

## **Documentazione Progetto: Unisci-Puntini**

<b>Titolo del progetto:</b>	Documentazione Progetto: Unisci-Puntini
<b>Alunno/a:</b>	Lorenzo Di Stefano
<b>Classe:</b>	Info I3BB
<b>Anno scolastico:</b>	2023/2024
<b>Docente responsabile:</b>	Geo Petrini

1	Introduzione.....	3
1.1	Informazioni sul progetto.....	3
1.2	Abstract .....	3
1.3	Scopo .....	3
2	Analisi .....	4
2.1	Analisi del dominio .....	4
2.2	Analisi e specifica dei requisiti .....	4
2.2.1	Spiegazione elementi tabella dei requisiti:.....	<b>Fehler! Textmarke nicht definiert.</b>
2.3	Use case .....	7
2.4	Pianificazione .....	7
2.5	Analisi dei mezzi.....	8
2.5.1	Software .....	8
2.5.2	Hardware.....	8
3	Progettazione .....	8
3.1	Design dell'architettura del sistema .....	9
3.2	Design dei dati e database.....	<b>Fehler! Textmarke nicht definiert.</b>
3.3	Design delle interfacce .....	9
3.4	Design procedurale .....	10
4	Implementazione .....	11
5	Test.....	27
5.1	Protocollo di test.....	27
5.2	Risultati test.....	30
5.3	Mancanze/limitazioni conosciute.....	30
6	Consuntivo.....	30
7	Conclusioni .....	30
7.1	Sviluppi futuri.....	30
7.2	Considerazioni personali.....	30
8	Glossario .....	30
9	Bibliografia .....	31
9.1	Bibliografia per articoli di riviste:.....	<b>Fehler! Textmarke nicht definiert.</b>
9.2	Bibliografia per libri.....	31
9.3	Sitografia .....	31
10	Allegati .....	31

## 1 Introduzione

### 1.1 Informazioni sul progetto

Il Progetto dell'applicativo web “Unisci Puntini” è stato sviluppato dalle seguenti persone nelle tempistiche riportate.

- Progettista: Lorenzo Di Stefano
- Designer: Lorenzo Di Stefano
- Web Developer: Lorenzo Di Stefano
- Inizio Progetto: 08.09.2023
- Consegna Progetto: 01.12.2023

### 1.2 Abstract

- **Situazione iniziale:** Scarse conoscenze nel framework TypeScript e nella suite SVG, discrete conoscenze in HTML, CSS e JavaScript.
- **Problematiche:** La problematica affrontata è stata trovare una suite di disegno più appropriata per creare un unisci puntini. Altra problematica è stata trovare il modo di far adattare la grandezza della pagina all'elemento SVG dato che impossibilitato di aggiungere grandezze in percentuale.
- **Approccio/Metodi:** è stata prima creata la pagina HTML nella quale sono stati inseriti gli elementi di interazione di modo che si potesse testare le nuove aggiunte passa passo durante il progetto.
- **Risultati:** Questo prodotto non è completo come suite di disegno professionali ma comprende le funzioni essenziali per lo scopo per cui creato ed è più accessibile in facilità d'utilizzo.

### 1.3 Scopo

Lo scopo di questo progetto è quello di permettere a chiunque di creare un unisci puntini in modo facile e veloce, da ogni piattaforma. Su internet sono presenti svariati siti e applicativi che permettono di fare ciò, ma spesso questi sono incompleti di tutte le funzionalità desiderate dall'utente o sono troppo complessi. L'applicativo in questione è la soluzione, rendendo questa mansione più accessibile a chiunque.

## 2 Analisi

### 2.1 Analisi del dominio

Il prodotto è concepito per essere utilizzato da insegnanti delle scuole materne, asili, scuole elementari genitori e bambini. Lo scopo del progetto è di poter creare i propri unisci puntini da unire, colorare e intrattenersi nella creazione di nuovi unisci puntini.

### 2.2 Analisi e specifica dei requisiti

Mi è stato incaricato di creare un applicativo che permetta all'utente di creare un unisci puntini. Questo applicativo deve poter essere utilizzato da ogni dispositivo, avere un'interfaccia grafica e seguire i punti sottoelencati:

ID: REQ-001	
<b>Nome</b>	Gestione import / export dei file
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Bisogna gestire i formati (.jpg .png. webb .svg)
Sotto requisiti	
<b>001</b>	Bisogna poter importare le immagini
<b>002</b>	Bisogna poter esportare il progetto
<b>003</b>	Durante l'esportazione bisogna poter scegliere se caricare solo i puntini o anche la soluzione

ID: REQ-002	
<b>Nome</b>	Posizionamento dei puntini sull'immagine
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	
Sotto requisiti	
<b>001</b>	I puntini devono poter essere spostati
<b>002</b>	Si devono poter aggiungere puntini nel percorso tra altri già posizionati
<b>003</b>	Si deve poter riordinare l'ordine dei puntini
<b>004</b>	I puntini devono essere numerati automaticamente.

ID: REQ-003	
<b>Nome</b>	Creazione della soluzione dell'unisci puntini

<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	
<b>Sotto requisiti</b>	
<b>001</b>	La costruzione della soluzione deve essere autonoma
<b>002</b>	Di deve poter attivare / disattivare la costruzione della soluzione

<b>ID: REQ-004</b>	
<b>Nome</b>	Utilizzo di strumenti di disegno per modificare il file
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	
<b>Sotto requisiti</b>	
<b>001</b>	Si deve poter modificare il colore
<b>002</b>	Si deve poter modificare la grandezza
<b>003</b>	

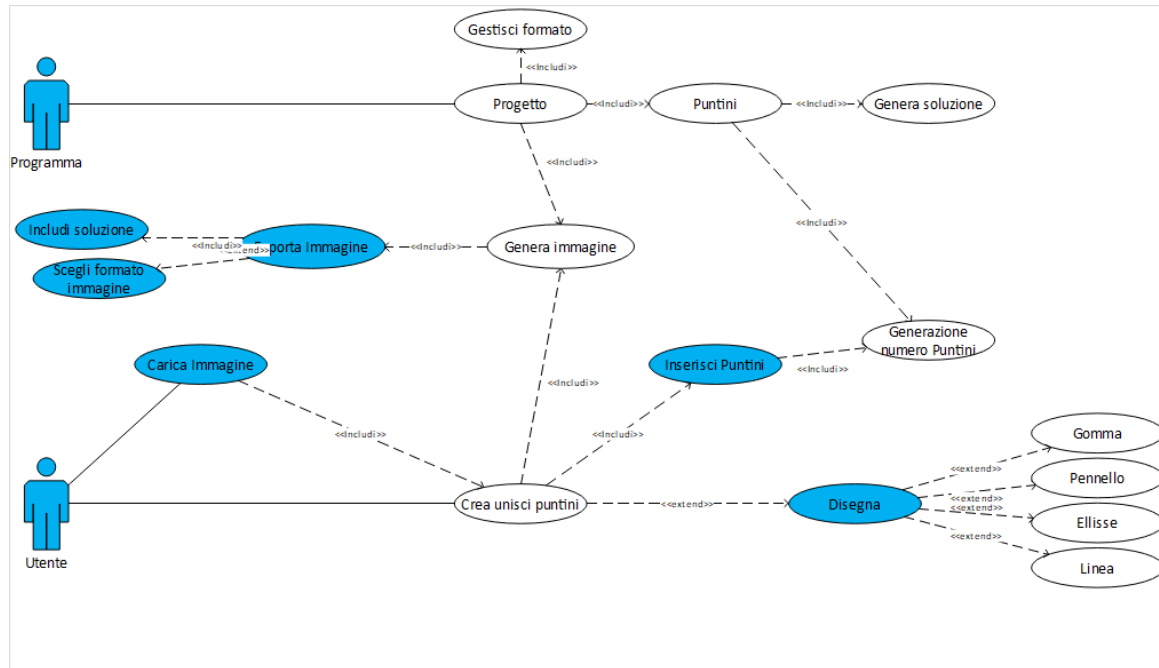
ID: REQ-005	
<b>Nome</b>	Navbar in alto per gestire le opzioni
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	
Sotto requisiti	
<b>001</b>	Presenza di un hamburger per aprire un menu a scomparsa
<b>002</b>	
<b>003</b>	

ID: REQ-006	
<b>Nome</b>	Gestione dei layer
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	
Sotto requisiti	
<b>001</b>	Si devono poter nascondere i layer
<b>002</b>	Si devono poter scartare i layer
<b>003</b>	È necessario poter vedere i layer contemporaneamente

ID: REQ-007	
<b>Nome</b>	Predisposizione GUI
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	La GUI è divisa in tre sezioni principali
Sotto requisiti	
<b>001</b>	C'è la sezione per gli strumenti di disegno
<b>002</b>	Al centro è presente il foglio di disegno
<b>003</b>	Nella terza sezione è presente il gestore dei layer

## 2.3 Use case

I casi d'uso rappresentano l'interazione tra i vari attori e le funzionalità del prodotto.



## 2.4 Pianificazione

Gantt del Progetto:

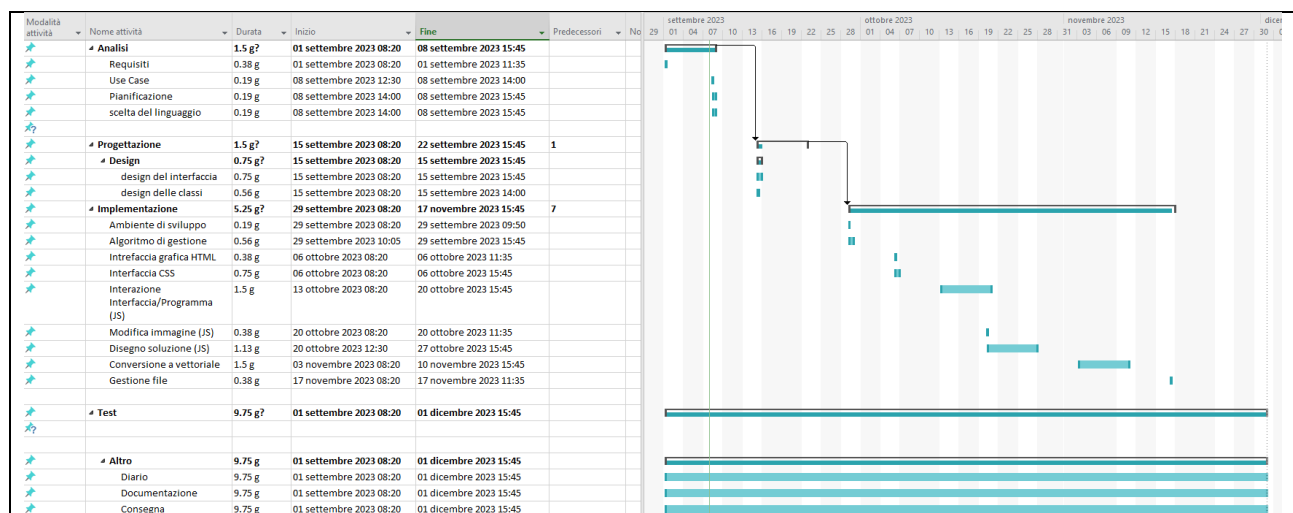


Figura 1: Esempio di diagramma di Gantt.

## 2.6 Analisi dei mezzi

Il Progetto è affidato ad un singolo allievo, durante le giornate scolastiche di venerdì. L'intervallo di tempo effettivo per lavorare al progetto è compreso tra il 22.09.2023 e il 01.12.2023.

Ore disponibili: 80  
Peso massimo dell'applicativo: Non specificato

### 2.6.1 Software

Il Progetto è sviluppato come Applicazione Web, utilizzando il framework TypeScript che necessita l'utilizzo di Node.

Browser:

Firefox (Non basato su Chromium)  
Edge (Basato su Chromium)

VisualStudio Code:

[Estensioni]  
- Typescript (Microsoft)  
- JavaScript (ES6) code snippets (charalampos karypidis)  
- HTML Boilerplate (sidthesloth)  
- Live Server (Ritwick Dey)  
- Emoji (Perkovec)

Node.js:

[Librerie]  
- Typescript

### 2.6.2 Hardware

#### UTILIZZO

Il prodotto è sviluppato su piattaforma web ed è concepito per essere utilizzato da postazioni PC (Windows/Linux/macOS) con la possibilità di eseguire azioni anche da dispositivi mobile (ciò comporta alcune limitazioni come: difficoltà nell'utilizzo di alcuni strumenti concepiti per essere utilizzati con mouse e lo schermo ridotto rispetto ad altre piattaforme).

#### SVILUPPO

PC desktop scolastico (lavoro in presenza)

Specifiche:

OS: Windows 10 enterprise  
CPU: Intel I7 9700  
GPU: Nvidia GeForce RTX 2060  
RAM: 32GB RAM DDR4

PC desktop personale (lavoro da remoto)

Specifiche:

OS: Windows 11 pro  
CPU: Intel I7 6700  
GPU: Intel Graphics 530  
RAM: 16GB RAM DDR4



### 3 Progettazione

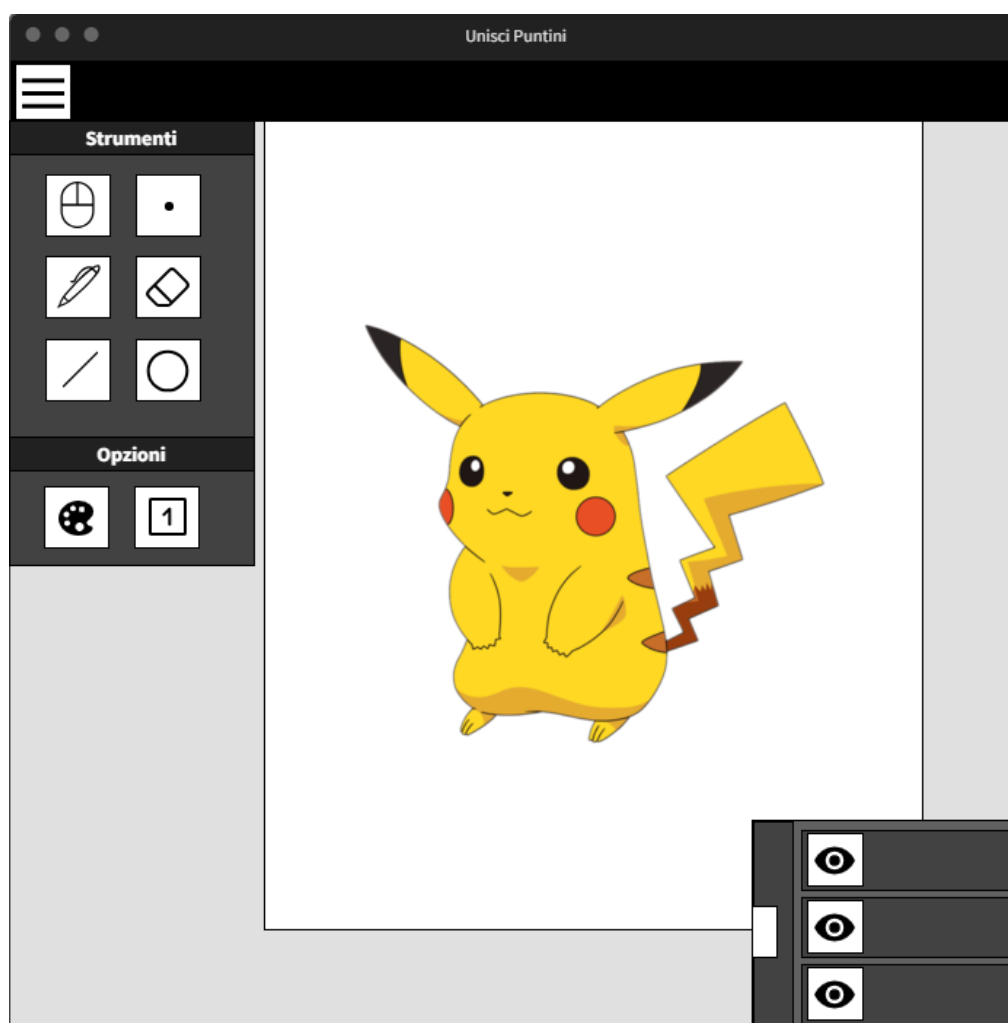
#### 3.1 Design dell'architettura del sistema

L'applicativo web presenta la seguente struttura.

```
-css
-img
-js
|-- <file compilati>
-test
-ts
|-- <file sorgente>
```

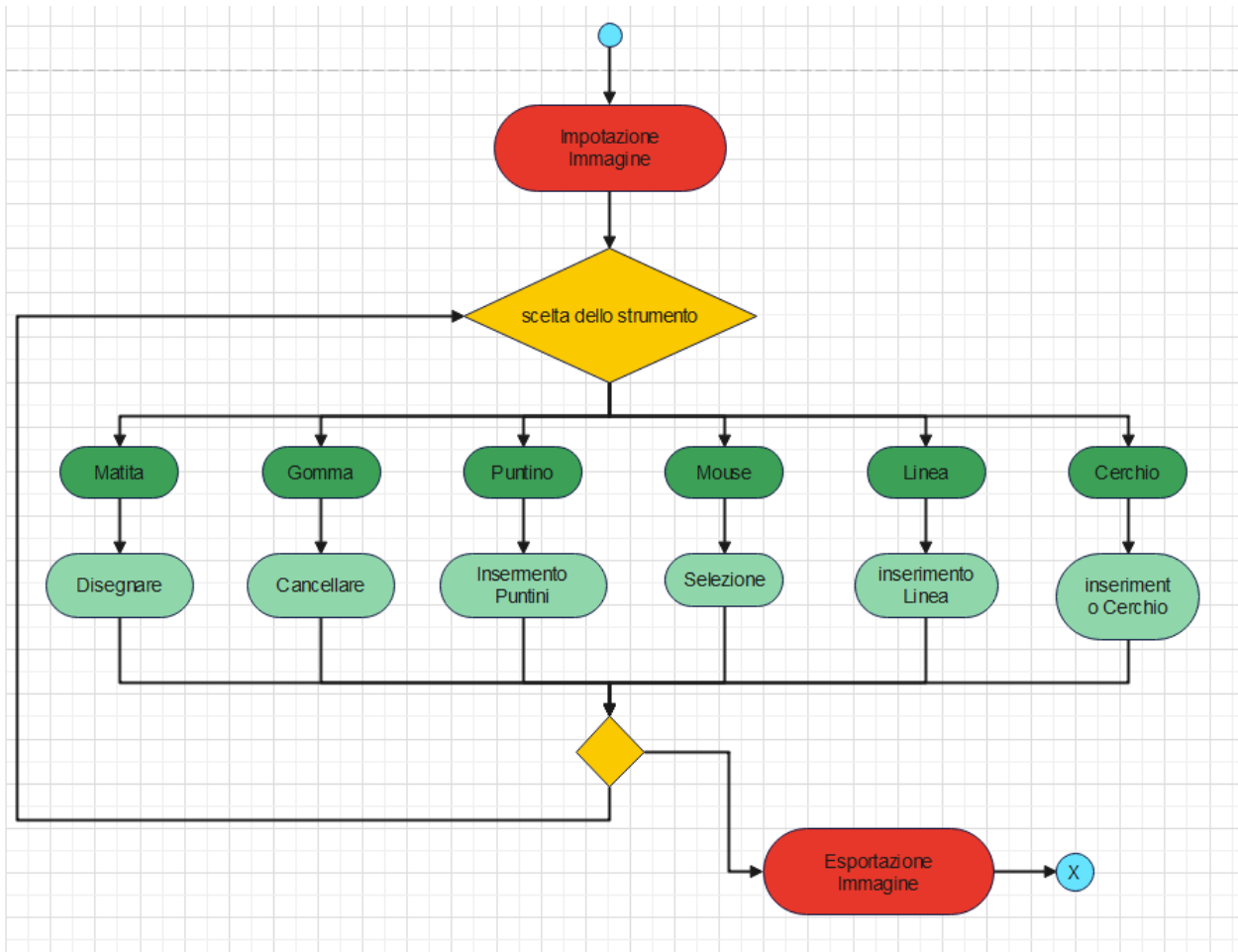
#### 3.2 Design delle interfacce

Secondo la progettazione, l'applicativo avrà una Navbar in alto avente un hamburger a sinistra, una sezione contenete gli strumenti di disegno in alto a sinistra, una sezione contenete i tasti di visualizzazione dei layer in basso a destra e la pagina di disegno al centro.



### 3.3 Design procedurale

Lo schema illustrato mostra il design procedurale dell'applicativo.



## 4 Implementazione

### 4.1 Scelta tra Canvas & SVG

In HTML per disegnare, vi sono due principali opzioni. La prima opzione è utilizzare Canvas, mentre la seconda è di utilizzare SVG.

#### Canvas

Il Tag Canvas è dipendente dalla risoluzione, causando una carenza di dettagli nel momento in cui viene fatto uno zoom. Canvas però ha il vantaggio di renderizzare le immagini più velocemente rispetto al tag SVG. Questo rende Canvas una suite migliore per lo sviluppo di giochi, ma data la scarsa complessità delle immagini gestite, ciò non risulta necessario.

#### SVG

SVG contrariamente a Canvas non è dipendente dalla risoluzione, essendo in formato vettoriale. Ciò è limitante quando si deve creare immagini complesse che risulterebbero più facili da gestire tramite un formato bitmap, ma non è questo il caso essendo che le immagini create sono dei basilari disegni geometrici aventi possibilità di essere modificati tramite una serie di strumenti, permettendo di mantenere il massimo della risoluzione senza spreco di risorse.

Un altro vantaggio del tag SVG è il fatto che gli elementi inseriti sempre diversamente a Canvas vengono aggiunti come elementi DOM alla pagina HTML rendendo estremamente più facile la modifica di essi.

## 4.2 Disegno in SVG HTML

Nel progetto, per disegnare gli elementi, è stato deciso di utilizzare il tag SVG di HTML. Ovviamente l'esempio qui mostrato serve dare un'idea di come saranno costruiti i puntini con il fine di visualizzare i parametri da inserire come membri della classe che definisce l'oggetto **Point**.

Ogni volta che si premerà sullo schermo per inserire un puntino verrà istanziato un nuovo oggetto della classe **NumberedPoint** che verrà generato sulla pagina.

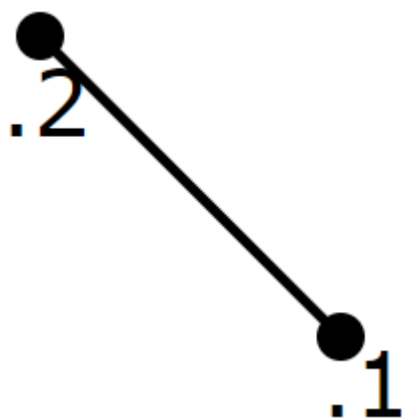
La tabella sottostante rappresenta il codice HTML necessario per creare 2 puntini collegati da una linea:

```
<!DOCTYPE html>
<html>
<body>
<svg width="500" height="500">
  <!--Primo Punto-->
  <circle cx="200" cy="200" r="10"
    stroke="black" stroke-width="4" fill="black" />
  <text fill="#000000" font-size="45" font-family="Verdana" x="190"
    y="240">.1</text>

  <!--Secondo Punto-->
  <circle cx="50" cy="50" r="10"
    stroke="black" stroke-width="4" fill="black" />
  <text fill="#000000" font-size="45" font-family="Verdana" x="30"
    y="100">.2</text>

  <!--Linea-->
  <polyon points="200,200 50,50 "
    style="fill:none;stroke:black;stroke-width:5;fill-rule:evenodd;" />
  Sorry, your browser does not support inline SVG.
</svg>
</body>
</html>
```

Il codice nella tabella precedente genera il seguente risultato.



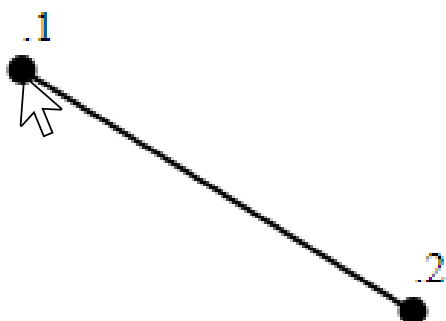
### 4.3 GUI

L'applicativo gode di un design semplice ed intuitivo. La prima sezione nominata “Strumenti” permette di selezionare l'utensile desiderato per creare il proprio unisci puntini personalizzato. Sottostante alla prima sezione, vi stanno le “Opzioni” che permettono di scegliere il colore e la grandezza degli elementi da inserire.



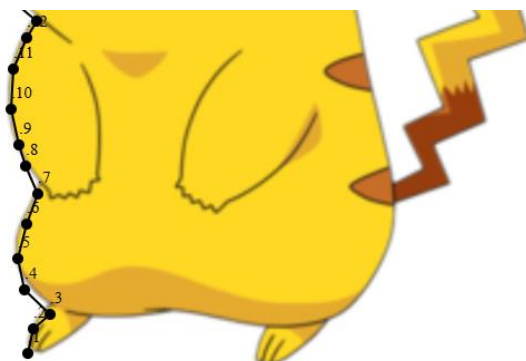
#### MOUSE

Lo strumento Mouse può essere utilizzato per muovere i puntini cliccando su di essi e tendo premuto muovendosi nella pagina per spostarli. Nel caso si volesse rimuovere un puntino, bisognerebbe cliccare tasto destro sul puntino selezionato.



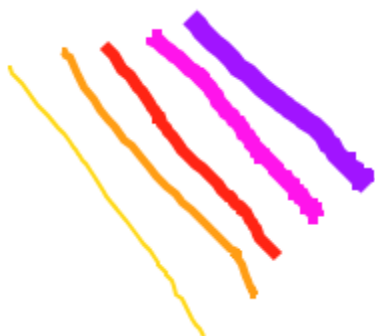
## PUNTINO

Selezionato il Puntino, sarà possibile inserire dei puntini numerati cliccando sulla pagina. Nei casi si inserissero erroneamente puntini di troppo o in punti sbagliati sarà sempre possibile spostarli o eliminarli.



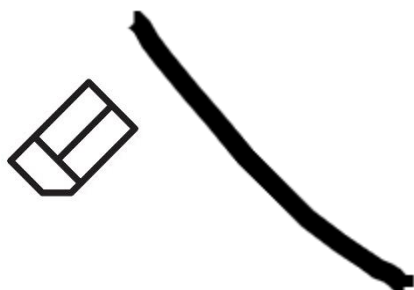
## MATITA

Lo strumento matita permette di disegnare delle linee a mano libera. Ad ogni click verrà creato un nuovo segmento separato dal precedente. Essi potranno avere più colori, e grandezze.



## GOMMA

Lo strumento gomma permette di eliminare i componenti inseriti con gli strumenti: cerchio, matita, linea. L'eliminazione comprenderà l'intero elemento selezionato. Se si volesse ad esempio eliminare parte di un segmento disegnato a matita, ciò non sarà possibile dato che verrà eliminato l'intero segmento.



## LINEA

Selezionato questo strumento, è possibile disegnare linee segmentate. Ad ogni click effettuato sullo schermo a strumento selezionato verrà aggiunto un nuovo segmento alla linea segmentata appena creata. Per creare una linea segmentata separata, deselegionare lo strumento e rirrelazionarlo. Esse potranno avere più colori, e grandezze.



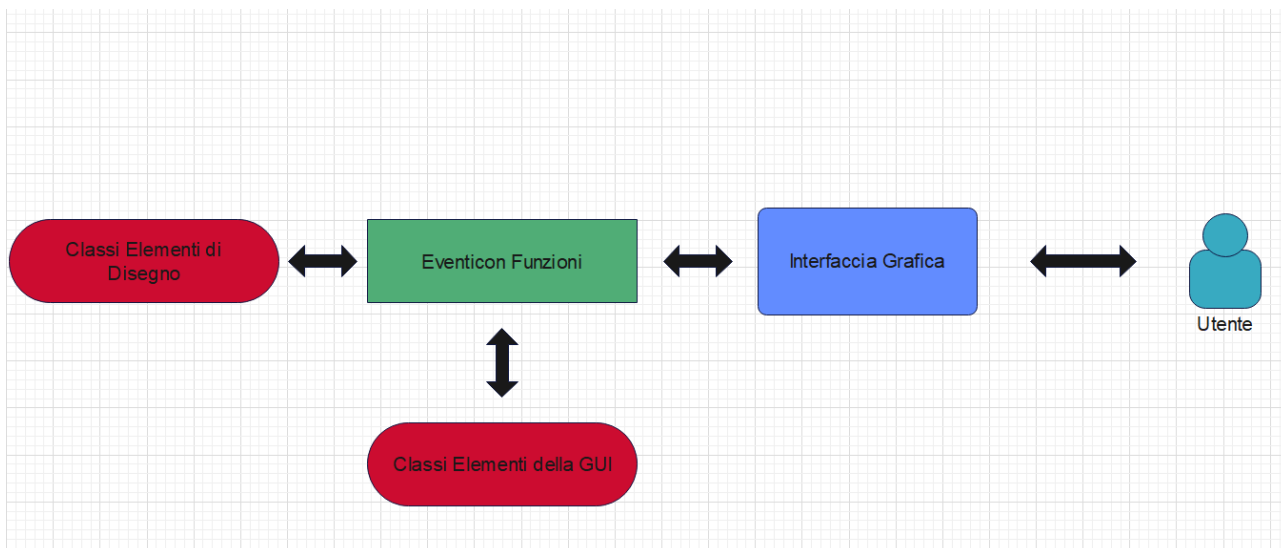
## CERCHIO

Lo strumento cerchio permette di inserire cerchi di dimensione e colore completamente personalizzabile. Selezionato lo strumento, cliccare con il mouse nella zona desiderata per inserire il cerchio, muovere il mouse fino a raggiungere la dimensione desiderata e cliccare una seconda volta per confermare l'inserimento del cerchio.



#### 4.4 Interazione GUI / Applicativo

In questo progetto gli elementi di interazione della GUI dell'applicativo sono affiancati da classi per facilitare la modularità del progetto. Gli oggetti istanziati che affiancano gli elementi HTML rimangono in ascolto, richiamando le funzioni corrispondenti allo scatenamento di eventi specifici. Gli eventi scatenati possono richiamare funzioni che vanno a interagire con quello che può essere lo spostamento o l'inserimento di elementi nella pagina di disegno, il salvataggio dell'immagine o altre interazioni.



#### PULSANTI

All'interno del programma, elementi come pulsanti sono gestiti da classi. Questo permette di semplificare la gestione di questi elementi, dato che per ogni pulsante aggiunto vanno modificati pochi parametri al nuovo oggetto per assegnarli la funzione desiderata.

I pulsanti vengono dichiarati nel file **main.ts**, nel quale vengono specificati i seguenti parametri per ogni pulsante: funzione da eseguire alla pressione del pulsante, immagine default di sfondo (scr), immagine di sfondo a pulsante premuto (scr), se tener conto dei click del pulsante come se fosse un **ON/OFF** (boolean) e id dell'elemento in questione.

```

export const MOUSE_BTN = new Button(
    operations.useMouse, // Funzione da eseguire
    "../img/mouse.png", // Immagine di default
    "../img/mouse.png", // Immagine a pulsante premuto
    true,                // attivare il contatore ON/OFF
    "toolbtn1"           // Id del pulsante (HTML)
);
  
```



## 4.5 Elementi di Disegno

L'applicativo presenta le seguenti classi necessarie a descrivere gli elementi DOM da inserire:

- DrawingElement
  - Circle
  - Eraser
  - Line
  - Point
  - Text
  - NumberedPoint

### DrawingElement

La classe DrawingElement è la classe padre delle classi sottoelencate che serve ad astrarre gli elementi comuni tra le altre classi.

Essa contiene variabili corrispondenti all'elemento SVG alla quale aggiungere il DOM, la variabile corrispondente all'elemento DOM relativo alla forma in questione.

#### DrawingElement

```
-element: HTMLElement
-svg: HTMLElement
-position: Position
-width: number
-height: number
-color: string
+constructor(id: string): constructor
+getElement(): HTMLElement
```

### Position

Position non è una classe effettiva, ma bensì una variabile di tipo oggetto nella quale sono memorizzate le coordinate x & y. L'inserimento di tale oggetto semplifica l'utilizzo delle coordinate all'interno delle classi.

#### Position

```
x: number
y: number
```

## Circle

Circle è una classe che definisce un oggetto che permette di inserire elementi DOM di tipo Circle. All'interno di questa classe viene istanziato un Circle necessario alla gestione della posizione del centro di esso.

### getRadiusSize()

Il metodo getRadiusSize() permette di modificare la grandezza del Circle all'inserimento. I parametri x & y rappresentano le coordinate del mouse al momento di inserimento. Il valore che viene ritornato corrisponde al raggio calcolato dalla distanza tra la posizione del Point centrale al Circle e la posizione del Mouse.

```
private getRadiusSize(x:number, y:number){
    var deltaX = Math.abs(this.point.position.x-x);
    var deltaY = Math.abs(this.point.position.y-y);
    this.radius = Math.sqrt(Math.pow(deltaX,2)+Math.pow(deltaY,2));
    return this.radius;
}
```

### preview()

Il metodo preview() ridisegna il Circle nella dimensione del raggio restituita dal metodo getRadiusSize()

```
preview(x:number, y:number): void{
    this.element.setAttribute("r" , String(this.getRadiusSize(x,y)));
}
```

## Eraser

La classe Eraser serve descrivere un oggetto necessario ad eliminare gli elementi DOM dalla pagina di disegno facenti parte della classe

### getArea()

Questo metodo restituisce un valore boolean, che verifica che l'elemento alla posizione dei x & y corrispondenti ai parametri del metodo rientrino nella zona occupata dalla gomma.

Questo metodo è utilizzato dai metodi **lines()**, **circles()** e **pencil()** per eliminare gli elementi nella quale **getArea()** restituisce il valore **true**;

```
private getArea(radius:number = 0,x:number, y:number): boolean{
    if(
        x + radius >= (this.position.x - this.radius * 2 *
            counters.controls.values.size) &&
        y + radius >= (this.position.y - this.radius * 2 *
            counters.controls.values.size) &&
        x - radius <= (this.position.x + this.radius * 2 *
            counters.controls.values.size) &&
        y - radius <= (this.position.y + this.radius * 2 *
            counters.controls.values.size)
    ){
        return true;
    }else{
        return false;
    }
}
```

## Eraser

```
- -points[]: Point[]
- coordinate: string
+constructor(radius:number, x: number,
y:number): constructor
+getArea(x: number, y:number): boolean
+move(x: number, y:number): void
+remove(): void
-lines(): Line
-pencil(): Line
+draw(): void
+undraw(): void
```

## Line

La classe Line basa la costruzione del proprio elemento DOM su un array di Point che usa come appoggio per il disegno dei segmenti.

### addPoint()

Il metodo utilizza i parametri x & y per istanziare uno nuovo Point nell'array, seguentemente esso viene ciclato per salvare le i valori della posizione nella variabile testuale coordinate, utilizzata per la ricostruzione dell'elemento DOM nella pagina.

```
public addPoint(x:number,y:number): void{
    var id = "lp"+this.points.length;
    this.points.push(new Point(1,String(id),x,y));
    this.coordinate = "";
    for(var i=0;i<this.points.length;i++){
        this.points[i].getElement().style.display = "none";
        this.coordinate += " "+this.points[i].getPosition().x
        +", "+this.points[i].getPosition().y;
    }
    this.draw();
}
```

## Line

```
- points[]: Point[]
- coordinate: string

+ constructor(id: string, x: number,
y:number): constructor
+ getPoints(): Point[]
+ addPoint(x:number, y:number): void
+ move(index:number, x:number,
y:number): void
+ remove(): void
+ draw(): void
```

## Point

La classe Point serve a descrivere un oggetto di costruzione utilizzato da altre classi come: NumberedPoint, Line, Circle.

### checkIsMoving

Questo metodo permette di constatare che il mouse sia sopra il puntino settando la variabile isMoving su true se vero altrimenti la setta su false. La posizione del mouse viene inserita dai parametri x e y dal file nella quale viene gestito l'evento.

```
checkIsMoving(x: number, y: number): void {
    if(
        x >= (this.position.x - this.radius*2) &&
        y >= (this.position.y - this.radius*2) &&
        x <= (this.position.x + this.radius*2) &&
        y <= (this.position.y + this.radius*2)
    ){
        this.isMoving = true;
    }else{
        this.isMoving = false;
    }
}
```

## Point

```
-radius: number
-isMoving: boolean
+constructor(radius:number, id: string, x:
number, y:number): constructor
+getIsMoving(): boolean
+getPosition(): Position
+setIsMoving(): void
+setId(): void
+checkIsMoving(x:number, y:number):
boolean
+move(x:number, y:number): void
+draw(): void
+remove() void
```

## Text

La classe Text serve da elemento complementare da essere utilizzato per la costruzione della classe del puntino numerato. La funzione della classe Text la sola gestione dell'elemento DOM Text in modo semplificato. Il motivo dell'implementazione di questa classe all'interno del progetto è il fatto che implementare la gestione del DOM del Text all'interno della classe NumberedPoint sarebbe stato caotico e avrebbe creato difficoltà nella gestione dell'oggetto.

### Text

-Attributo 1: tipo

```
+constructor(id: string, x: number,
y:number): constructor
+setId(): void
+setTextContent(): void
+move(x:number, y:number): void
+draw(): void
+remove(): void
```

## NumberedPoint

La classe istanzia sia la classe Text che la classe Point al suo interno. Queste 2 compongono l'elemento grafico di NumberedPoint. La classe Point include anche le funzioni necessarie allo spostamento di esso all'interno della pagina.

NumberedPoint gestisce le 2 istanze, creando un elemento unico. Questo ha permesso di non dover specializzare eccessivamente le classi e rendere il progetto più modulare.

### move()

Il metodo **move()** di **NumberedPoint** va a interagire direttamente con i metodi move delle classi istanziate al suo interno, ciò permette mantenere maggiore ordine e gestire meglio gli errori.

```
move(x: number, y: number): void{
    this.position.x = x;
    this.position.y = y;
    this.point.move(x,y);
    this.text.move(x,y - 10);
}
```

### remove()

Allo stesso modo, im metodo **remove()** richiama i metodi interni alle classi istanziate nella classe. Facendo rimuovere a queste gli elementi DOM. Successivamente verrà settato lo stato della variabile **isRemoved** su **true** di modo che l'oggetto possa essere rimosso dall'array dove memorizzato seguentemente.

```
remove(): void{
    this.point.remove();
    this.text.remove();
    this.isRemoved = true;
}
```

## NumberedPoint

```
-text: Text
-point: Point
-isRemoved: boolean

+constructor(id: string, x: number,
y: number): constructor
+getId(): string
+getPoint(): Point
+getText(): Text
+getIsRemoved
+move(x: number, y: number): void
+remove(value: number): void
```

## 4.6 Gestione Eventi

In questo file, nominato **events.ts** sono presenti listener che richiamano le opportune funzioni allo scatenamento dei relativi eventi. I listener ascoltano gli eventi:

- **mouseup**
- **mousedown**
- **mousemove**
- **mouseclick**
- **contextmenu (click destro)**

Questo file importa operazioni da file esterni da richiamare all'interno di funzioni interne che vengono richiamate durante questi eventi.

### **MouseUp**

Resetta contatori, variabili di stato per cambiare il comportamento di funzioni richiamate da altri eventi.

### **MouseDown**

Imposta valori delle variabili di stato, questo permette di attivare l'incrementazione dei contatori, attivare il funzionamento di funzioni per compiere azioni come l'inserimento di elementi DOM.

### **MouseMove**

Questo evento richiama funzioni che in base allo strumento utilizzato compiono diverse azioni. Queste azioni variano dai valori delle variabili di stato, impostate dagli eventi sopra elencati.

### **MouseClick**

L'evento richiama una funzione che incrementa un contatore (che conta i click), richiama altre funzioni utilizzate per l'inserimento di elementi DOM.

### **ContextMenu**

Questo evento è unicamente utilizzato per cambiare valore ad una variabile di stato che secondo l'impostazione data da questo evento permette di eliminare i puntini numerati.



#### 4.7 Funzioni degli Strumenti di Disegno

Le funzioni necessarie all'utilizzo di tool di disegno contengono al loro interno un controllo tramite un if per verificare che sia effettivamente in utilizzo lo strumento in questione. Il controllo viene fatto su variabili presenti nel file **counters.ts** nella quale vi sono presenti tutte le variabili di controllo.

Esempio

```
export function mouseSelect(){
    // CONTROLLA CHE SIA IN USO LO STRUMENTO MOUSE
    if(operations.actions.useMouse){
        selectNumberedPoint()
    }
}
```

Questo permette di richiamare in massa tutte le funzioni inerenti all'utilizzo di tool nel file events.ts senza dover utilizzare controlli che sporcherebbero il codice.

events.ts

```
// Esegue le funzioni ad ogni click sulla pagina di disegno.
export function pageClickEvent(){
    functionTool.circle();           // Inserisce un Cerchio.
    functionTool.point();            // Inserisce un Puntino Numerato.
    functionTool.line();             // Inserisce un Segmento.
    counters.controls.mouse.clicks++; // Incrementa il contatore.
}
```

#### 4.8 Contatori e Gestione dei Valori

Gran parte dei contatori e dei valori memorizzati da programma sono salvati all'interno di variabili presenti nel file **counters.ts**. Le variabili contengono valori di controllo, come lo stato dei pulsanti del mouse, il numero di click, il numero di azioni compiute dal click, array che contengono le istanze delle classi necessarie all'inserimento di elementi DOM nella pagina di disegno.

Questo file contiene anche funzioni necessarie al riordinamento dei valori dopo l'eliminazione di un elemento dall'array.

Un esempio di ciò è la serie di funzioni per l'eliminazione di un Puntino Numerato.

##### **renumberNumberedPoints()**

Questa funzione viene richiamata all'interno della funzione per la rimozione di un puntino numerato dall'array di contenimento a fine procedura. In questa funzione vengono creati 2 nuovi array. Il primo contiene i valori identici all'array dei puntini originale e il secondo è vuoto. Tramite l'utilizzo di un'ulteriore funzione **minNumberedPoint()** per ottenere il puntino con il valore più piccolo ed inserirlo nel array vuoto, rimuoverlo dall'array con i valori simili a quello originale e rinumerato. Questa operazione viene ripetuta fino allo svuotamento dell'array con valori uguali all'originale tramite un ciclo for.

Questo permette di eliminare puntini facendo in modo che essi vengano rinumerati in modo appropriato.

```
export function renumberNumberedPoints(): void{
    var puntiNumeratiVecchio = puntiNumerati;
    var puntiNumeratiNuovo: NumberedPoint[] = [];
    var indice = puntiNumerati.length;
    for(var i=0;i<indice;i++){
        for(var ii=0;ii<puntiNumeratiVecchio.length;ii++){
            if(
                puntiNumeratiVecchio[ii].getId()==minNumberedPoint()
            ){
                puntiNumeratiVecchio[ii].renumber(i+1);
                puntiNumeratiNuovo.push(puntiNumeratiVecchio[ii]);
                puntiNumeratiVecchio.splice(ii,1);
                puntiNumerati = puntiNumeratiVecchio;
                break;
            }
        }
        puntiNumeratiVecchio = puntiNumerati;
    }
    puntiNumerati = puntiNumeratiNuovo;
}
```

## 5 Test

### 5.1 Protocollo di test

Seguentemente saranno riportati dei test eseguiti manualmente sull'applicativo. Al capitolo 5.2 sarà riportato come approfondire l'esecuzione dei test direttamente sull'applicativo.

<b>Test Case:</b>	TC-001	<b>Nome:</b>	Numerazione dei Puntini Consecutiva
<b>Descrizione:</b>	I Puntini devono essere numerati consecutivamente da 1 a N.		
<b>Prerequisiti:</b>			
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Inserire una sequenza di 8 puntini nella pagina</li> <li>2. Eliminare 3 puntini in posizioni randomiche dalla pagina</li> <li>3. Inserire altri 4 puntini alla pagina</li> </ol>		
<b>Risultati attesi:</b>	Numerazione corretta		

<b>Test Case:</b>	TC-002	<b>Nome:</b>	Selezione Puntini
<b>Descrizione:</b>	I Puntini devono poter essere selezionati per muoverli o rimuoverli.		
<b>Prerequisiti:</b>			
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Inserire una sequenza di 3 puntini nella pagina</li> <li>2. Selezionare lo strumento Mouse e cliccare su un puntino e muovere il mouse tenendo premuto</li> <li>3. Premere tasto destro del Mouse per rimuovere il puntino</li> </ol>		
<b>Risultati attesi:</b>	Selezione corretta		

<b>Test Case:</b>	TC-003	<b>Nome:</b>	Generazione Corretta della soluzione
<b>Descrizione:</b>	A qualsiasi modifica della sequenza di puntini, la soluzione si dovrà adattare di conseguenza.		
<b>Prerequisiti:</b>			
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Inserire una sequenza di 5 puntini</li> <li>2. Muovere 2 puntini in posizioni randomiche</li> <li>3. Eliminare un puntino</li> </ol>		
<b>Risultati attesi:</b>	Costruzione adattata correttamente		

<b>Test Case:</b>	TC-004	<b>Nome:</b>	Inserimento elementi con proprietà custom
<b>Descrizione:</b>	Gli elementi inseriti dovranno avere dimensioni e colore personalizzabile		
<b>Prerequisiti:</b>			
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Selezionare un colore</li> <li>2. Scegliere la grandezza desiderata</li> <li>3. Inserire un elemento casuale all'interno della pagina</li> </ol>		
<b>Risultati attesi:</b>	Personalizzazione avvenuta correttamente		

<b>Test Case:</b>	TC-005	<b>Nome:</b>	Eliminazione elementi
<b>Descrizione:</b>	La gomma deve poter eliminare linee e cerchi disegnati nella pagina		
<b>Prerequisiti:</b>			
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Inserire 2 cerchi</li> <li>2. Inserire 3 linee segmentate</li> <li>3. Disegnare 3 linee a matita</li> <li>4. Eliminare il tutto con la gomma</li> </ol>		
<b>Risultati attesi:</b>	L'eliminazione non avviene sempre in maniera corretta		

## 5.2 Test più approfonditi

Per visualizzare test più approfonditi sull'applicativo è possibile aprire un editor di codice il file index.html e rimuovere il commento nella sezione di codice visualizzata.

```
<!-- ACCESSO AGLI UNIT TEST-->
<!--
<div class="toollabel">Dev</div>
<a href="./test/test.html">Test</a>
-->
```

Post modifica

```
<!-- ACCESSO AGLI UNIT TEST-->
<div class="toollabel">Dev</div>
<a href="./test/test.html">Test</a>
```

Effettuata questa modifica sarà possibile accedere alla pagina di Unit Test da questo link.



Aperto il link si potrà visualizzare la seguente pagina nella quale sono presenti tutti gli Unit Test:

# TEST

## TEST COUNTERS: controls init

Layer 01	Actual value: true	Expected value: true	✓ Test Successful ✓
Layer 02	Actual value: true	Expected value: true	✓ Test Successful ✓
Layer 03	Actual value: true	Expected value: true	✓ Test Successful ✓
Mouse Clicked	Actual value: false	Expected value: false	✓ Test Successful ✓
Mouse Clicks	Actual value: 0	Expected value: 0	✓ Test Successful ✓
Point Selected	Actual value: false	Expected value: false	✓ Test Successful ✓
Elements Color	Actual value: black	Expected value: black	✓ Test Successful ✓

### 5.3 Risultati test

Tabella riassuntiva in cui si inseriscono i test riusciti e non del prodotto finale. Se un test non riesce e viene corretto l'errore, questo dovrà risultare nel documento finale come riuscito (la procedura della correzione apparirà nel diario), altrimenti dovrà essere descritto l'errore con eventuali ipotesi di correzione.

### 5.4 Mancanze/limitazioni conosciute

L'eliminazione di elementi di disegno tramite la gomma non genera sempre il risultato sperato. Spesso passando su un elemento velocemente, quando presenti molti elementi vicini, gli elementi selezionati non vengono eliminati al primo tentativo. Può accadere che essi non vengano eliminati (situazione più rara).

## 6 Consuntivo

Consuntivo del tempo di lavoro effettivo e considerazioni riguardo le differenze rispetto alla pianificazione (cap. 1.7) (ad esempio Gantt consuntivo).

## 7 Conclusioni

La mia soluzione permette di creare unisci puntini e creare dei disegni tramite browser web.

### 7.1 Sviluppi futuri

In attesa di critiche dai consumatori.

### 7.2 Considerazioni personali

Questa esperienza mi ha permesso di approfondire e convalidare le basi di TypeScript apprese durante l'estate. Ho migliorato la mia filosofia di sviluppo di applicativi Grafici.

## 8 Glossario

Termine	Descrizione
CSS	<b>Cascading Style Sheets:</b> linguaggio che permette di definire il layout e la grafica di una pagina web.
TSC	<b>TypeScript</b> è un linguaggio di programmazione open source sviluppato da Microsoft. Si tratta di un'estensione di JavaScript che basa le sue caratteristiche su <b>ECMAScript 6</b>

## 9 Bibliografia

---

### 9.1 Bibliografia per libri

- |                          |                       |
|--------------------------|-----------------------|
| 1. Organizzazione        | [SAM Trevano],        |
| 2. Numero di edizione    | [Prima Edizione],     |
| 3. Nome dell'editore     | [Lorenzo Di Stefano], |
| 4. Anno di pubblicazione | [2023],               |
| 5. ISBN.                 |                       |

### 9.2 Sitografia

1. URL del sito (se troppo lungo solo dominio, evt completo nel diario),
2. Eventuale titolo della pagina (in italico),
3. Data di consultazione (01-12-2023).

## 10 Allegati

---

Elenco degli allegati:

- Diari di lavoro
- Codici sorgente/documentazione
- Documentazione di typescript utilizzata
- Eventuali guide utente / Manuali di utilizzo
- Mandato e/o QdC
- UMLClassDiagram