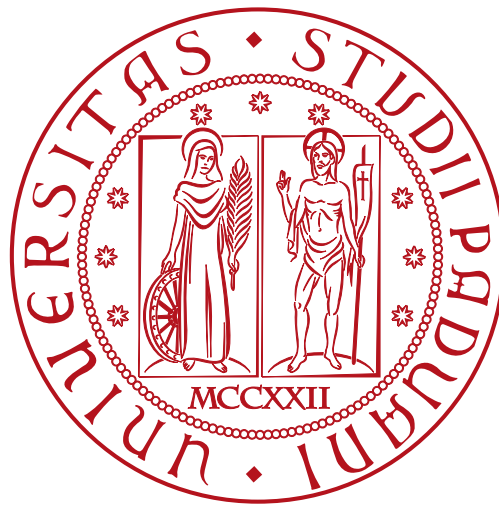


Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

CORSO DI LAUREA IN INFORMATICA



WebApp per attività laboratoriali di OpenDay

Tesi di laurea

25 Luglio 2025

Relatrice

Prof.ssa Ombretta Gaggi

Laureando

Orlando Virgilio Maria Ferazzani

Matricola 2058653

Citazione qui

—

Sommario

Il seguente elaborato descrive l'attività di tirocinio, della durata complessiva di xxx ore, svolta presso l'Università di Padova. Questa attività è stata portata avanti sotto la guida della Prof.ssa Ombretta Gaggi. Il Prof. Claudio Palazzi ha ricoperto il ruolo di tutor accademico.

L'Università degli Studi di Padova durante i suoi OpenDay, utilizza WebApp _G interattive per avvicinare i ragazzi delle scuole superiori al corso di laurea in Informatica. Queste consentono di far conoscere le basi della programmazione attraverso giochi che stimolano la logica e la creatività. Tuttavia, queste applicazioni sono spesso molto tediose da utilizzare dato il breve tempo a disposizione per le suddette attività e non sempre riescono a coinvolgere gli studenti, soprattutto chi di programmazione non ha mai intrapreso degli studi di alcun genere.

Il tirocinio effettuato mira proprio a risolvere questa problematica, sviluppando una WebApp interattiva che permetta di avvicinare i ragazzi al mondo della programmazione in modo divertente e stimolante.

Ringraziamenti

Padova, Luglio 2025

Orlando Virgilio Maria Ferazzani

Indice

| | |
|-------------------------------------|-----------|
| Introduzione | 1 |
| 1.1 Motivazioni e Contesto | 1 |
| 1.2 Strumenti e processi | 2 |
| 1.2.1 Suddivisione del lavoro | 2 |
| 1.3 Struttura del Documento | 4 |
| | |
| Scopo del tirocinio | 6 |
| 2.1 Scopo del progetto | 6 |
| 2.2 Obiettivi prefissati | 7 |
| 2.3 Tecnologie utilizzate | 10 |
| 2.3.1 NextJS | 10 |
| 2.3.2 TailwindCSS | 11 |
| 2.4 Prodotti attesi | 11 |
| | |
| Implementazione | 12 |
| Retrospettiva finale | 13 |
| Glossario | 15 |
| Bibliografia | 17 |

Elenco delle Figure

Figura 1.1 vista della Kanban alla seconda settimana 3

Figura 1.2 vista diagramma di Gantt alla seconda settimana 4

Elenco delle Tabelle

Capitolo 1

Introduzione

1.1 Motivazioni e Contesto

Il corso di laurea triennale in *Informatica* che offre l'Ateneo dell'*Università Di Padova* è a numero chiuso, con un massimo di 220 posti, cifra che nel corso degli anni è aumentata visto il grande interesse da parte di studenti delle superiori di intraprendere questo percorso. Tuttavia, le scuole di provenienza delle matricole mostrano una grande disparità, con il numero degli studenti provenienti da istituti ad indirizzo informatico (come istituti tecnici o liceo delle scienze applicate) che supera di gran lunga il numero di studenti provenienti da licei tradizionali. In particolare, parliamo di circa il 75% di studenti che proviene da istituti tecnici, con il restante 25% proveniente da altre scuole.

Questa disparità è data sicuramente dalla «credenza» che l'informatica sia solo programmazione e che quindi uno studente proveniente da un Liceo Scientifico non abbia le competenze necessarie per questo affrontare questo percorso.

Tuttavia, di contro a questa «credenza», è ovvio che non mancano le competenze, ma solamente le conoscenze, che è proprio il vuoto che questo corso va a colmare. Se magari uno studente proveniente da un istituto tecnico ha già delle competenze di programmazione, uno studente proveniente da un liceo scientifico ha sicuramente delle com-

petenze matematiche e logiche che sono fondamentali per affrontare questo percorso.

Altro obiettivo molto importante, è quello di aumentare il numero di iscritti di genere femminile, dimostrando, grazie anche a vari testimonial ([WomenInCS](#)) alle varie studentesse, che la storia dell'informatica è stata fatta in non piccola parte da donne e scienziate.

L'informatica non è solo programmazione, ma è anche progettazione e design. Infatti, il mio progetto si basa sulla creazione di un'applicazione web interattiva, che richiede competenze di progettazione e design oltre a quelle di programmazione.

Personalmente, essendo uno di quei ragazzi timorosi di iniziare il percorso, arrivando da un Liceo Scientifico, mi sono sentito moralmente in dovere di aiutare tutti coloro che si trovano nella posizione in cui io stesso mi sono trovato, anche per questo quindi ho deciso di fare questo tirocinio con la Prof.ssa Ombretta Gaggi.

1.2 Strumenti e processi

Durante il corso del tirocinio, mi sono avvalso di diversi strumenti che ho imparato ad utilizzare nel corso della mia carriera universitaria, e che mi hanno aiutato a portare avanti il mio progetto, come [Git](#) e [GitHub](#), utilissimi per tenere traccia di ogni modifica effettuata al codice sorgente dell'applicazione, nonché per la condivisione di tale codice con la mia relatrice. In ogni caso, tutte le tecnologie saranno discusse nel dettaglio [secondo capitolo](#).

1.2.1 Suddivisione del lavoro

Dovendo presentare un [Piano di Lavoro](#) per iniziare il mio tirocino, e volendo rispettare gli insegnamenti appresi dal corso di Ingegneria del Software, ho da subito deciso di impostare il mio Way of Working.

Ho quindi dapprima definito tutti gli obiettivi da raggiungere durante il percorso, trovandone 19. A questo punto, ho suddiviso il lavoro da svolgere nelle 8 settimane, potendo quindi definire degli « [sprint G](#) ». A fine di ogni sprint, controllo di aver completato tutto ciò che mi ero prefissato nel [backlog G](#) di lavoro e, se ci fosse qualcosa che non ho completato, lo sposto nel backlog del prossimo sprint.

Per il tracciamento delle task da completare, ho utilizzato una [Kanban G board](#)^[1.1], divisa in 3 colonne:

1. **ToDo:** attività da completare
2. **In Progress:** attività in corso
3. **Done:** attività completate

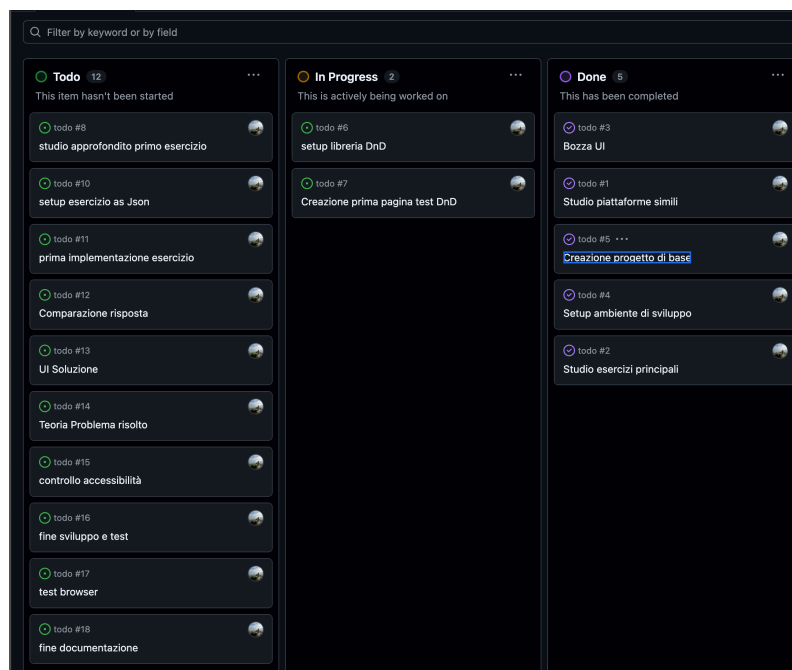


Figura 1.1: vista della Kanban alla seconda settimana

Invece, per visualizzare i tempi di svolgimento previsti ed effettivi di tali task, ho deciso di utilizzare un diagramma di Gantt ^[1.2].

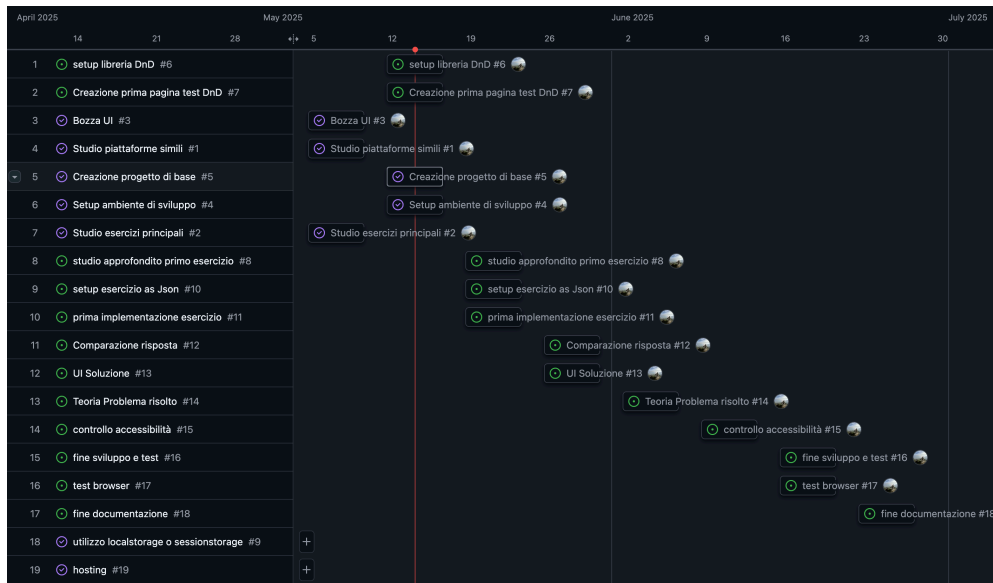


Figura 1.2: vista diagramma di Gantt alla seconda settimana

Tutto questo è conforme al metodo di lavoro **Scrum**, che prevede una suddivisione del lavoro in sprint e un monitoraggio costante dei progressi. Lo Scrum fa parte della metodologia Agile, creata per migliorare lo sviluppo di prodotti software rallentati dalle tediose fasi di analisi e documentazione.

1.3 Struttura del Documento

- **Cosa:** Il **secondo capitolo** fornisce una panoramica di tutto il progetto, dal suo scopo, a i prodotti attesi, descrivendo nel dettaglio il concetto di webapp interattiva e le tecnologie utilizzate e le motivazioni dietro queste.
- **Come:** Il **terzo capitolo** espone come le scelte descritte nel capitolo precedente sono state implementate. Inoltre, viene fornita una panoramica del codice sorgente, con i file più significativi e le loro funzionalità, oltre che la descrizione delle caratteristiche di accessibilità.
- **Conclusioni:** Il **quarto capitolo** fornisce una panoramica dei risultati ottenuti, sia a livello di codice sorgente che di accessibilità. Inoltre,

viene fornita una panoramica dei test effettuati e dei risultati ottenuti.

- Nel [Glossario](#) sono riportati i termini tecnici e le abbreviazioni utilizzate nel corso del documento.

Oltre alla struttura qui sopra descritta, si adottano anche i seguenti accorgimenti tipografici:

- le abbreviazioni, termini tecnici (o comunque di uso non comune), o in lingua straniera in prima occorrenza nel documento sono definiti nel glossario consultabile alla fine del documento. Ogni termine nel glossario è evidenziato come segue: [Parola_G](#).
- Altri termini che richiedono un'attenzione particolare, ma che non hanno bisogno di essere definiti, saranno evidenziati in corsivo: *Parola*.

Capitolo 2

Scopo del tirocinio

Il prossimo capitolo fornisce una panoramica dettagliata del progetto di stage, partendo dal suo scopo, passando per gli obiettivi prefissati e le tecnologie utilizzate, arrivando ai prodotti attesi.

2.1 Scopo del progetto

Come evidenziato nel capitolo precedente, questo progetto di stage è volto a colmare una grande “disparità” tra gli studenti che si iscrivono al corso di laurea in *Informatica dell’Università di Padova*.

Possiamo comunque distinguere due “sotto-scopi” che convogliano poi in nel progetto finale:

Il primo scopo, definibile *scopo tecnico*, è rivolto al laureando, è quello di portare a limite le proprie abilità, competenze e conoscenze cercando di creare un prodotto che non solo sia funzionale, ma anche facile da utilizzare, accessibile a tutti e che aiuti effettivamente gli studenti. Questo scopo si traduce, all’atto pratico, nello sviluppo della WebApp interattiva.

Il secondo scopo, definibile *scopo sociale*, è rivolto agli studenti delle scuole superiori, e ha come obiettivo quello di mostrare loro che l’informatica non è solo programmazione, ma anche progettazione e design, e che non importa quale sia il loro *background* scolastico per affrontare questo percorso. In particolare, si vuole dimostrare che ogni scuola superiore, anche se in maniera e con contenuti diversi,

permette di affrontare questo percorso, chi in un modo e chi in un altro.

Se dovessi dare quindi una definizione sintetica, quasi matematica come il mio corso ha insegnato, dello scopo del progetto, direi che è:

Si definisce il seguente progetto di sviluppo di una WebApp interattiva per l'orientamento agli Open Day di Informatica il processo che, dato un insieme eterogeneo di studenti delle scuole superiori, realizza uno strumento digitale accessibile e inclusivo, volto a ridurre le disparità di background e di genere e a mostrare che l'informatica è disciplina aperta a tutti, indipendentemente dal percorso scolastico precedente.

2.2 Obiettivi prefissati

Nelle fasi iniziali di quella che possiamo chiamare "candidatura" al progetto, ho definito ciò che ritenevo fossero gli obiettivi principali del progetto, sapendo che questi comunque sarebbero stati mutevoli nel tempo, essendo aggiornati man mano che il progetto andava avanti sulla base delle esigenze e sull'effettivo progresso del progetto. Gli obiettivi definiti sono poi stati discussi insieme alla mia relatrice, e raggruppati in tre diverse categorie:

- **Obiettivi Obbligatori:** sono gli obiettivi che devono essere raggiunti per considerare il progetto completato e soddisfacente. Questi obiettivi sono stati definiti in modo da garantire che il prodotto finale sia funzionale, accessibile e utile per gli studenti delle scuole superiori.
- **Obiettivi Desiderabili:** sono obiettivi che, se raggiunti, migliorano significativamente il progetto e lo rendono più completo e interessante. Questi obiettivi possono essere considerati come "aggiunte"

che arricchiscono l'esperienza utente o aggiungono funzionalità utili, ma non sono strettamente necessari per il completamento del progetto.

- **Obiettivi Facoltativi:** sono obiettivi che, se raggiunti, arricchiscono il progetto e lo rendono più completo e interessante. Questi obiettivi possono essere considerati come "extra" che migliorano l'esperienza utente o aggiungono funzionalità utili.

Per la classificazione degli obiettivi è stata adottata la seguente nomenclatura:

- Obbligatorio: O
- Desiderabile: D
- Facoltativo: F

Con il codice identificativo dell'obiettivo che diventa quindi:

[X] [Y]

con X che rappresenta la categoria dell'obiettivo e Y che rappresenta il numero progressivo dell'obiettivo.

Gli obiettivi definiti inizialmente sono stati i seguenti:

Obbligatori

- O01: implementazione dell'applicativo;
- O02: funzionamento base dello stesso;
- O03: primo problema: produttore - consumatore;
- O04: simulazione della sequenza di operazione eseguita come live demo;
- O05: misurazione delle performance del punto O04;

Desiderabili

- D01: implementare il problema dei filosofi a cena;
- D02: possibilità di inserire una sequenza di operazioni;

Facoltativi

- F01: inserimento di uno username per mostrarlo nella scoreboard (no login);
- F02: inserimento di un punteggio in base alla percentuale di performance;
- F03: visualizzazione di una scoreboard di tutti gli utenti;

Come anticipato, questi obiettivi sono stati in parte rivisti in corso d'opera insieme alla Prof.ssa Ombretta Gaggi, e sono stati aggiunti altri obiettivi, che sono stati classificati come segue:

Obbligatori

- O01: implementazione dell'applicativo;
- O02: funzionamento base dello stesso;
- O03: primo problema: lettore - scrittore (molto più adatto al contesto);
- O04: Inserimento di username e scuola di provenienza per tenere traccia dell'utilizzo dell'applicativo;

Facoltativi

- F01: implementare il problema dei filosofi a cena;
- F02: possibilità di inserire una sequenza di operazioni;
- F03: Visualizzazione delle statistiche di risposte ai quesiti posti agli utenti;

Desiderabili

- D01: Creazione di un'area riservata per l'amministratore senza necessità di login particolari;
- D02: visualizzazione di una dashboard con tutti gli utenti;
- D03: simulazione della sequenza di operazione eseguita come live demo;

Alla fine del progetto, la percentuale di obiettivi completati risulta la seguente:

- Obbligatori: 100%

- Desiderabili: 70%
- Facoltativi: 70%

con un totale di obiettivi completati pari al 70%.

COMMENTO (questo dato dovrebbe cambiare in futuro, in base all'implementazione dei video simulazione e dei grafici delle risposte)

2.3 Tecnologie utilizzate

Le tecnologie utilizzate nel progetto e la loro selezione sono state fatte in comune accordo con la Prof.ssa Ombretta Gaggi, tenendo conto delle competenze pregresse, acquisite durante il corso di *Ingegneria del Software* e delle tecnologie più moderne e adatte al contesto del progetto. Tali scelte sono state fatte con l'obiettivo di assicurare un prodotto finale di alta qualità, facilmente manutenibile e modificabile in futuro, e che potesse essere utilizzato da altri studenti come me, garantendo anche una rapida curva di apprendimento.

In particolare, ho scelto di utilizzare [NextJS](#), che è un framework di [ReactJS](#) per lo sviluppo di applicazioni web, e [TailwindCSS](#), un framework CSS che permette di creare interfacce utente reattive e personalizzabili in modo semplice e veloce.

2.3.1 NextJS

Nonostante fin da subito fosse stato chiaro che non sarebbe stato necessario un backend, ho comunque optato per NextJS, perché sapevo che avrei dovuto implementare funzionalità che richiedevano di effettuare chiamate alle [API](#) di [GitHub](#). Inoltre, grazie al rendering lato server (SSR) e alla generazione statica delle pagine (SSG) con i [server-side components](#), Next permette di limitare il caricamento di codice JavaScript necessario per il rendering della pagina verso il client, rendendo così il sito più leggero sul browser, più veloce e piacevole da utilizzare, e soprattutto più [SEO](#) friendly, quindi, in un'ottica

aziendale, più *appealing* a livello marketing. COMMENTO(non so se aggiungere una parte «teorica» su NextJS)

2.3.2 TailwindCSS

Ho scelto di utilizzare TailwindCSS per la sua flessibilità e per la sua capacità di creare interfacce utente reattive e personalizzabili in modo semplice e veloce. Inoltre, grazie alla sua natura *utility-first*, permette di scrivere meno codice CSS, rendendo il progetto più leggero e veloce da caricare, senza preoccupazioni di conflitti tra stili dati dalla rigidità della gerarchia di specificità di CSS.

2.4 Prodotti attesi

Al completamento del progetto, è prevista la consegna di un prodotto funzionante che rispetti i requisiti definiti in fase di progettazione.

Sarà consegnato il codice sorgente del progetto, che sarà ben documentato e commentato, in modo da permettere una facile comprensione e manutenzione del codice stesso. Inoltre, sarà fornita una documentazione tecnica che descrive le funzionalità implementate, le tecnologie utilizzate e le modalità di utilizzo del prodotto, oltre che un breve documento che spiega all'utente come utilizzare l'applicativo, e un documento che spiega ad eventuali futuri programmatori come è strutturato il codice, per guidarli in un eventuale manutenzione o potenziamento del progetto.

Oltre che alla documentazione tecnica e al codice sorgente, sarà prodotto questo documenti di tesi, che descrive appunto il progetto sotto ogni punto di vista e ne analizza i risultati ottenuti, le difficoltà incontrate e le soluzioni adottate.

Capitolo 3

Implementazione

Espongono come le scelte descritte nel capitolo precedente sono state implementate, attraverso una panoramica del codice sorgente, delle feature di accessibilità e delle best practices attuate durante il processo di sviluppo.

Capitolo 4

Retrospettiva finale

Glossario

API: Interfaccia di programmazione delle applicazioni. Permette la comunicazione tra diverse applicazioni o servizi. 10

Git: Sistema di controllo versione distribuito. Permette di tenere traccia delle modifiche apportate a file e cartelle nel tempo. 2

GitHub: Piattaforma di hosting per progetti software che utilizza Git come sistema di controllo versione. Permette la collaborazione tra sviluppatori e la gestione del codice sorgente. 2, 10

Kanban: Sistema di gestione del lavoro che utilizza schede per visualizzare il flusso di lavoro e le attività in corso. 3

NextJS: Framework React per lo sviluppo di applicazioni web. Permette il rendering lato server e la generazione statica delle pagine. 10

Piano di Lavoro: Piano di lavoro. Documento che descrive le attività e gli obiettivi di un tirocinio o di un progetto. 2

ReactJS: Libreria JavaScript per la creazione di interfacce utente. Permette di costruire applicazioni web complesse e reattive in modo efficiente. 10

SEO: ing: Search Engine Optimization. Ottimizzazione per i motori di ricerca. Insieme di tecniche e pratiche per migliorare la visibilità di un sito web sui motori di ricerca. 10

Scrum: Framework Agile per la gestione dei progetti. Si basa su iterazioni brevi (sprint) e su riunioni regolari per monitorare i progressi. 4

TailwindCSS: Framework CSS per la creazione di interfacce utente reattive e personalizzabili. Utilizza classi utility per semplificare lo sviluppo front-end. 10

WebApp: abbr. Applicazione Web iv

backlog: Backlog. Elenco di attività o funzionalità da completare in un progetto. In questo caso, si riferisce all'elenco delle funzionalità e dei bug da risolvere nel progetto. 3

server-side components: Componenti lato server. Permettono di eseguire il rendering delle pagine sul server prima di inviarle al client, migliorando le prestazioni e l'ottimizzazione SEO. 10

sprint: Sprint. Periodo di tempo definito in cui un team di sviluppo (o in questo caso, un solo dev) lavora per completare un insieme specifico di attività o obiettivi. 3

Bibliografia

- [1] Robert C. Martin, Kent Beck, Martin Fowler, e altri, «Manifesto test Agile Software Development». [Online]. Disponibile su: <http://agilemanifesto.org/>