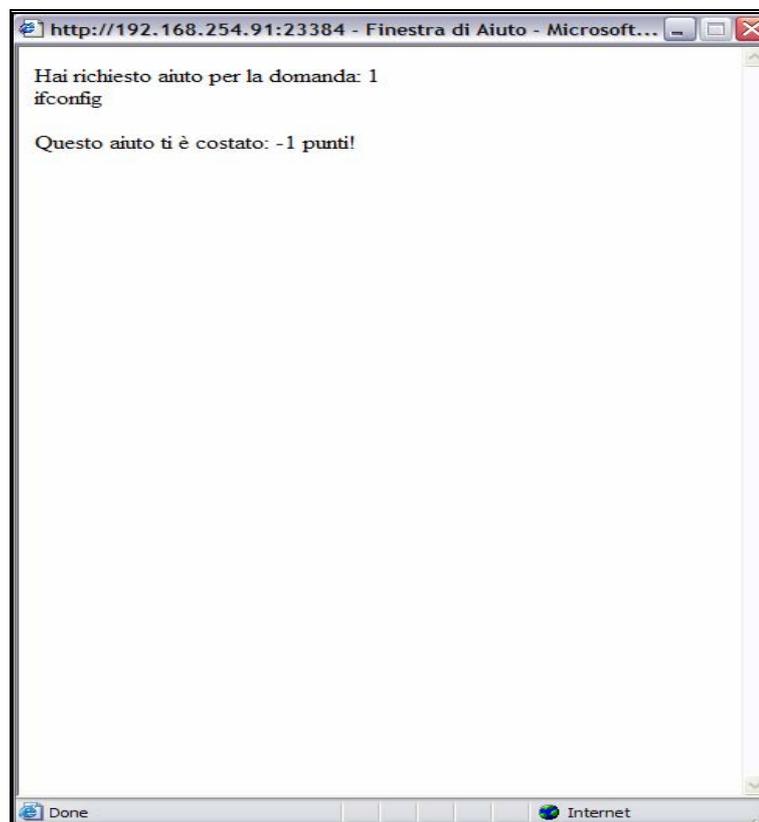


Fig. 4.9 – On-Lab. Esempio Aiuto1

Lo studente, nello svolgere l'esercitazione, dovrà scrivere la risposta che ritiene corretta nell'apposito campo e confermare con il pulsante ok. Se la risposta data è corretta, oppure errata ma il comando è nella lista dei comandi consentiti, l'output verrà visualizzato direttamente nella pagina web. È anche possibile testare un comando digitandolo nell'apposito campo, senza interferire col punteggio dell'esercitazione. Tale comando ovviamente verrà eseguito solo se si trova nella lista dei comandi permessi dall'interfaccia.

Lo studente potrà così rendersi effettivamente conto degli effetti che un determinato comando produce (Fig. 4.10a, 4.10b, 4.11 e 4.12).

```
OUTPUT RISPOSTA 0
```

```
eth0    Link encap:Ethernet HWaddr 00:0C:29:FE:22:8C
        inet addr:192.168.168.216 Bcast:192.168.168.255 Mask:255.255.255.0
        inet6 addr: fe80::20c:29ff:feff:228c/64 Scope:Link
        UP BROADCAST NOTRAILERS RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:9070 errors:0 dropped:0 overruns:0 frame:0
        TX packets:3024 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:1169403 (1.1 Mb) TX bytes:283332 (276.6 Kb)
        Interrupt:10 Base address:0x1080

eth1    Link encap:Ethernet HWaddr 00:0C:29:FE:22:96
        inet addr:192.168.254.91 Bcast:192.168.254.255 Mask:255.255.255.0
        inet6 addr: fe80::20c:29ff:feff:2296/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:2186 errors:0 dropped:0 overruns:0 frame:0
        TX packets:3085 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:276578 (270.0 Kb) TX bytes:3769959 (3.5 Mb)
        Interrupt:9 Base address:0x1400

eth2    Link encap:Ethernet HWaddr 00:0C:29:FE:22:A0
        inet addr:192.168.254.92 Bcast:192.168.254.255 Mask:255.255.255.0
        inet6 addr: fe80::20c:29ff:feff:22a0/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 b) TX bytes:460 (460.0 b)
        Interrupt:15 Base address:0x1480

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING MTU:16436 Metric:1
        RX packets:1199 errors:0 dropped:0 overruns:0 frame:0
        TX packets:1199 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:73712 (71.9 Kb) TX bytes:73712 (71.9 Kb)
```

Fig. 4.10a – On-Lab. Esempio di risposta ad un comando.



Fig. 4.10b – On-Lab. Esempio di risposta ad un comando di test.

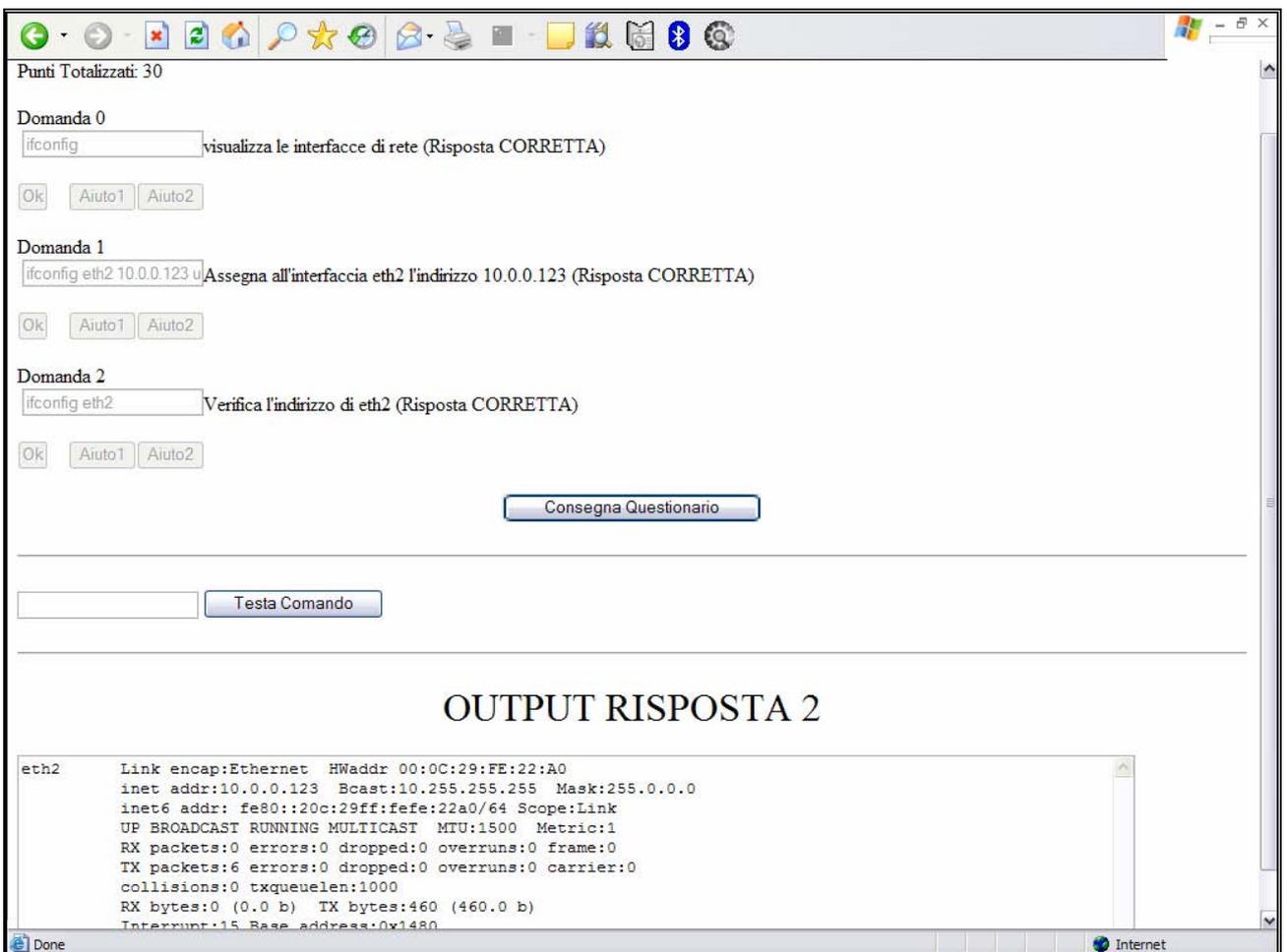


Fig. 4.11 – On-Lab. Esempio di risposta ad un comando.

Questionario - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://192.168.254.91:23384/servlet/vlab.GeneratoreQuestionari?NomeQuestionario=Matematica_1&Risposta=1&Docente=rvs96&Studiante=1111111

Ok Aiuto1 Aiuto2

f(x)=sin(x) funzione seno di x (Risposta CORRETTA)

Ok Aiuto1 Aiuto2

Logaritmo naturale di x (?)

Ok Aiuto1 Aiuto2

Resetta Questionario

OUTPUT RISPOSTA 1

NULL

(Seno di x)

Fig. 4.12 – On-Lab. Esempio di risposta ad un comando.

Dopo aver risposto a tutte le domande, lo studente non potrà più correggerle (Fig. 4.13), ma avrà la possibilità di rifare l'intero questionario, nel caso in cui si tratti di un questionario di autovalutazione, oppure di consegnarlo e il docente lo troverà nella lista delle esercitazioni eseguite. Inoltre, accanto ad ogni risposta fornita appare la scritta "risposta corretta" o "risposta errata" e, finito il questionario, all'alunno viene fornito il punteggio finale ottenuto.

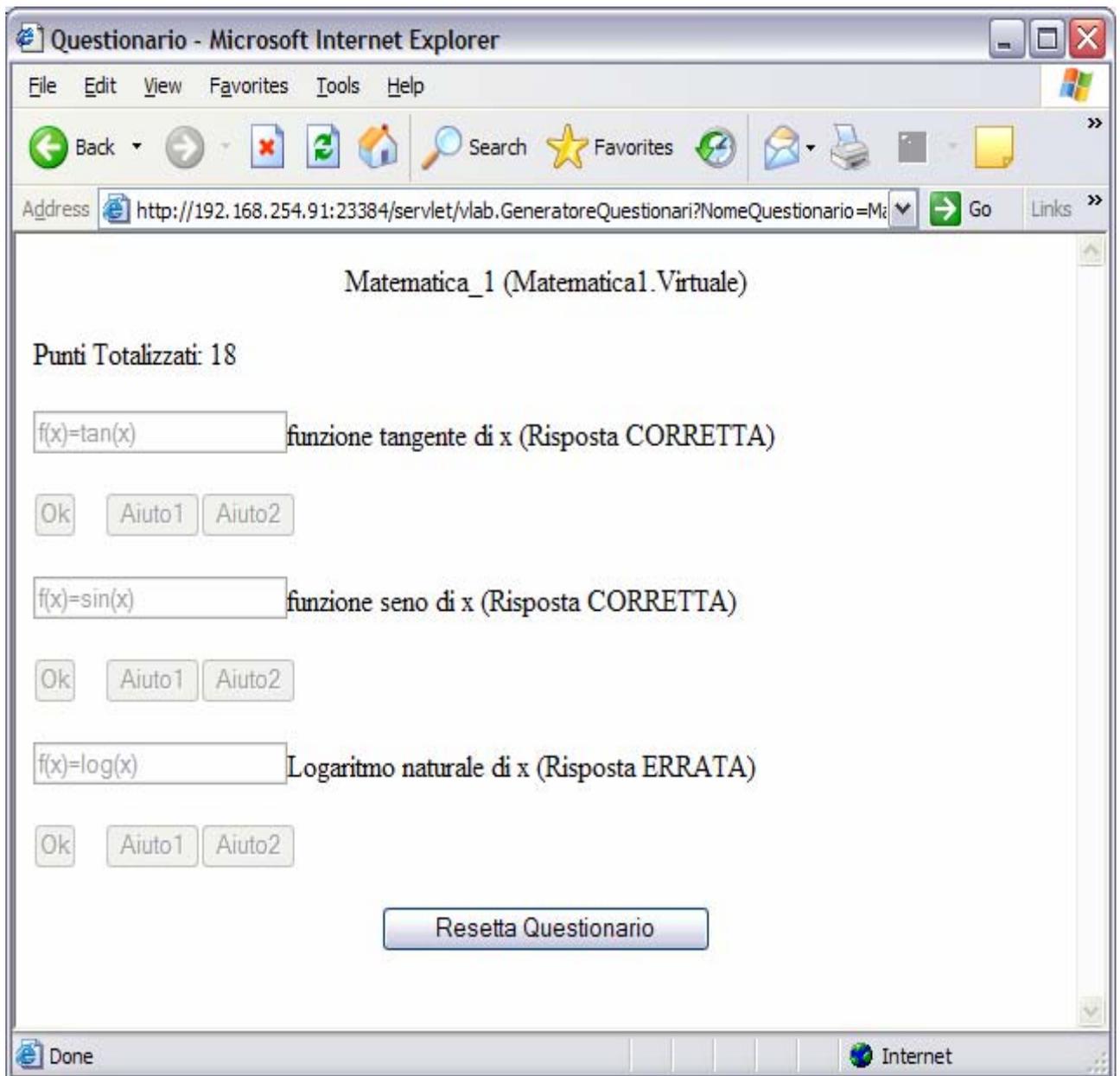


Fig. 4.13 – On-Lab. Questionario Completato.

Conclusioni

Nell'elaborazione di questo lavoro si è partiti dal concetto di didattica tradizionale, analizzandone i pregi ed i difetti e rapportandola con la didattica on line. Si è poi evidenziata l'importanza che il laboratorio assume nella didattica e si è analizzata la differenza fra i laboratori tradizionali fisici e i laboratori on line.

Si è notato, anche attraverso l'utilizzo e lo studio di soluzioni proposte da altri, che la flessibilità e la comodità di utilizzare tali strumenti didattici on line è notevole e si è mostrato che le impostazioni standard dei vari laboratori trovati in rete, seguono due vie principali:

- la prima prende la forma di questionari valutativi o di autovalutazione, che verificano la preparazione dello studente per mezzo di domande, quiz e quesiti.;
- la seconda consiste nell'implementare strumenti virtuali, oppure nell'implementare un'interfaccia (per lo più web) che consente di interagire con uno strumento fisico.

I vantaggi e gli svantaggi di entrambe le soluzioni sono chiari:

- nel primo caso lo studente verifica il proprio apprendimento sull'argomento mediante una sorta di esame on line che si avvicina molto alle tradizionali esercitazioni e il docente ha la possibilità di creare queste esercitazioni in maniera molto semplice e veloce, ma tale soluzione non fornisce alcun tipo di stimolo allo studente, in quanto molto simile ad un esame tradizionale;
- nel secondo caso, invece, lo studente si sente indubbiamente più coinvolto, in quanto ha la possibilità di interagire con gli strumenti, testare i comandi, visualizzare in tempo reale gli effetti. Di contro, però, la gestione e la creazione di un'interfaccia o di uno strumento virtuale non risulta immediata e semplice, e spesso il docente non ha le competenze tecniche per gestire il programma e, quindi, deve accontentarsi di utilizzare gli strumenti esistenti e poco, o per nulla, modificabili.

Per cercare di risolvere questi inconvenienti ed unire le due soluzioni che, comunque presentano entrambe aspetti importanti e in qualche modo si completano a vicenda, si è pensato di sviluppare la soluzione On-Lab.

On-Lab, grazie alla sua interfaccia web immediata e flessibile, consente di creare velocemente ed intuitivamente interfacce a strumenti fisici e virtuali e, ancor più velocemente, consente di creare esercitazioni basate sulle interfacce create.

Nello sviluppo del progetto si è cercato di rispettare alcuni punti chiave, principalmente:

- la modularità
- la sicurezza del sistema
- la semplicità di uso
- la flessibilità
- la personalizzazione dell'aspetto grafico delle esercitazioni.

La *modularità* è implementata grazie alle classi java che si occupano ognuna di un compito ben preciso, in modo da evitare un eccessivo carico da parte di una sola servlet e rendere così la risorsa disponibile a più utenti contemporaneamente.

La *sicurezza* è stata curata, grazie ad un filtro sui comandi eseguibili. Ovviamente questa sicurezza viene meno se l'amministratore che crea l'interfaccia per lo strumento fisico, consente l'esecuzione di comandi "pericolosi".

L'applicazione è *semplice* da utilizzare e molto *flessibile*:

- l'amministratore deve semplicemente inserire la lista dei comandi abilitati per l'interfaccia di strumento che sta creando.
- il docente deve solamente scegliere domanda e comando corretto (specificando, eventualmente, i parametri aggiuntivi). Può inoltre decidere il punteggio da assegnare ad una risposta corretta o da decrementare per una richiesta di un aiuto e può stabilire colore di sfondo e colore dei caratteri;
- lo studente può solamente eseguire le esercitazioni create dal docente. Per poterle eseguire deve semplicemente fare il login e selezionare l'esercitazione desiderata dalla lista presente nella propria homepage. Durante lo svolgimento dell'esercitazione, ha inoltre la possibilità (se il docente li ha attivati) di richiedere due tipi di aiuti: uno suggerisce il comando corretto, l'altro i parametri.

Da alcuni esperimenti effettuati, questa prima realizzazione è risultata molto stabile e funzionale, anche se alcuni aspetti possono essere rivisti nelle eventuali versioni successive, come ad esempio consentire una maggiore personalizzazione grafica delle esercitazioni, magari aggiungendo la possibilità di inserire immagini di abbellimento oppure inerenti alla domanda o anche una serie di template grafici di esercitazioni in modo che il docente possa variare l'aspetto dell'esercitazione con un solo click.

Un altro punto da approfondire potrebbe essere l'inserimento di un vero e proprio parser che analizza il comando inserito dall'utente, in modo da rendere meno rigido il controllo sui comandi inviati, senza però incidere sulla sicurezza del sistema che deve essere sempre curata attentamente.

Potrebbe essere infine implementata una pagina di statistiche, consultabile da parte dello studente, in modo che possa controllare l'andamento ed i risultati delle proprie esercitazioni.

Appendice – Codice java

Di seguito vengono riportate alcune procedure utilizzate nell'implementazione di On-Lab.

GeneratoreQuestionari.java

```
Boolean IsDocente = false;  
Cookie[] cookies = req.getCookies();  
String CookieStudente = "NESSUNO";  
String CookieDocente = "NESSUNO";  
if(cookies!=null)  
{  
    CookieStudente = fileio.getCookieValue(cookies,"CookieStudente","NESSUNO");  
    CookieDocente = fileio.getCookieValue(cookies,"CookieDocente","NESSUNO");  
}  
if (Docente.equals(CookieDocente))  
{  
    IsDocente=true;  
}  
else  
{  
    IsDocente=false;  
    if (!(Studente.equals(CookieStudente)))  
    {  
        res.setContentType("text/html");  
        res.setStatus(res.SC_MOVED_PERMANENTLY);  
        res.setHeader("Location","vlab.Errorre?err=ERRORE!:%20WRONG_SESSION");  
    }  
}
```



```

{ //INIZIO Aggiungi Risposta
    if (IsDocente){ //il docente non può rispondere alle domande!
        res.setContentType("text/html");
        res.setStatus(res.SC_MOVED_PERMANENTLY);
        res.setHeader("Location","vlab.Errorre?err=ERRORE!:%20NOT_ALLOWED");
    }
    else{ // IsDocente == false
        String RStud=req.getParameter("T"+Nr);
        if(RStud.trim().equals("")){
            RStud="Null";
        }
        String StrAdd=Nr+"§"+RStud;
        String[] TMPLN=null;
        Boolean Salvala=true;
        for(int k=0;k<=FReadRisposte.length-2;k++) {
            TMPLN=null;
            TMPLN=FReadRisposte[k].split("§");
            if(TMPLN[0].equals(Nr)){
                Salvala=false;
            }
        }
        if(Salvala) {
            FReadRisposte[FReadRisposte.length-1]=StrAdd;
            fileio.FileSave("/root/vlab/Studente/ex/"+Studente+"/tmp"+
                NomeQuestionario,FReadRisposte);
        }
    } //FINE IsDocente == false
} //FINE Aggiungi Risposta

```

/ Questa routine, dopo aver verificato che a rispondere è lo studente corretto, aggiunge la risposta al file temporaneo, se la stessa non è già presente (per evitare che lo studente possa "correggere" la risposta data premendo back nel browser) */*

```

if(Azione.equals("TestaComando")) {
    String RStud=req.getParameter("txtProvaComando");
    String Caption="OUTPUT TEST COMANDO";
    if(VerificaEsegui(IF,FRead,BodyAdd,Nr,RStud)){
        try{
            fileio.EseguiComando(RStud+" > /root/vlab/Amministratore/ifs/"+IF+".output");
        }
        catch(IOException ex){Caption="IO ERROR!";}
        String[] FOutput=fileio.FileRead("/root/vlab/Amministratore/ifs/"+IF+".output");
        BodyAdd="<p align=\"center\">"+"<font size=\"6\">"+
            Caption+"</font></p>"+
            "<p align=\"left\">";
        int i2=0;
        String Output="";
        try
        {
            while( (FOutput[i2] != null) )
            {
                Output+=FOutput[i2]+"\\n";// "\\n" perchè "<BR>" non verrebbe riconosciuto
                i2++; // come tag di carriage return, poichè la stringa
            } // "Output" viene letta da una TextArea
        }
        catch(ArrayIndexOutOfBoundsException ignored){};
        BodyAdd+="<p><textarea"+
            "rows=\""+Integer.toString(i2+1)+"\"name=\"txtArea1\""+
            " cols=\"110\" readonly>"+Output+"</textarea>";
    }
    else{// VerificaEsegui ha ritornato FALSE. Il comando non può essere eseguito.
        BodyAdd="<p align=\"center\">"+"<font size=\"6\""+
            ">COMANDO NON SUPPORTATO DA QUESTA INTERFACCIA</font></p>"+
            "<p align=\"left\">";
    }
}

```

```

    }
}

```

*/

Il controllo della pagina precedente si occupa dell'esecuzione dei comandi lanciati dal textbox presente in ogni esercitazione, che consente di testare un comando sull'interfaccia. Dopo un'attenta verifica sull'effettiva possibilità di esecuzione del comando (mediante la chiamata a "VerificaEsegui"), chiama la procedura EseguiComando ed intercetta l'output generato. La chiamata EseguiComando è definita nella classe fileio.class e serve appunto per lanciare fisicamente il comando.

*/

```

public Boolean VerificaEsegui(String IF, String[] FRead, String BodyAdd,
String Nr, String RStud){
if((IF.length())>7) &&(IF.substring(IF.length()-7,IF.length()).equals(".Fisica"))){
Boolean EseguiComando = false;
String[] FileIF=fileio.FileRead("/root/vlab/Amministratore/ifs/"+IF);
int x=0;
String Comando="";
String linea="";
String[] ArrayLinea = null;
try{
while( (! (linea=FileIF[x]).equals("<CMD")) ){ // Controllo sull'interfaccia
ArrayLinea=null;
linea="";
linea=FileIF[x];
ArrayLinea=linea.split("$");
Comando=ArrayLinea[0];
if(Comando.equals(RStud)) {
EseguiComando=true;
break; //Possiamo fermarci qui!
}
x++;
} //WEND
}
catch(ArrayIndexOutOfBoundsException ignored){};
x=0; //Continua nella pagina seguente

```

```

Comando="";
linea="";
try {
while( (linea=FRead[x]) != null ){ //Controllo sull'esercitazione
    ArrayLinea=null;
    ArrayLinea=linea.split("$");
    if( !(ArrayLinea[0].equals("Layout")) ) { //Escludiamo la parte che si occupa
        Comando=ArrayLinea[0];           //del LayOut
        if(Comando.equals(RStud)) {
            EseguiComando=true;
            break; //Possiamo fermarci qui!
        }
    }
    x++;
} //WEND
}
catch(ArrayIndexOutOfBoundsException ignored){};
if(EseguiComando){
    return(true);
}
return(false);
} //ENDIF (verifica tipo interfaccia)
return(false); //INUTILE, ma necessario perché richiesto dal parser Java!
}
/*
Questa procedura, dopo aver verificato che l'interfaccia sia fisica, controlla prima fra i comandi dell'interfaccia,
poi fra quelli dell'esercitazione (il docente può, volendo aggiungere parametri ad un comando definito
nell'interfaccia) se il comando da eseguire è lecito. Se in uno dei due cicli trova una risposta affermativa, allora
ritorna true ed esce, altrimenti, dopo aver esaurito i cicli, esce ritornando false.
*/

```

fileio.java

```
public static void EseguiComando(String Cmd)
```

```
    throws IOException
```

```
{
```

```
    FileSaveLn("/root/vlab/executor",Cmd);
```

```
    String Comando="/root/vlab/executor &";
```

```
    Runtime r=Runtime.getRuntime();
```

```
    Process proc=r.exec(Comando);
```

```
    try {
```

```
        proc.waitFor();
```

```
    }
```

```
    catch(ArrayIndexOutOfBoundsException ignore){}
```

```
    catch(InterruptedException ignore){}
```

```
        BufferedReader res=new BufferedReader(new InputStreamReader(proc.getInputStream()))
```

```
    );
```

```
        BufferedReader err=new BufferedReader(new
```

```
InputStreamReader(proc.getErrorStream()) );
```

```
    }
```

```
/*
```

Questa semplice procedura inserisce il comando da eseguire in un file script (/root/vlab/executor), lo esegue, attende la fine dell'esecuzione, e ritorna il controllo alla procedura chiamante. Lo script reindirizza l'output e l'errore (*stdout* ed *stderr*) in un file, che verrà poi letto e riportato nella pagina HTML dell'esercitazione.

```
*/
```

```

public static String[] FileRead(String fname)
{
    String[] Res=null;
    FileReader fileReader = null;
    BufferedReader bufferedReader = null;
    String RES="";
    try
    {
        fileReader = new FileReader(fname);
        bufferedReader = new BufferedReader(fileReader);
        String ln = "";

        while ( (ln=bufferedReader.readLine() ) != null){
            RES +=ln+"ç";}
        }
        catch(FileNotFoundException ignored) {}
        catch(IOException ignored) {}
    finally
    {
        try
        {
            if(bufferedReader != null){
                bufferedReader.close();}
            }
            catch(IOException ignored){}
        }
        Res=RES.split("ç");
        return(Res);
    }
}

```



```

public static void FileSave(String fname,String[] lines)
{
    FileWriter fileWriter = null;
    PrintWriter printWriter = null;
    try
    {
        fileWriter= new FileWriter(fname);
        printWriter = new PrintWriter(fileWriter);
        String ln = "";
        int i=0;
        try
        {
            while( (ln=lines[i]) != null )
            {
                ln=lines[i];
                if ( !(ln.equals("__NONE__")) && !(ln.indexOf("[")==0)){ //evita di duplicare la
                    printWriter.println(ln);                // linea inserita da
                    i++;                                       // java.lang.string[]
                }                                           //{{java.lang.string@....}
            }
            catch(ArrayIndexOutOfBoundsException ignored){};
            printWriter.println(lines);
            return;
        }
        catch(IOException ignore){}
    } finally{
        if(printWriter != null){
            printWriter.close(); }
        }
    }
}

```

/* Le due precedenti funzioni si occupano di leggere e salvare un file di testo. *FileRead*(*java.lang.string*) legge il file dal nome passato come parametro e ritorna le linee del file in un array di stringhe (*java.lang.string*[]). *FileSave*(*java.lang.string*,*java.lang.string*[]) accetta come parametri il nome del file da salvare e le linee da inserire. Con un semplice ciclo inserisce i valori nel file e lo salva. */

```
public static String getCookieValue(Cookie[] cookies,  
                                   String cookieName,  
                                   String defaultValue) {  
    for(int i=0; i<cookies.length; i++) {  
        Cookie cookie = cookies[i];  
        if (cookieName.equals(cookie.getName()))  
            return(cookie.getValue());  
    }  
    return(defaultValue);  
}
```

/*Questa funzione ritorna il valore del cookie cercato. Se il cookie non esiste allora ritorna il valore di default scelto da noi al momento di chiamarla */.

questionari.java

```
if( (action.equals("NuovaDomanda")) &&(Comando!=null) ) {
    if (CmdParams.length(>1){
        CmdParams=" "+CmdParams;}
    else{
        CmdParams="";}
    String StrAdd=
        Comando+CmdParams+"§"+Domanda+"§"+Help1+"§"+Help2+
            "§"+PuntiDomanda+"§"+PuntiHelp1+"§"+PuntiHelp2;
    String[] tmp=null;
    Domande[Domande.length-1]=StrAdd;
    fileio.FileSave("/root/vlab/Docente/usr/"+Docente+"/tmp_"+IF,Domande);
    tmp=StrAdd.split("§");
    String[] DomandeF=fileio.FileRead("/root/vlab/Docente/usr/"+Docente+"/tmp_"+IF);
    DomandePresenti="";
    i=0;
    try{
        while( (ln=DomandeF[i]) != null ){
            LN=null;
            ln="";
            ln=DomandeF[i];
            LN=ln.split("§");
            DomandePresenti += "<p><a href=\"vlab.cancella?fname=/root/vlab/Docente/usr/"+
                Docente+"/tmp_"+IF+"&ln="+Integer.toString(i)+
                "&loc=vlab.questionari?Docente="+Docente+"&IF="+
                IF+"\">"+LN[1]+" (" +LN[4]+" Punti) - Risposta: "+
                LN[0]+" Aiuto1: "+LN[2]+" Aiuto2: "+LN[3]+"</a></p>";
            i++;
        }
    }
    catch(ArrayIndexOutOfBoundsException ignored){}
}
```

```

if ( action.equals("SalvaQuestionario") ) &&( Intestazione!=null ) {
    String StrAdd="Layout"+"§"+Intestazione+"§"+ColoreS+"§"+
        ColoreT+"§"+PdP+"§"+IF+"§"+CK;
    String url="vlab.GeneratoreQuestionari?NomeQuestionario="+
        Intestazione+"&Docente="+Docente;
    Domande[Domande.length-1]=StrAdd;
    fileio.FileSave("/root/vlab/Docente/usr/"+Docente+"/ex/"+Intestazione,Domande);
    fileio.FileDelete("/root/vlab/Docente/usr/"+Docente+"/tmp_"+IF);
    Domande=null;
    DomandePresenti="";
    res.setStatus(res.SC_MOVED_PERMANENTLY);
    res.setHeader("Location",url);
}

```

/* I due precedenti controlli rispettivamente aggiungono una domanda e salvano l'esercitazione. Durante la costruzione dell'esercitazione, viene allocato un file temporaneo (*tmp_NomeInterfaccia*) in modo da consentire di poter ultimare la costruzione dell'esercitazione in un secondo momento. */

Bibliografia

- [1] Carla Bertacchini
“Lo schedario nella didattica laboratoriale”
Casa Editrice: “Centro programmazione editoriale”
- [2] Gabriella Belladelli e Rita Ferrarini,
“Rapporto tra gruppo classe e laboratorio: Il ruolo dei laboratori”
(<http://www.tanoni.net/Slides/Riforma%20Moratti/laboratori.pdf>)
- [3] Antonio Calvani e Mario Rotta ed. Erickson,
“Fare formazione in Internet. Manuale di didattica online”;
- [4] Stefano Penge,
“La rete per l'apprendimento e la didattica on line”
(<http://www.infoaccessibile.com/penge.htm>)
- [5] NUS Internet Remote Experimentation
Vlab
(<http://vlab.ee.nus.edu.sg/vlab>)
- [6] Corso di Reti Di Calcolatori
(<http://media.pearsoncmg.com>)
- [7] Paul Falstad's,
Math and Physics Applets
(<http://www.falstad.com/mathphysics.html>)
- [8] Alessandra Lumini, Raffaele Cappelli e Annalisa Franco,
“Laboratorio on line per le verifiche di Query e Quiz”
(<http://bias.csr.unibo.it/ExerciseLab/EntraSessione.asp>)

- [9] LabSubmittor,
(<http://www.ugrad.math.ubc.ca/coursedoc/math100/labs/>)
- [10] Scuola Virtuale,
(<http://www.scuolavirtuale.it>)
- [11] National Instruments
Labview
(<http://www.natinst.com/labview/>)
- [12] prof. Enrico Purchi,
Tutorial su Labview
(utenti.romascuola.net/aristotele-2e/labview/)
- [13] Autore sconosciuto,
Labview. Esempio di utilizzo.
(<http://zer0un0.altervista.org/labview>)
- [14] GONETWORK,
Organizzazione Aziendale e Soluzioni Internet
(<http://www.gonetwork.it>)
- [15] Roberto Tecchio,
Questionari On line
(<http://www.tecchio.net/clienti/questionari/>)
- [16] Smartlite Software,
WinAsk2000 & WebQuiz 2002
(<http://www.smartlite.it>)
- [17] Autori Vari,
Notizie generali su Java e Servlets
(<http://www.latoserver.it>)