



# TechnoLum250Festival

Progetto Tecnologie Web 2023/2024

*Componenti:*

Davide Malgarise - 2009994

Giacomo D'Ovidio - 2009101

Marco Piccoli - 2045039

Riccardo Alberto Costantin - 2013451

*Indirizzo sito web:*

<http://tecweb.studenti.math.unipd.it/gdovidio>

*Credenziali utente:*

username: user

password: user

*Credenziali admin:*

username: admin

password: admin

*Email Referente:*

[giacomo.dovidio@studenti.unipd.it](mailto:giacomo.dovidio@studenti.unipd.it)



# Indice

<b>1 Introduzione .....</b>	<b>4</b>
<b>2 Analisi dei requisiti .....</b>	<b>5</b>
2.1 Target di utenza .....	5
2.2 Tipi di utenti e funzionalità .....	5
2.2.1 Visitatori standard: .....	5
2.2.2 Utenti registrati: .....	5
2.2.3 Admin: .....	5
<b>3 Progettazione .....</b>	<b>6</b>
3.1 Wireframe e mockup .....	6
3.2 Convenzioni adottate .....	6
3.2.1 Link .....	6
3.2.2 Logo cliccabile .....	6
3.2.3 Breadcrumb .....	6
3.3 Pagine .....	6
3.3.1 Area comune .....	6
3.3.2 Area riservata utente .....	7
3.3.3 Area Amministrativa .....	7
3.4 Struttura gerarchica .....	7
3.5 Schema organizzativo .....	7
3.5.1 Lingua .....	7
<b>4 Realizzazione .....</b>	<b>8</b>
4.1 HTML .....	8
4.2 Struttura principale .....	8
4.3 Pagine di errore .....	8
4.4 CSS .....	8
4.4.1 Layout di stampa .....	9
4.5 JavaScript .....	9
4.6 PHP .....	9
4.6.1 Template .....	9
4.6.2 Link circolari .....	9
4.6.3 Connessione al Database .....	9
4.6.4 Form e validazione degli input .....	10
4.6.5 Variabili di sessione .....	10
4.6.6 Sicurezza .....	10
4.7 Database .....	10
<b>5 SEO .....</b>	<b>11</b>
5.1 Possibili ricerche .....	11
<b>6 Test .....</b>	<b>12</b>
6.1 Validazione sito .....	12
6.2 Browser web .....	12
6.3 Test umani .....	12

<b>7 Accessibilità .....</b>	<b>13</b>
7.1 Tabindex .....	13
7.2 Aiuti alla navigazione .....	13
7.3 Colori .....	13
7.4 Tabella .....	13
<b>8 Installazione Progetto .....</b>	<b>14</b>
<b>9 Organizzazione .....</b>	<b>15</b>
9.1 Organizzazione interna .....	15
9.2 Suddivisione lavoro .....	15
<b>10 Conclusioni .....</b>	<b>16</b>

# **1 Introduzione**

Per il progetto didattico del corso di Tecnologie Web, nell'Anno Accademico 2023-2024, il nostro gruppo si è dedicato alla realizzazione di un sito per un ipotetico Festival musicale chiamato TechnoLum250Festival.

## **2 Analisi dei requisiti**

### **2.1 Target di utenza**

Il suddetto festival si rivolge principalmente a un pubblico giovane, la cui età spazia tra i 16 e i 35 anni, attirati da generi musicali quali hip-hop, R&B e musica elettronica. Il sito sviluppato deve di conseguenza rivolgersi principalmente a queste categorie di utenti. La demografia rilevata apprezza un'interfaccia utente intuitiva, un design accattivante e facilmente navigabile da dispositivi mobili, riflettendo il loro stile di vita dinamico. Sebbene il sito web sia da sviluppare in modo tale da strizzare l'occhio ad un pubblico giovane e costantemente al passo con la tecnologia, è fondamentale che il design non escluda fasce d'età più mature. Di conseguenza è necessario che il sito offra un'esperienza utente accessibile a un utente più anziano, che potrebbe essere interessato a esplorare e partecipare all'evento. Più in generale, il contesto in oggetto si propone di essere un punto di incontro per l'intera comunità. Di conseguenza, in linea con questo presupposto di inclusione, è fondamentale che il sito web sia pienamente accessibile alle persone con disabilità per garantire loro una navigazione piacevole e chiara.

### **2.2 Tipi di utenti e funzionalità**

Il gruppo già in fase di analisi ha stabilito che non si sarebbero implementate una serie di funzionalità che, in ottica di un reale utilizzo, sarebbero ideali. Per esempio la possibilità di avere delle prevendite nominative, la possibilità di refund delle stesse oppure la possibilità per l'amministratore di visualizzare l'andamento delle vendite. Questo è dovuto principalmente alle limitazioni di tempo e risorse tipiche di un progetto accademico. Ci si è infatti posti l'obiettivo di realizzare funzionalità diversificate, in modo tale da poter esplorare a pieno le possibilità offerte dai linguaggi utilizzati. Successivamente sono presentate le categorie di utenti e le funzionalità che il team ha individuato in sede di analisi.

#### **2.2.1 Visitatori standard:**

Questa categoria deve poter aver accesso a tutte le informazioni generali riguardanti il festival: in particolare deve poter esplorare la lineup degli artisti, visualizzare le FAQ e avere una preview del merch disponibile in Loco. Inoltre deve poter effettuare la registrazione.

#### **2.2.2 Utenti registrati:**

Tali utenti, oltre a tutte le possibilità offerte al visitatore Standard, devono poter aggiornare i propri dati personali, poter cambiare la password, acquistare prevendite e visualizzare i dettagli degli acquisti sotto forma tabellare. Infine hanno la possibilità di eliminare definitivamente l'account dal sito web.

#### **2.2.3 Admin:**

L'admin ha inoltre la possibilità di modificare la lineup del festival inserendo ed eliminando gli eventi. Tali eventi sono caratterizzati univocamente dal performer, da una sua foto, da una data e un orario e infine, da una descrizione.

## 3 Progettazione

### 3.1 Wireframe e mockup

Prima di avviare lo sviluppo, abbiamo adottato una metodologia di progettazione che includeva la creazione di wireframe e mockup. L'attività è stata fondamentale: abbiamo infatti potuto circoscrivere meglio i requisiti e ragionare subito sulle possibili implementazioni. Inoltre ci ha permesso di definire un'idea comune di quello che saremmo andati a realizzare. In questa sede abbiamo anche avuto modo di approfondire il tema in oggetto, analizzando le caratteristiche presenti in siti simili. I disegni realizzati si sono concentrati sulla logica della disposizione degli elementi, sulla navigazione e sulla gerarchia degli elementi, lasciando da parte dettagli grafici. Negli incontri successivi, a seguito dell'individuazione di una prima palette di colori, abbiamo specializzato il dettaglio di quei disegni, senza però entrare troppo nello specifico. Questo processo preparatorio è risultato molto utile per scolpire un primo aspetto, prima di entrare nella fase di sviluppo.

### 3.2 Convenzioni adottate

Si riportano di seguito le convenzioni adottate dal gruppo:

#### 3.2.1 Link

Il team ha deciso di mantenere i link sottolineati come da Standard WCAG, inoltre per ridurre il disorientamento cognitivo si è optato per mantenere un colore diverso per i link visitati. I colori individuati sono stati il bianco e il giallo rispettivamente per i link visitati e non visitati.

#### 3.2.2 Logo cliccabile

Il team ha scelto di adottare la pratica ormai quasi universale nel web design di associare al logo un link cliccabile che riporti alla pagina home. Tramite espressioni regolari si è garantita l'eliminazione dei link circolari.

#### 3.2.3 Breadcrumb

Abbiamo abbracciato la convenzione di adottare una breadcrumb per favorire la navigazione e limitare il sovraccarico cognitivo per l'utente.

### 3.3 Pagine

Si riporta una breve descrizione delle pagine implementate e disponibili alle varie categorie di utenti

#### 3.3.1 Area comune

- **Home:** pagina principale del sito e la prima visualizzata quando si accede al sito. Incorpora un carosello scorrevole in modo dinamico in cui vengono mostrate varie foto del Festival. Per ogni giornata del festival vengono mostrati gli artisti che si esibiscono con l'orario rispettivo.
- **Chi siamo:** pagina che descrive brevemente il Festival e contiene i vari ringraziamenti.
- **Location:** pagina in cui sono presenti le informazioni per raggiungere il Festival.

- **Merch:** pagina in cui vengono visualizzati e descritti gli item del merch che possono essere acquistati in Loco.
- **Prevendita:** pagina in cui vengono mostrati i biglietti che si possono acquistare online. Inoltre vengono descritti i vantaggi dell'acquisto di un biglietto categoria VIP, l'acquisto rimane però bloccato ad utenti non autenticati.
- **Domande:** pagina in cui vengono mostrate le domande più frequenti e le relative risposte.
- **Account:** pagina per poter effettuare l'accesso.
- **Privacy Policy:** pagina che contiene le informative riguardo la privacy.

### 3.3.2 Area riservata utente

- **Registrazione:** pagina che permette all'utente generico di potersi registrare. L'attività di registrazione richiede nome, cognome, età, indirizzo di residenza, email, username e password.
- **Account:** una volta effettuato l'accesso si potranno visualizzare i propri dati personali inseriti precedentemente, nella fase di registrazione. Una sezione permette all'utente di modificare indirizzo di residenza, email o password, un'altra permette la visualizzazione delle prevendite acquistate sotto forma tabellare (se presenti) e infine un'ultima sezione permette l'accesso alla pagina di eliminazione dell'account.
- **Eliminazione:** pagina che consente l'eliminazione dell'account dal sito web. È richiesta la password.

### 3.3.3 Area Amministrativa

- **Account:** sempre in questa pagina l'admin può visualizzare gli eventi e gli artisti che si esibiranno. Ha la possibilità di effettuare l'inserimento di eventi o la loro rimozione.

## 3.4 Struttura gerarchica

La gerarchia è stata sviluppata principalmente in ampiezza. Il menù principale ha come sezioni: *Home*, *Chi Siamo*, *Location*, *Merch*, *Prevendite*, *Domande* e *Account* per un totale di 7 voci. La profondità massima è di 3 elementi che si verifica nella pagina *Account*.

## 3.5 Schema organizzativo

È stato adottato uno schema esatto per i contenuti ospitati nel sito, garantendo che ogni sezione sia mutualmente esclusiva, con contenuti distinti e senza sovrapposizioni. La categorizzazione degli eventi è stata implementata seguendo un ordine cronologico.

### 3.5.1 Lingua

Il sito web rispetta la lingua e cultura italiana, eventuali parole inglesi sono state marcate con il tag *span* e attributo *lang='en'*.

## 4 Realizzazione

In questa sezione vengono mostrate le decisioni intraprese nel corso dello sviluppo.

### 4.1 HTML

Il sito, come da specifiche di progetto, è stato sviluppato seguendo la sintassi di HTML5. Il gruppo si è impegnato nell'utilizzare i tag semantici corretti già dai primi momenti dello sviluppo, inoltre il processo di scrittura di codice HTML è stato sempre accompagnato da relativa validazione.

Nel corso dello sviluppo abbiamo cercato di mantenere un rapporto di massima separazione tra il contenuto HTML e le componenti di PHP, competenti della loro unione in quello che sarà il codice HTML finale disponibile all'utente. Nella nostra repository è infatti presente una cartella *html* in cui sono contenute tutte le componenti HTML necessarie alla visualizzazione del sito. Le pagine *struttura.html*, *header.html*, *footer.html*, *menu.html* contengono la struttura portante di tutte le pagine presenti sul sito, sono state usate come template e poi dinamicamente modificate a formare il risultato finale.

### 4.2 Struttura principale

La struttura di ogni pagina si caratterizza di un header, un main e un footer. Nell'header possiamo trovare il logo, il nome del festival, le icone dei social e il menù principale. Nel Main il contenuto della pagina. Nel footer le icone dei social, i diritti di copyright e l'informativa riguardante la privacy.

### 4.3 Pagine di errore

Sono state personalizzate le pagine di errore così da non disorientare l'utente e contribuire a mantenere un clima di fiducia. Questo accade ad esempio nella pagina di errore 404.

### 4.4 CSS

Il design è stato sviluppato inizialmente per il sito nella sua versione Desktop, successivamente è stato rielaborato per l'accesso tramite schermi di dimensioni minori. Le principali differenze tra le due visualizzazioni si trovano nel menù di navigazione e nella tabella relativa agli ordini degli utenti. Il primo viene trasformato in un menù ad hamburger accessibile anche agli screen reader. La tabella viene invece resa verticale, eliminando l'header dalla visualizzazione (anche per gli screen reader), e sostituendo le funzionalità tramite l'utilizzo del costrutto *td:before* per iniettare nella singola cella il contenuto dell'attributo *data-title*. Il menù ad hamburger e il suo funzionamento sono stati garantiti solo attraverso l'utilizzo di CSS, una funzione JavaScript permette invece di poterlo cliccare una volta raggiunto tramite navigazione con il tasto *tab*. Per garantire una maggiore accessibilità è stata implementata una classe CSS chiamata *accessibleHide*: questa classe ci permette di eliminare gli elementi dalla vista mantenendoli però rilevabili dagli screen reader. Il layout finale è responsive: si utilizzano punti di rottura e all'interno di essi si garantisce fluidità.



#### 4.4.1 Layout di stampa

Come da specifiche di progetto viene fornito un file .css, dedicato al layout di stampa del sito. Sono stati rimossi il menù, le immagini del carosello, i pulsanti di tipo submit dai vari form. Nella stampa le pagine sono solamente in bianco e nero, l'unica eccezione è il logo. Per garantire che un elemento non venisse spezzato su varie pagine si è utilizzata la proprietà *page-break*. Su alcune pagine è stata cambiata totalmente la disposizione dei vari elementi per garantire una lettura più piacevole.

### 4.5 JavaScript

Il linguaggio JavaScript è stato utilizzato per lo sviluppo del carosello dinamico, per il countdown al giorno di inizio del festival e per mostrare in modo dinamico il prezzo totale di acquisto dei biglietti. Inoltre è stato essenziale nel processo di validazione degli input inseriti da parte dell'utente, ogni form infatti presenta controlli lato client e produce degli errori che tramite la funzione *appendError* vengono mostrati a schermo all'interno di un determinato *div*. Abbiamo provveduto, per quanto fattibile, a mantenere gli stessi controlli lato client e lato server. Per i controlli che interagiscono con il database o con le variabili di sessione i controlli rimangono solamente lato server. Inoltre tutto il sito è stato sviluppato considerando il fatto che sarebbe dovuto rimanere pienamente accessibile e utilizzabile anche nel momento in cui JavaScript non fosse disponibile o disattivato. Tutto il codice è stato incorporato all'interno di un unico file in modo tale da limitare le richieste HTTP e per una più agevole manutenzione. Per garantire che tutti gli script aspettassero l'effettivo caricamento del DOM prima di operare è stato fatto ampio uso di event listener legati all'evento *DOMContentLoaded*.

### 4.6 PHP

PHP è stato ampiamente utilizzato. Si riportano successivamente le principali funzioni svolte.

#### 4.6.1 Template

Per evitare duplicazione di codice e favorire il riuso di quest'ultimo, il PHP si occupa della costruzione dinamica delle pagine, importando i vari file template HTML e, attraverso ancoraggi e funzioni di string replace, iniettando il contenuto. Gli ancoraggi vengono definiti nei file HTML con le doppie parentesi graffe.

#### 4.6.2 Link circolari

Per rimuovere i link circolari, ovvero link che portano alla stessa pagina, è stata sviluppata una funzione in PHP che tramite espressioni regolari permette di rimuovere direttamente il tag `<a>` se ci troviamo in quella specifica pagina.

#### 4.6.3 Connessione al Database

La classe *DBAccess* effettua il collegamento al database e costituisce l'oggetto effettivo della connessione, mentre le query vengono effettuate tramite la classe *DBOperation*: per interfacciarsi al database è stata utilizzata la libreria *mysqli*.

#### 4.6.4 Form e validazione degli input

Ogni form è stato configurato per essere gestito da un file PHP dedicato, utilizzando il metodo POST per la trasmissione sicura dei dati. Sono stati garantiti gli stessi controlli presenti nella validazione lato client, inoltre i messaggi di errore vengono ristampati nel medesimo contenitore utilizzato da JavaScript. Per i form di dimensione maggiore è stata inoltre implementata la funzionalità di ricostruzione dell'input.

#### 4.6.5 Variabili di sessione

La gestione delle sessioni utente è stata interamente delegata al linguaggio PHP tramite variabili di sessione. Questo approccio produce un cookie di sessione esistente solo ed esclusivamente nel browser dell'utente, motivo per cui nel nostro sito non è presente un form per acconsentire all'uso dei cookie. Per la pagina *Account*, è risultato molto utile salvare l'username su una variabile di sessione, per gestire in modo efficace l'accesso e le interazioni dell'utente.

#### 4.6.6 Sicurezza

- Nel database le password non sono salvate in chiaro, bensì utilizzando l'algoritmo di hashing di default di PHP usando la funzione *PASSWORD\_DEFAULT*;
- Per prevenire attacchi di tipo *XSS Cross-Site-Scripting* e *Javascript Injection* sono state sviluppate delle funzioni che tramite espressioni rilevano l'immissione di tag HTML all'interno degli input form;
- Sono state realizzate query parametriche tramite la funzione *prepare()* di *mysqli* per prevenire attacchi di tipo SQL Injection.

### 4.7 Database

Come database si è deciso di usare *MariaDB*, classica base di dati di tipo relazionale. Disponiamo di 5 tabelle:

- Tabella *users* in cui vengono elencati tutti gli utenti registrati al sito, con relative informazioni anagrafiche;
- Tabella *Programma* in cui si memorizzano gli artisti e l'orario in cui si esibiranno, insieme all'indirizzo relativo;
- Tabella *Biglietti* in cui si salvano le varie tipologie di Biglietti che è possibile acquistare;
- Tabella *Ordini* per registrare tutti gli ordini effettuati dagli utenti;
- Tabella *Shop* per salvare gli articoli, che è possibile acquistare al Festival.

La tabella *users* in realtà non è totalmente ottimizzata, contiene infatti un campo booleano che definisce la tipologia di utente, si è comunque deciso di mantenere una tabella unica e non dividerla appunto per mantenere delle query più semplici. Nelle tabelle *Shop* e *Programma*, per gestire le immagini, si è salvato il path della locazione delle foto. Tutte le foto hanno una dimensione inferiore al MB, e sono stati sviluppati i relativi controlli, lato PHP e JavaScript, per evitare il caricamento di immagini più pesanti o di formati non accettati.

## 5 SEO

Vengono elencate le considerazioni che il gruppo ha adottato per favorire un buon indice di posizionamento all'interno dei motori di ricerca:

1. Codice HTML5 e CSS sono stati validati;
2. L'adozione di parole chiave comuni a tutte le pagine e alcune varianti a seconda della pagina, all'interno del meta tag *keyword*;
3. In ogni pagina è presente un *title* che va dal particolare al generale, in modo da fornire un contesto specifico;
4. È stato usato un unico file JavaScript;
5. È presente un design responsive;
6. È stato creato un file *robots.txt*, per evitare l'indicizzazione di alcune pagine dai motori di ricerca, considerate non essenziali, in modo tale che le risorse dei crawler siano orientate verso le pagine più ricche di contenuto e non verso pagine sensibili o di amministrazione.

### 5.1 Possibili ricerche

Vengono di seguito elencate, in ordine di importanza, le ricerche a cui il sito si propone di rispondere:

1. Nome del Festival (TechnoLum250);
2. Date dello svolgimento del Festival;
3. Parole generiche quali Festival, Padova, Evento etc.;
4. Nomi degli artisti presenti al Festival.

Essendo comunque tutti campi di ricerca generali, escludendo chiaramente il titolo stesso del festival, sono dunque ormai affollati da altri siti. Sarà dunque necessario, in fase di rilascio del sito al pubblico, andare ad effettuare una qualche campagna di pubblicizzazione tramite piattaforme social in maniera tale da andare ad aumentare il quantitativo di visite in entrata e dunque a migliorare il suo posizionamento.

## 6 Test

### 6.1 Validazione sito

Per validare il codice HTML5 del sito è stato usato come strumento <https://validator.w3.org/> e Total Validator Basic presente nei computer del Paolotti. Per validare il codice CSS è invece stato usato <https://jigsaw.w3.org/css-validator/>.

### 6.2 Browser web

Sono stati testati i seguenti browser:

- Google Chrome (Window 10 e 11 e Linux);
- Mozilla Firefox (Window 10 e 11 e Linux);
- Microsoft Edge (Windows 10 e 11 e Linux);
- Opera (Windows 11);
- Brave (Linux);

Per quanto riguarda il testing su Safari, dato che nessuno dei componenti possiede alcun dispositivo Apple, e che l'ultima versione rilasciata per dispositivi Window non supporta *flexbox*, non è stato possibile. Per testare il sito su dispositivi mobili, si sono usati gli strumenti per sviluppatori sia di Google Chrome che di Firefox.

### 6.3 Test umani

Per testare navigabilità e usabilità del sito è stato fatto provare ad utenti reali delle cerchie di conoscenza dei componenti del gruppo, in generale si sono sempre ottenuti responsi positivi.

## 7 Accessibilità

Per rendere il sito accessibile si sono usati i seguenti strumenti:

- Total Validator presente nei pc di laboratori del Paolotti;
- Wave, un'estensione per Google Chrome;
- L'analisi dell'accessibilità offerta da Mozilla Firefox;
- Lo screen reader NVDA su Windows e per ambienti Linux il sistema *orca*.

### 7.1 Tabindex

Non è stato alterato l'ordine naturale dei *tabindex*.

### 7.2 Aiuti alla navigazione

È presente un pulsante per saltare direttamente al contenuto.

### 7.3 Colori

Si è prestata molta attenzione ai colori e ai contrasti, i contrasti sono stati per lo più rilevati tramite strumenti automatici. Per quanto riguarda gli utenti soggetti ad alterazione del senso cromatico, il gruppo ha effettuato varie simulazioni a garantire che il contenuto potesse rimanere ugualmente fruibile. Inoltre si precisa che i colori non sono mai stati usati come unica modalità di trasmissione dell'informazione, per esempio tutti i link risultano essere sottolineati.

### 7.4 Tabella

La tabella degli ordini nella Pagina *Account* è stata resa accessibile, adottando i criteri standard.

## 8 Installazione Progetto

Questa guida consente, a partire dalla consegna effettuata tramite moodle, di reinstallare il progetto nel server dedicato. Se il progetto dev'essere installato sul server gdovidio non è necessario aggiornare il contenuto dei file. Si elencano i passaggi da eseguire.

1. Nel file *src/phputilities/DBAccess.php* si devono cambiare le credenziali di accesso al database;
2. Accedere a *phpmyadmin* e eliminare il contenuto del proprio database;
3. Sostituire il nome del database usato nel file *init-scripts/init.sql*;
4. Importare il database tramite *phpmyadmin*;
5. inserire tutto il contenuto della cartella *src* (non la cartella stessa) all'interno della cartella *public\_html* del server;
6. la gestione degli errori 404 e 500 è già stata settata per funzionare anche tramite tunnel ssh e non ha bisogno di ulteriori modifiche.

## 9 Organizzazione

### 9.1 Organizzazione interna

Per lavorare in maniera efficiente al progetto, si è deciso di adottare Docker: è stato realizzato un file *docker compose* che una volta avviato mettesse a disposizione del team tutte le componenti necessarie allo sviluppo nelle medesime versioni. Inoltre durante lo sviluppo si sono fatte continue prove nel server dedicato. Per la condivisione e il versionamento del codice sorgente si è utilizzato GitHub.

### 9.2 Suddivisione lavoro

- Davide Malgarise:
  - CSS;
  - Layout di stampa;
  - HTML pagine statiche;
  - Popolamento database;
  - Verifica relazione;
  - Attività di revisione del codice.
- Giacomo D'Ovidio:
  - Template HTML;
  - PHP;
  - Database: accesso e operazioni;
  - JavaScript: controlli front-end;
  - Garante dell'accessibilità;
  - Validazione input;
  - Relazione.
- Marco Piccoli:
  - HTML pagine statiche;
  - CSS;
  - Carosello dinamico e animazioni CSS e JavaScript;
  - Verifica relazione;
  - Attività di revisione del codice.
- Riccardo Alberto Costantin:
  - Amministratore della repository;
  - Template HTML;
  - PHP;
  - Database: accesso e operazioni;
  - JavaScript: controlli front-end;
  - Validazione input;
  - Relazione.

## 10 Conclusioni

Il gruppo conviene nel dire che il progetto è stata un'esperienza estremamente formativa e interessante. Si è infatti avuto modo di sviluppare e consolidare competenze tecniche molto importanti anche in vista di ipotetici sbocchi professionali. Il gruppo essendo per lo più composto da componenti provenienti da Licei scientifici tradizionali non aveva mai avuto modo di interfacciarsi direttamente a queste tecnologie. Inoltre si sono dovute affrontare tutta una serie di difficoltà legate allo sviluppo non individuale (problemi di comunicazione, di gestione del codice sorgente condiviso, di pianificazione, etc). In generale possiamo dire di aver imparato molto e sicuramente nel complesso l'esperienza è stata formativa.