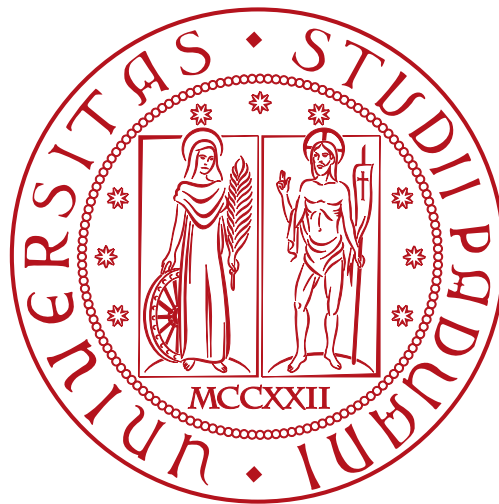


# Università degli Studi di Padova

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 <sub>G</sub> 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

<b>Introduzione</b>	<b>1</b>
1.1 Motivazioni e Contesto .....	1
1.2 Strumenti e processi .....	2
1.2.1 Suddivisione del lavoro .....	3
1.3 Struttura del Documento .....	4
<b>Scopo del tirocinio</b>	<b>7</b>
2.1 Scopo del progetto .....	7
2.2 Obiettivi prefissati .....	7
2.3 Tecnologie utilizzate .....	7
2.3.1 NextJS .....	8
2.4 Prodotti attesi .....	8
<b>Svolgimento del progetto</b>	<b>9</b>
<b>Retrospettiva finale</b>	<b>11</b>
<b>Glossario</b>	<b>13</b>
<b>Bibliografia</b>	<b>15</b>

# 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», 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 competenze

matematiche e logiche che sono fondamentali per affrontare questo percorso.

Lo scopo di questo tirocinio, quindi, è proprio quello di dimostrare a tutti gli studenti che hanno un interesse in informatica, che non importa quale sia il loro *background* scolastico per affrontare questo percorso.

Altro obiettivo molto importante, è quello di aumentare il numero di iscritti donna, 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 G](#) 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](#)<sup>[1.1]</sup>, divisa in 3 colonne:

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

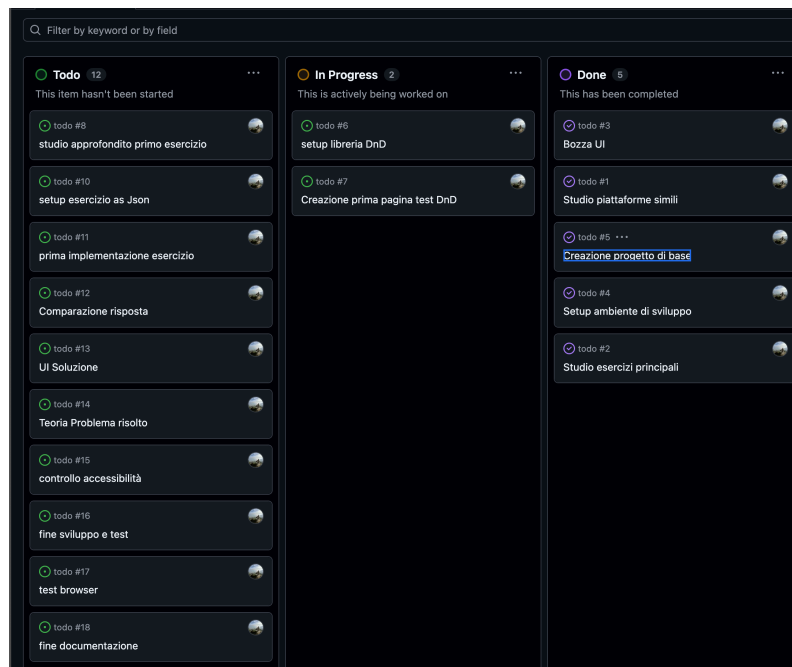


Figura 1.1: vista della Kanban alla seconda settimana

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

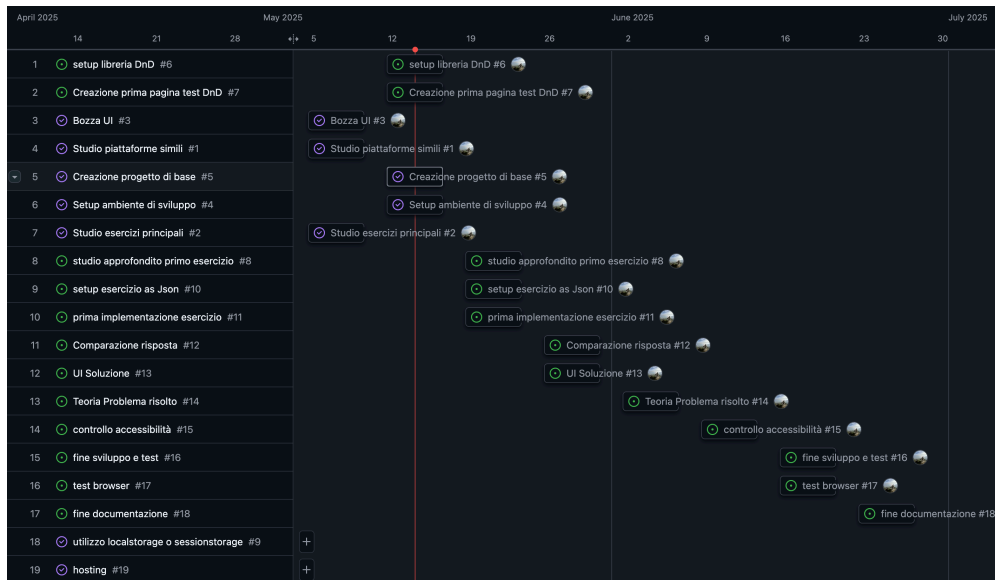


Figura 1.2: vista diagramma di Gantt alla seconda settimana

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<sup>[11]</sup>, 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<sub>G</sub>](#).
- 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

### 2.2 Obiettivi prefissati

### 2.3 Tecnologie utilizzate

Dopo una breve riflessione insieme alla mia relatrice, abbiamo concordato sull'utilizzo di tecnologie con cui avevo già familiarità, avendole usate anche durante il progetto di *Ingegneria del Software*.

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

Ho preso questa decisione con l'obiettivo non solo di crescere a livello di conoscenze, ma soprattutto con la cognizione di voler creare un prodotto che, seppur potrebbe non essere utilizzato, sia comunque creato con tecnologie moderne, facilmente mantenibile e modificabile in futuro da altri studenti come me.

### 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 e alla generazione statica delle pagine 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.

NextJS ha anche un sistema di routing che permette di creare pagine in modo semplice e veloce, senza dover configurare manualmente le varie rotte, attraverso l'utilizzo di file e cartelle.

## 2.4 Prodotti attesi



## **Capitolo 3**

# **Svolgimento del progetto**

*bruh.*



## **Capitolo 4**

# **Retrospettiva finale**



# Glossario

**API:** Interfaccia di programmazione delle applicazioni. Permette la comunicazione tra diverse applicazioni o servizi. 8

**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, 8

**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. 7

**Piano di Lavoro:** Piano di lavoro. Documento che descrive le attività e gli obiettivi di un tirocinio o di un progetto. 3

**ReactJS:** Libreria JavaScript per la creazione di interfacce utente. Permette di costruire applicazioni web complesse e reattive in modo efficiente. 7

**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. 8

**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. 7

*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. 8

*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/>