

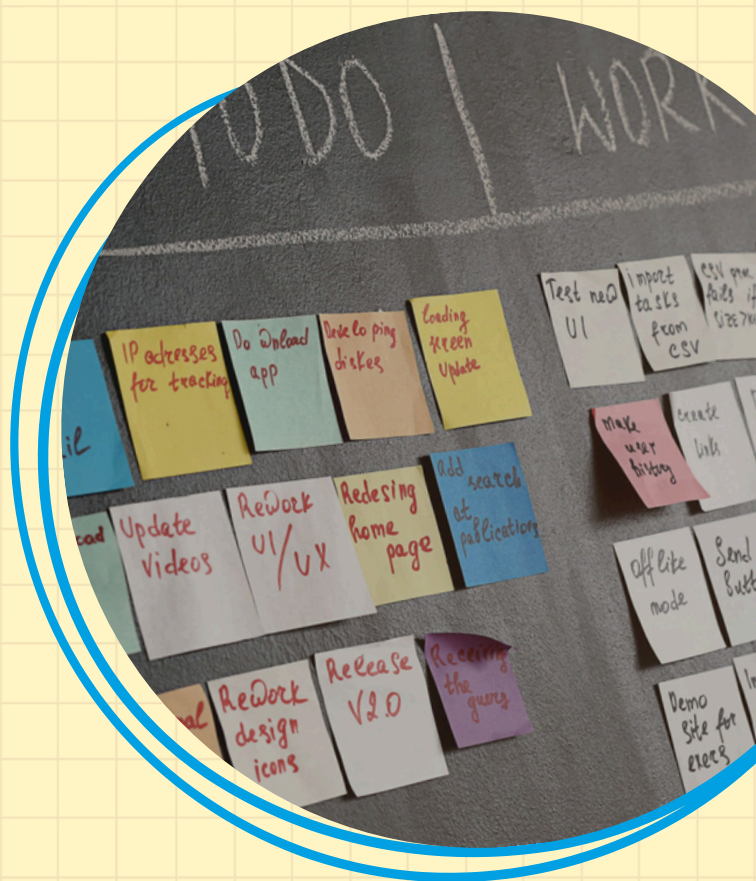


# A GEOMETRIC DRAWING PROGRAM

A.A. 2024/2025

Environment  
Configuration

Docente:  
**Pierluigi Ritrovato**



**A cura del gruppo 5:**

Apicella Antonio 0622702531

Celano Benedetta Pia 0622702558

Cuomo Carmine 0622702688

Guerra Simone 0622702675

# Environment Configuration

## 1. Team organization

### 1.1 Scheduling degli incontri

Nella fase di PreGame sono stati innanzitutto definiti i giorni e gli orari lavorativi così da rendere efficiente l'attività di Scrum:

- Mercoledì: 16:00 - 19:00 (3h)
- Giovedì: 9:00 - 11:00 (2h)
- Venerdì: 9:00 - 11:00 (2h)
- Lunedì: 14:30 - 16:30 (2h)

(Si considera caso eccezionale la prima settimana, per cui è stato necessario spostare le ore previste per il mercoledì al sabato mattina, data la consegna del materiale didattico avvenuta nella serata di mercoledì 7/05).

### 1.2 Piattaforme di collaborazione

- Per la stesura collaborativa dei documenti e dei grafici durante lo sprint: **GoogleDoc & GoogleSheets**.
- Per le videochiamate di daily scrum ove risulti difficile la comunicazione in presenza: **Discord**.
- Per la modellazione dei grafici che guidino l'implementazione durante lo sprint (relativamente al detailed design): **Draw.io**



## 2. Tools Setup

### 2.1 Trello

**Link Trello:** [Trello - A Geometric Drawing Program\\_Gruppo05-AH](#)

Su Trello le liste che si è deciso di inserire sono le seguenti:

- **Product Backlog** → gestita dal PO il quale carica le user stories con il livello di priorità. Successivamente il team aggiungerà le stime mediante story points.
- **Sprint Backlog** → gestita dal team, il quale sposta soltanto le user stories per la sprint corrente. Durante la sprint dal team vengono identificati i tasks associati a ciascuna user story.
- **In Progress** → si trovano solo le user stories in corso di svolgimento da parte dei membri del team, i quali abilitano la checkbox al completamento di ogni task associato alla user story di cui sono incaricati.
- **To Test** → le user stories vengono spostate in questa sezione quando deve essere effettuato unit ed integration testing.
- **Done** → nel momento in cui risultano completate e rispettano la definition of done, le user stories vengono spostate in questa sezione.

### 2.2 Git

- **Nome repository:** A-Geometric-drawing-program\_Gruppo05-AH
- **Link:** [https://github.com/apiantonio/A-Geometric-drawing-program\\_Gruppo05-AH](https://github.com/apiantonio/A-Geometric-drawing-program_Gruppo05-AH)

### 2.3 Piattaforma di sviluppo

Per l'implementazione il team ha deciso di utilizzare come IDE IntelliJ.

Le versioni che verranno utilizzate del Java JDK e di JavaFX saranno:

- Java JDK 23
- JavaFX 24

### 2.4 Coding Conventions

#### Nomi di classi, metodi, variabili e costanti

Si seguono le convenzioni Java standard [Code Conventions for the Java Programming Language](#):

- **Pacchetti:** tutto minuscolo.

- **Classi/Interfacce:** `PascalCase` (prima lettera maiuscola, restanti parole con maiuscola), i nomi devono essere autoesplicativi.
- **Metodi e variabili:** `lowerCamelCase` (prima lettera minuscola, verbo o azione). Usare preferibilmente verbi con parole interne maiuscole per i metodi, per le variabili evitare nomi costituiti da un unico carattere.
- **Costanti (static final):** `UPPER_SNAKE_CASE` e dunque tutte lettere maiuscole con le parole separate da underscore.

## Spaziatura, indentazione e lunghezza righe

Al fine di migliorare la leggibilità sono state definite le seguenti convenzioni.

- **Indentazione:** usare 4 spazi per livello di indentazione, evitando di mixare tab e spazi.
- **Lunghezza righe:** Se una riga è troppo lunga, è opportuno spezzarla dopo una virgola.
- **Linee bianche:** separare con una riga vuota blocchi di codice particolarmente complessi e metodi.
- **Spazi:** inserire uno spazio dopo ogni virgola e attorno agli operatori binari ( `+ - * / = ==` ecc.) al fine di ottenere maggiore chiarezza.

## Commenti e documentazione

- **Intestazione file:** all'inizio di ogni file è preferibile includere un commento in stile C con autore, data, versione, scopo (opzionale), un esempio è presentato di seguito:

```
/*
 * Autore: Gruppo05
 * Data: 10/05/2025
 * Scopo: Gestisce l'accesso degli utenti all'app.
 */
```

- **Commenti inline:** Per commenti brevi usare `//`, si evitino commenti inutili. Per blocchi di codice più complessi si usi il seguente formato `/* ... */` all'interno del metodo.