Scuola universitaria professionale della Svizzera italiana
Dipartimento tecnologie innovative

SUPSI

2D Editor for PNM files

Studente/i	Relatore	Correlatore
Dominik Panzarella Mattia Mancino Alessandro Spagnuolo	Corti Giancarlo	Brocco Amos

Corso di laurea	Modulo / Codice Progetto	Anno
Ingegneria Informatica	Ing. Software II	2024-2025

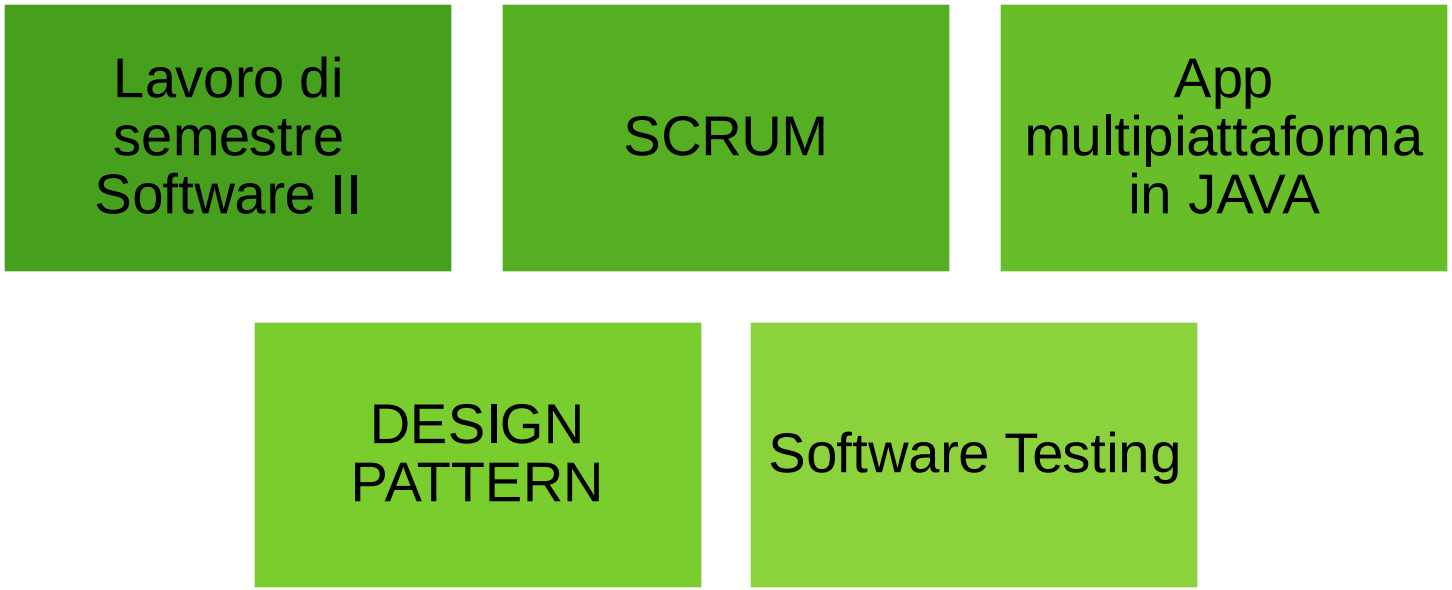
Committente	Data
Corti Giancarlo Brocco Amos	16.12.2024

STUDENTSUPSI

Indice

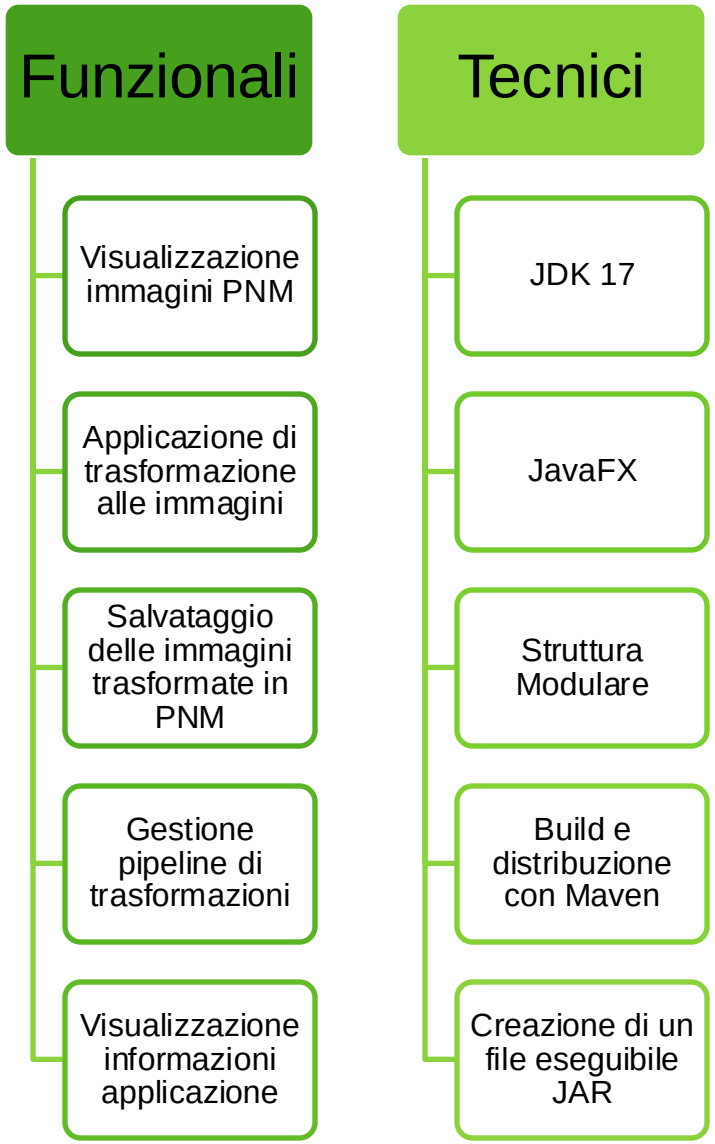
Contesto e motivazioni
Requisiti
Stato dell’arte
Approccio
Demo
Conclusione

Contesto e Motivazioni



STUDENTSUPSI

Requisiti



STUDENTSUPSI

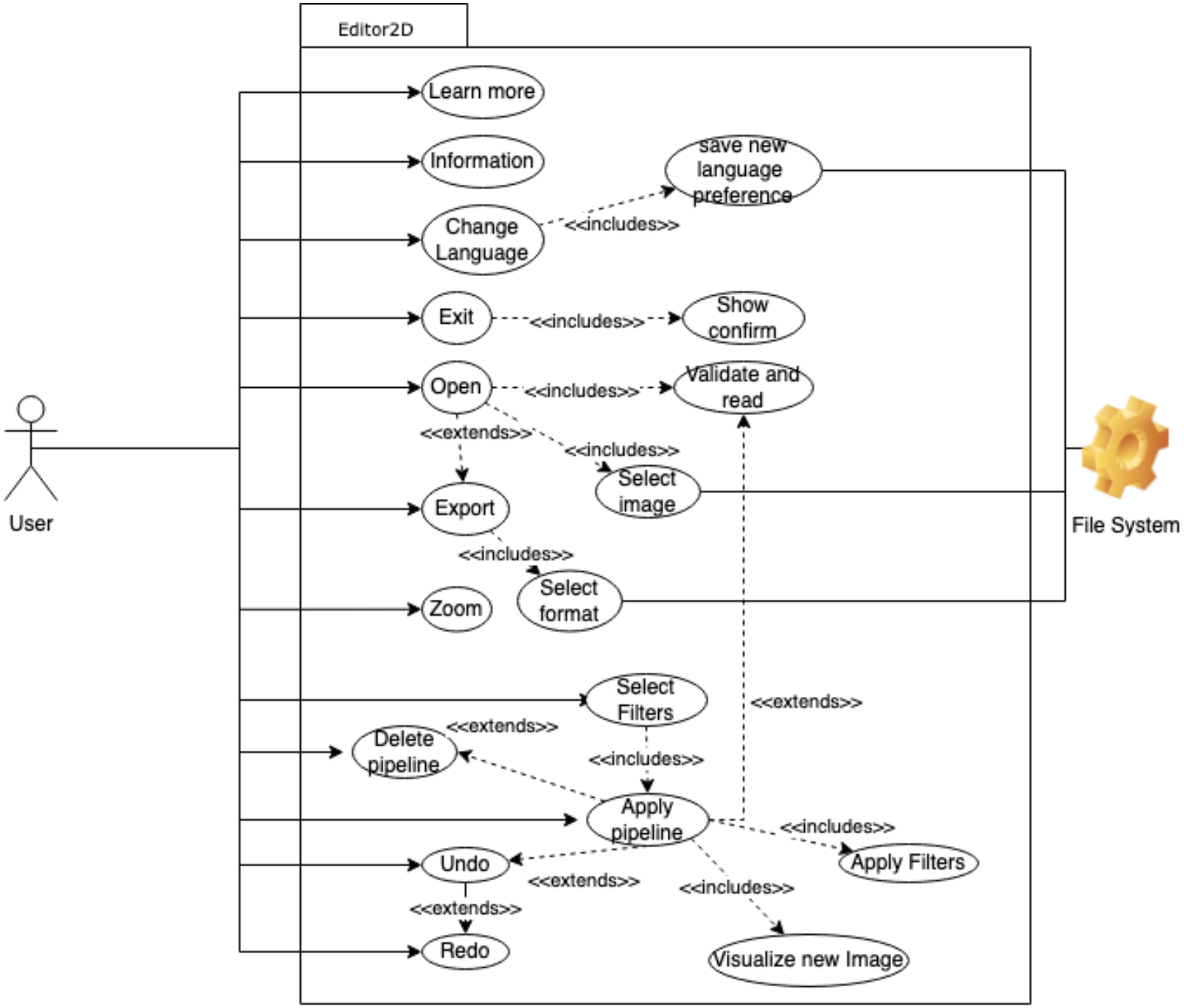
Stato dell’arte

- XnView
- ImageMagick (solo CMD)
- …

STUDENTSUPSI

Approccio

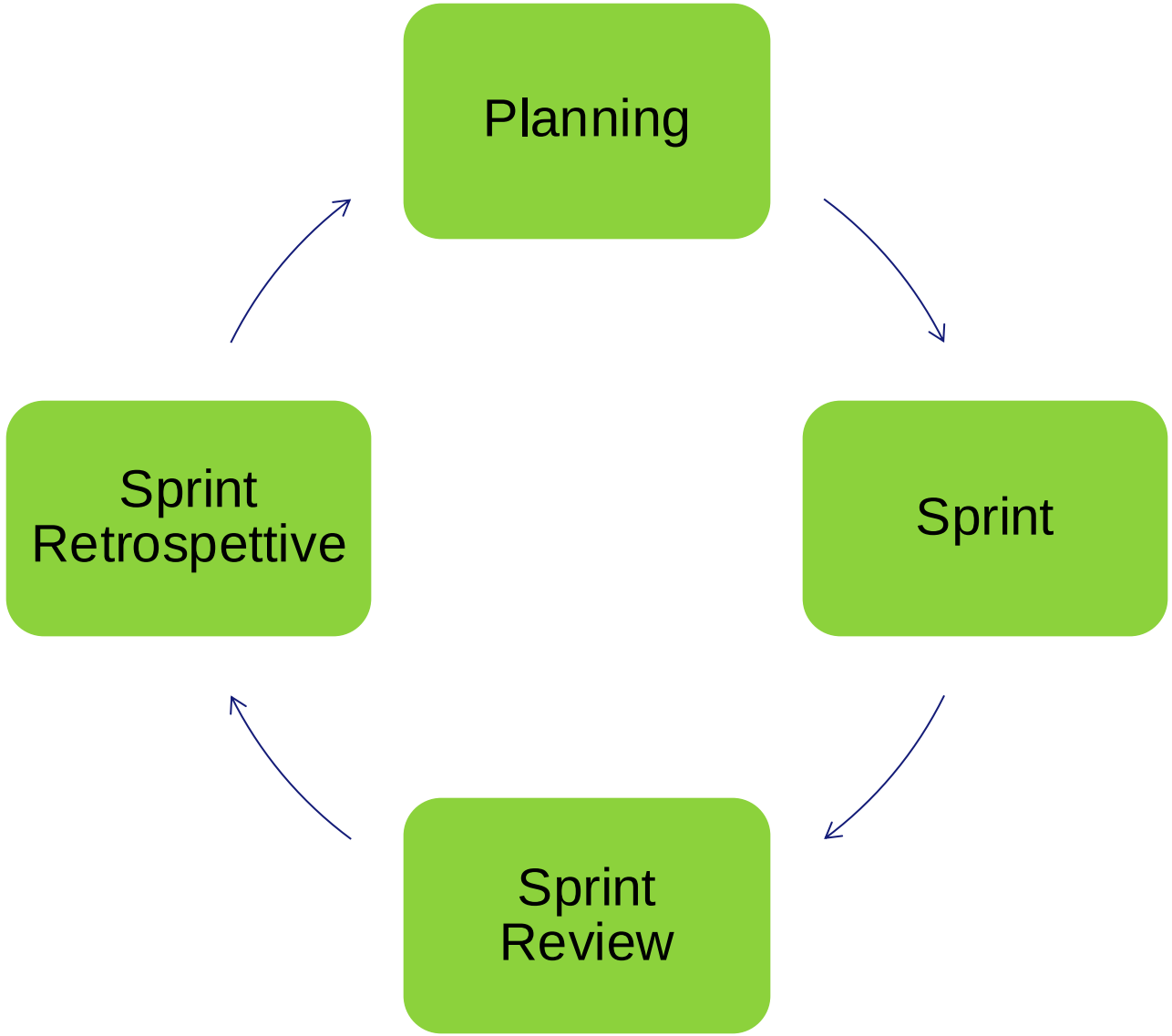
Use Case Diagram



STUDENTSUPSI

Approccio

Metodologia SCRUM



STUDENTSUPSI

Approccio

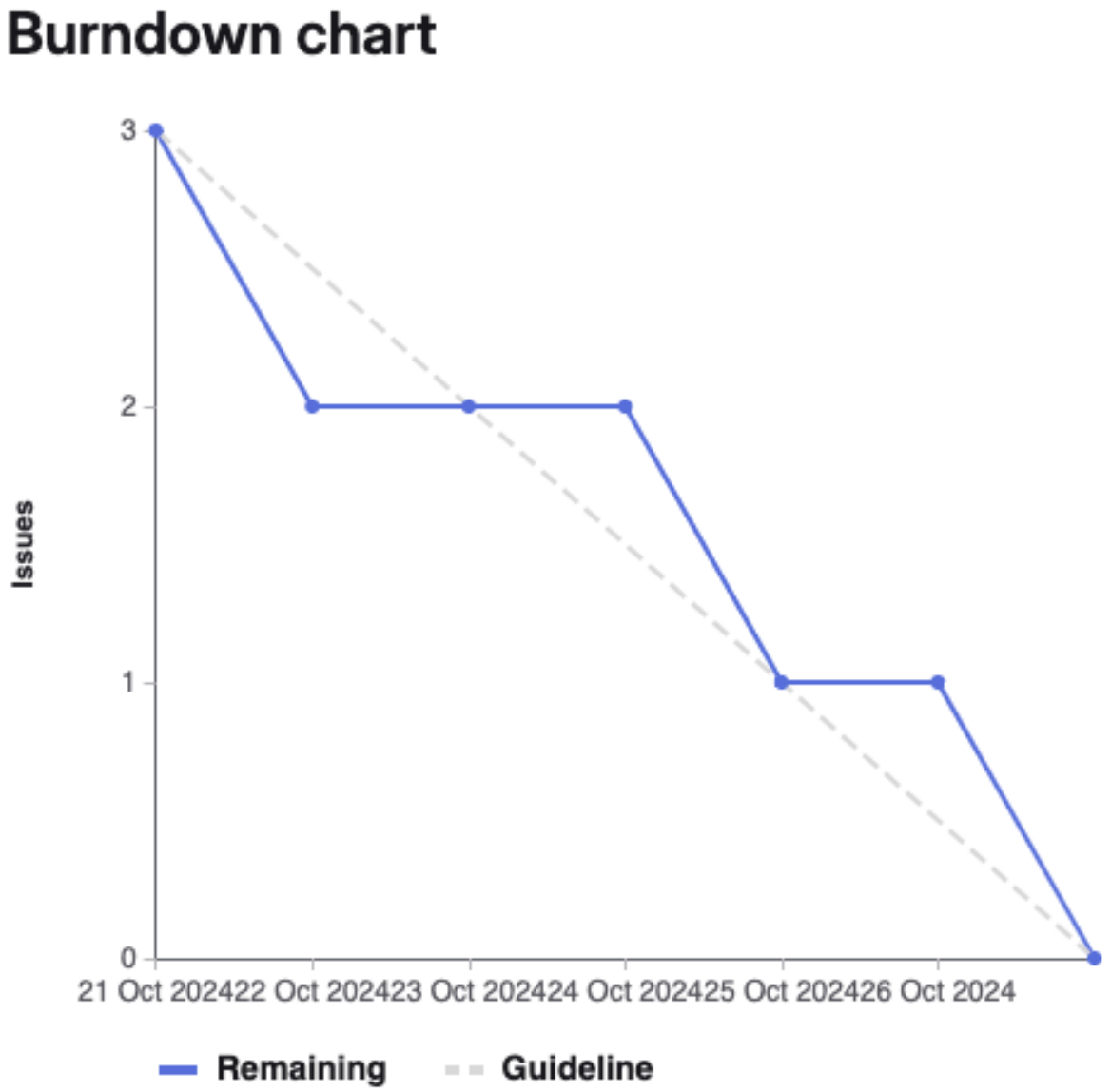
Sprint

- Sprint di una settimana
- Average Team Velocity: 76 Punti

ITERATION-01	DTI-ISIN / labingsw / labingsw02 / 20242025 / group08	Oct 7, 2024–Oct 13, 2024	Closed	7/7 complete 100%
ITERATION-02	DTI-ISIN / labingsw / labingsw02 / 20242025 / group08	Oct 14, 2024–Oct 20, 2024	Closed	9/9 complete 100%
ITERATION-03	DTI-ISIN / labingsw / labingsw02 / 20242025 / group08	Oct 21, 2024–Oct 27, 2024	Closed	6/6 complete 100%
ITERATION-04	DTI-ISIN / labingsw / labingsw02 / 20242025 / group08	Oct 28, 2024–Nov 3, 2024	Closed	2/2 complete 100%
ITERATION-05	DTI-ISIN / labingsw / labingsw02 / 20242025 / group08	Nov 4, 2024–Nov 10, 2024	Closed	2/2 complete 100%
ITERATION-06	DTI-ISIN / labingsw / labingsw02 / 20242025 / group08	Nov 11, 2024–Nov 17, 2024	Closed	0/0 complete 0%
ITERATION-07	DTI-ISIN / labingsw / labingsw02 / 20242025 / group08	Nov 18, 2024–Nov 24, 2024	Closed	6/6 complete 100%
ITERATION-08	DTI-ISIN / labingsw / labingsw02 / 20242025 / group08	Nov 25, 2024–Dec 1, 2024	Closed	3/3 complete 100%
ITERATION-09	DTI-ISIN / labingsw / labingsw02 / 20242025 / group08	Dec 2, 2024–Dec 8, 2024	Closed	2/2 complete 100%

Approccio

Sprint

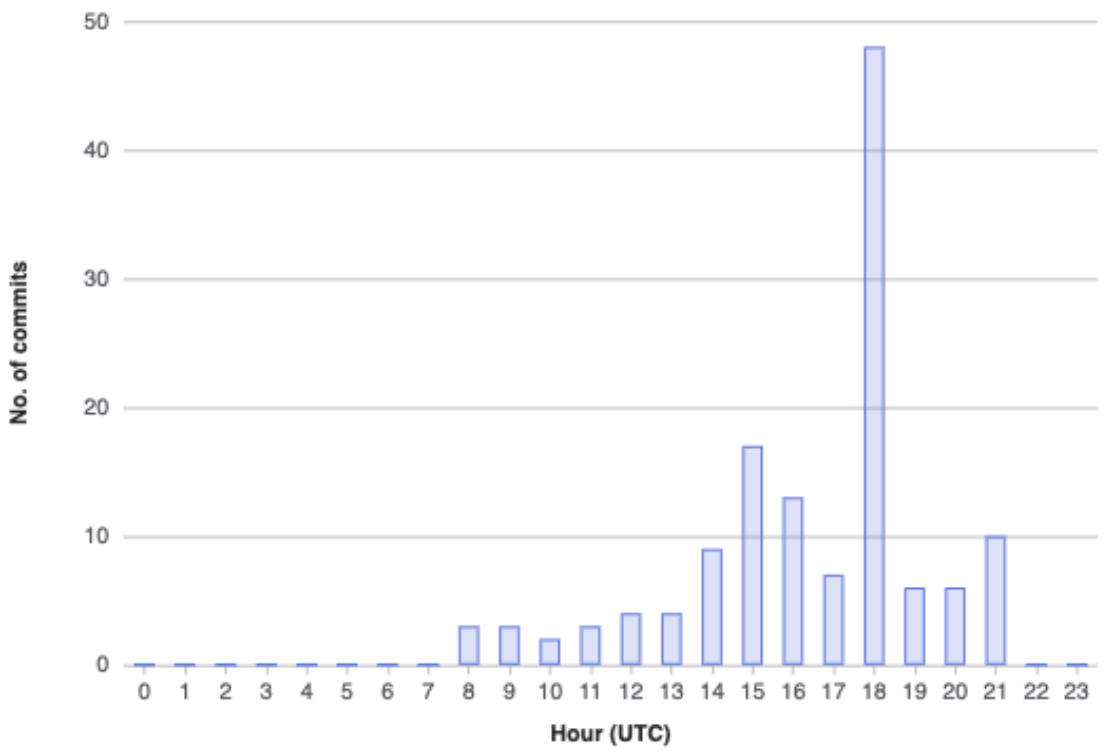


STUDENTSUPSI

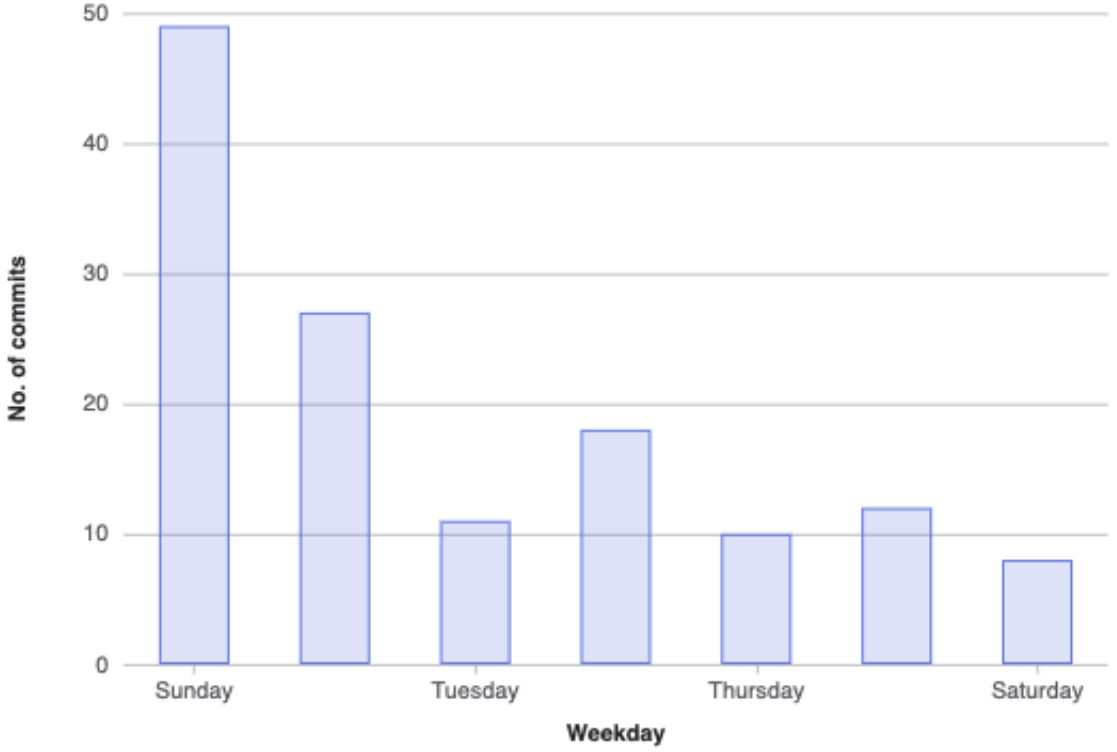
Approccio

Sprint

Commits per day hour (UTC)



Commits per weekday



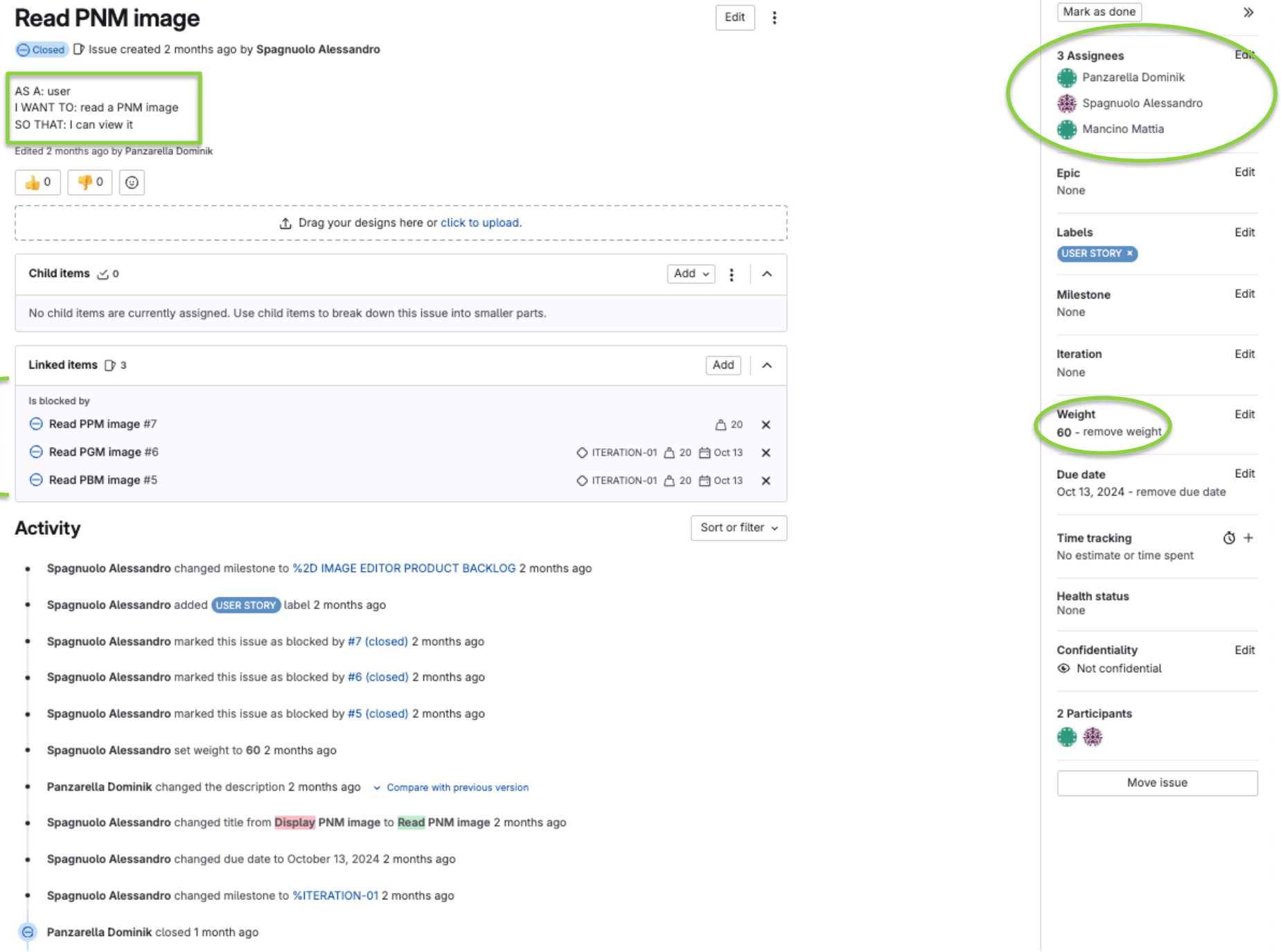
STUDENTSUPSI

SUPSI

Approccio

Esempio User Story

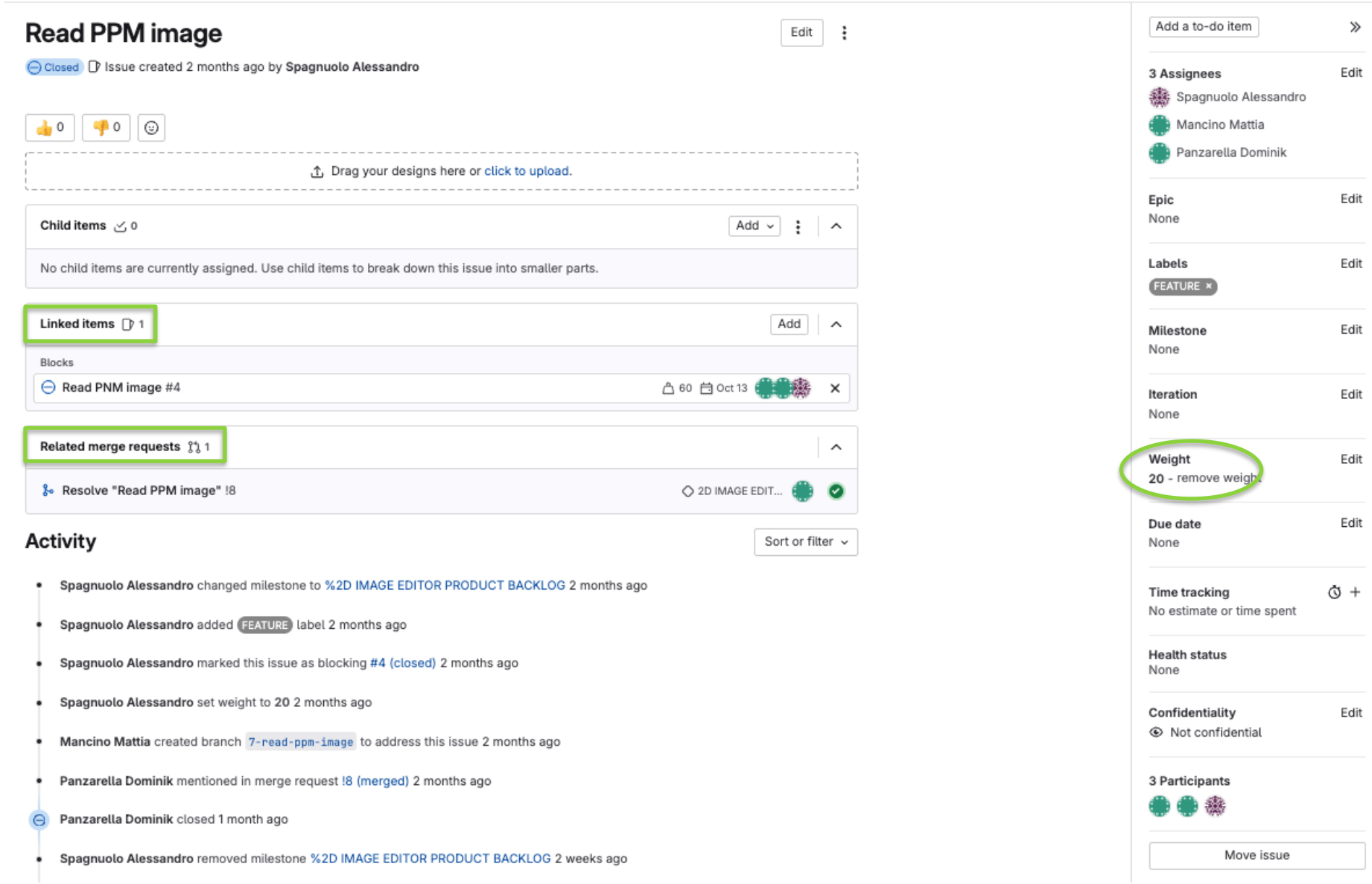
Features



STUDENTSUPSI

Approccio

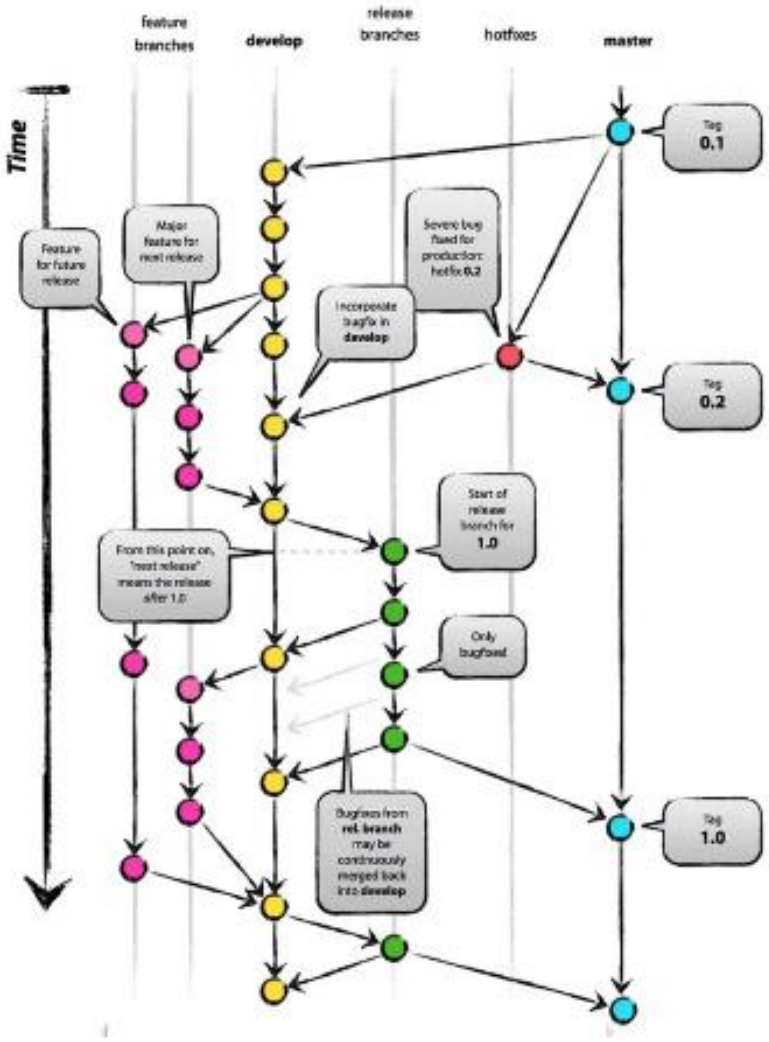
Esempio Feature

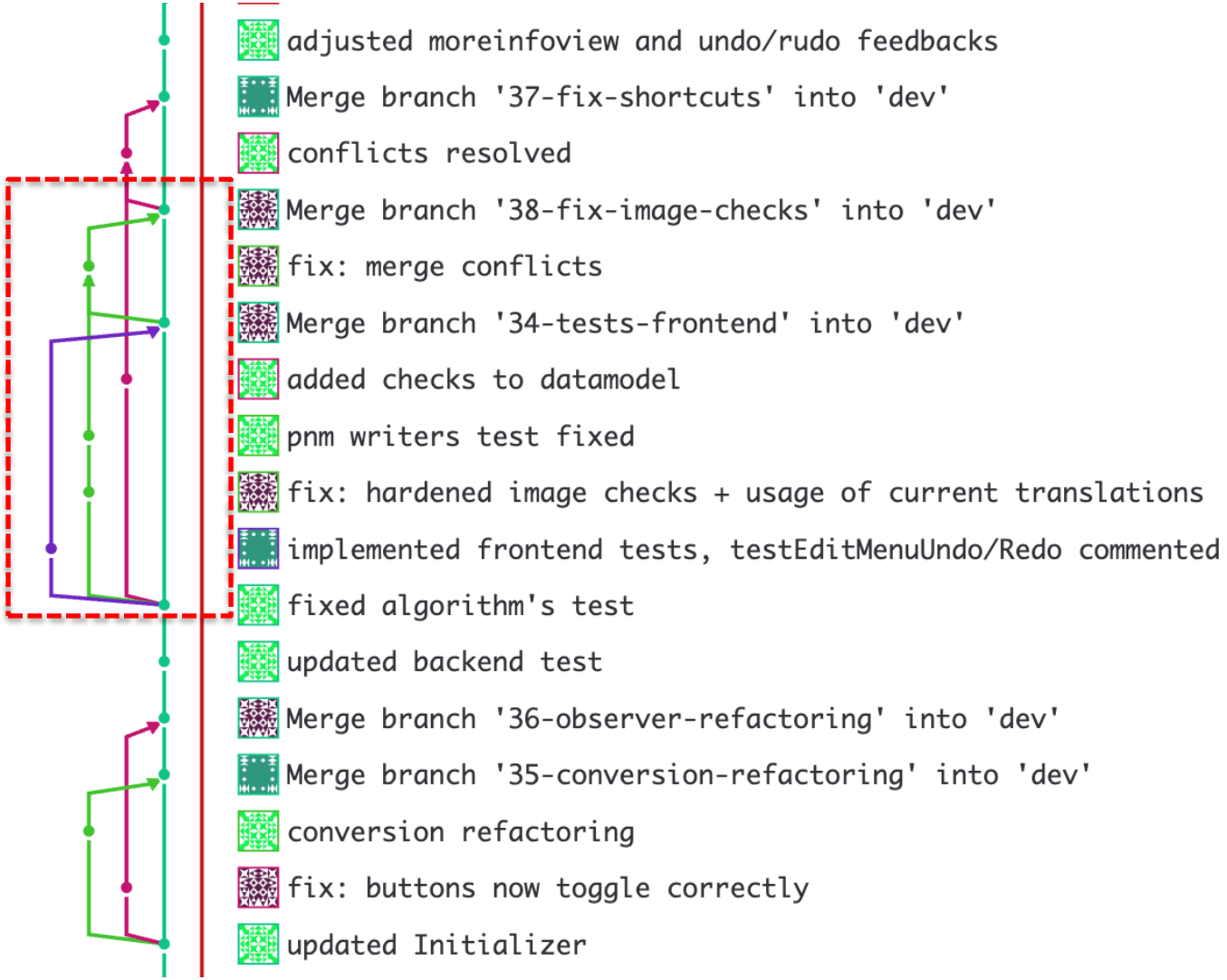


STUDENTSUPSI

Approccio

Repository Graph

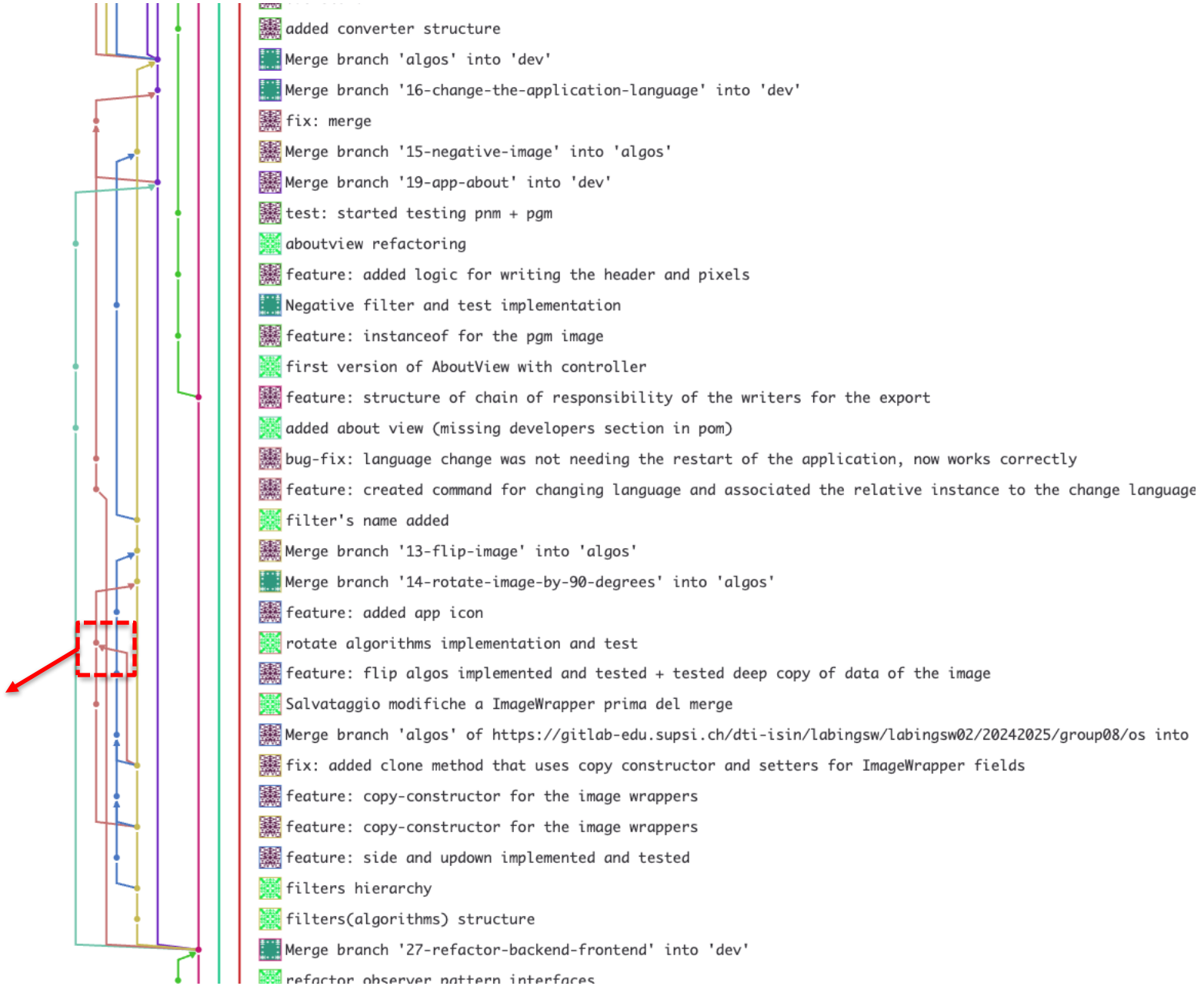




STUDENTSUPSI

Approccio

Repository Graph



Conflict solved

STUDENTSUPSI

Approccio

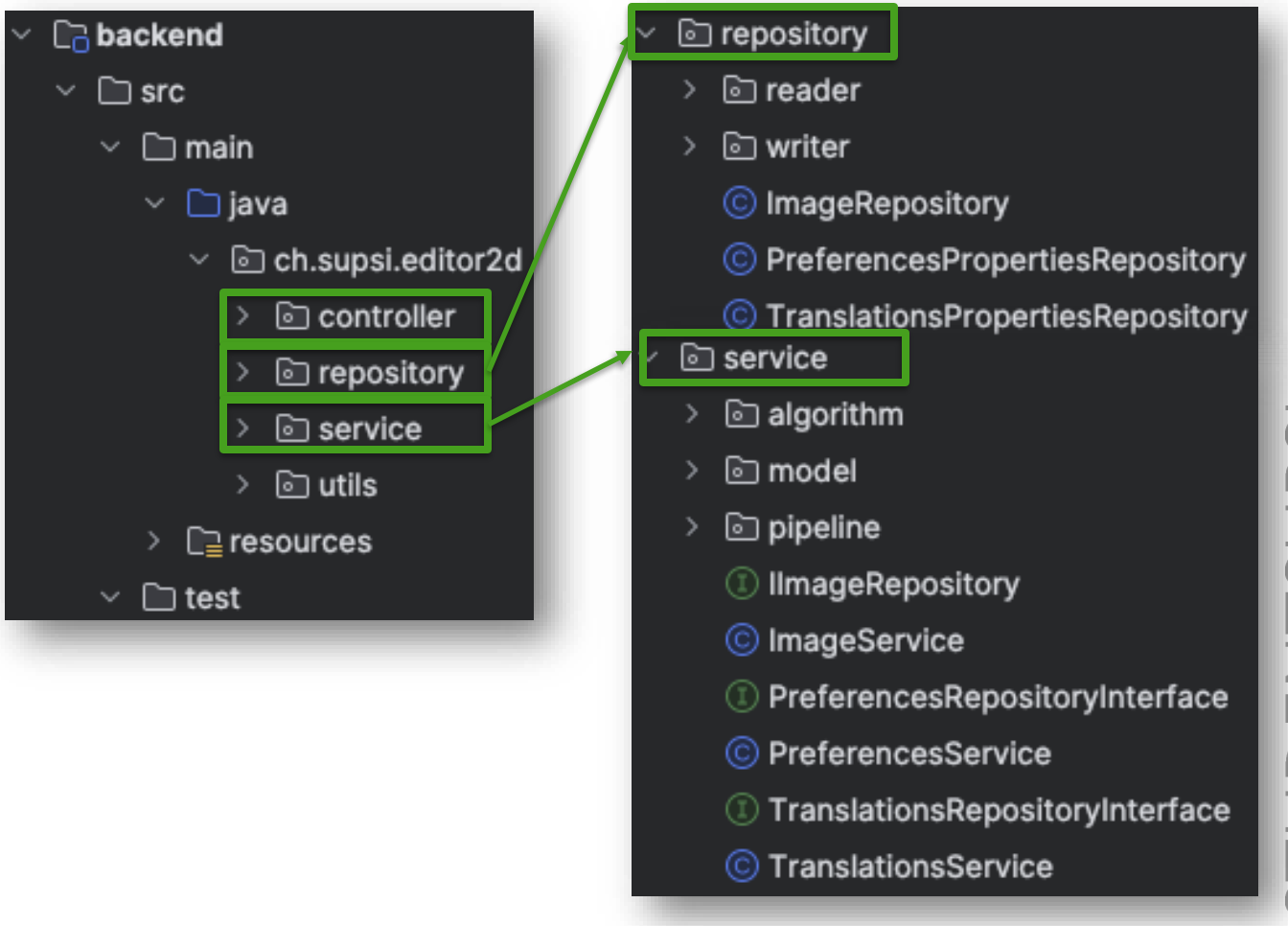
Backend - Struttura

- **Backend** -> Multi-Layer

  - Import/Export immagini
  - Internazionalizzazione
  - Pipeline di esecuzione dei filtri

- **Dipendenze**

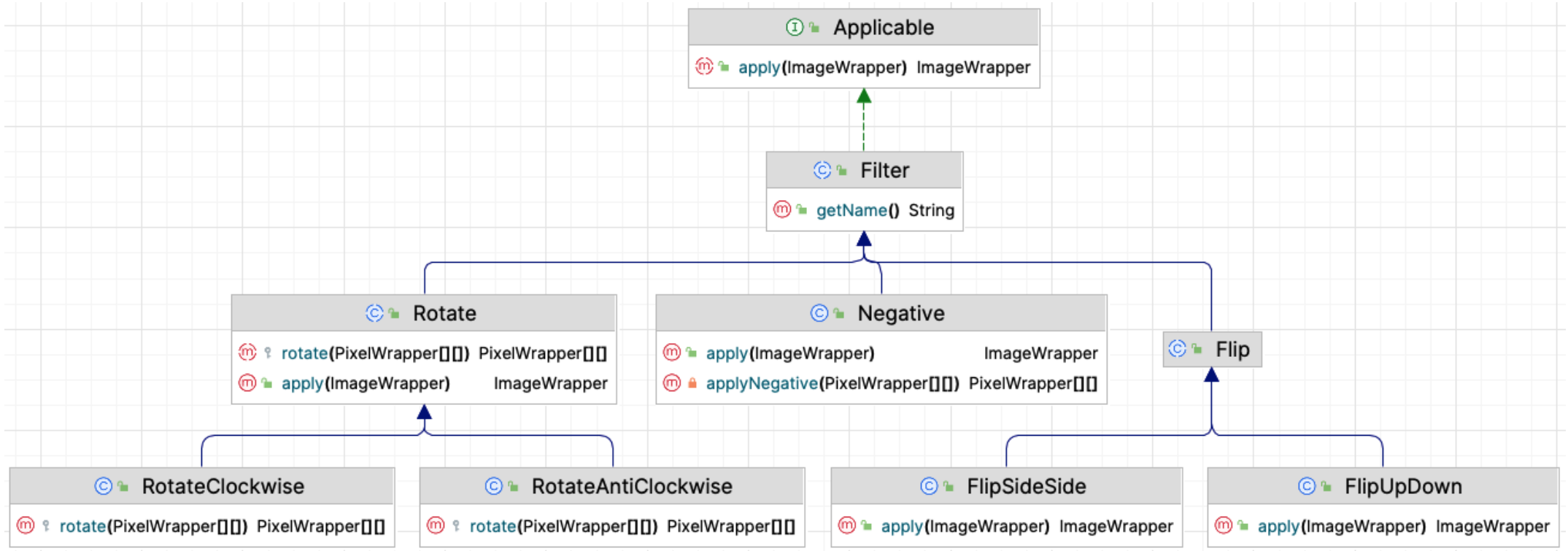
  - JUnit



STUDENTSUPSI

UML – Class diagram

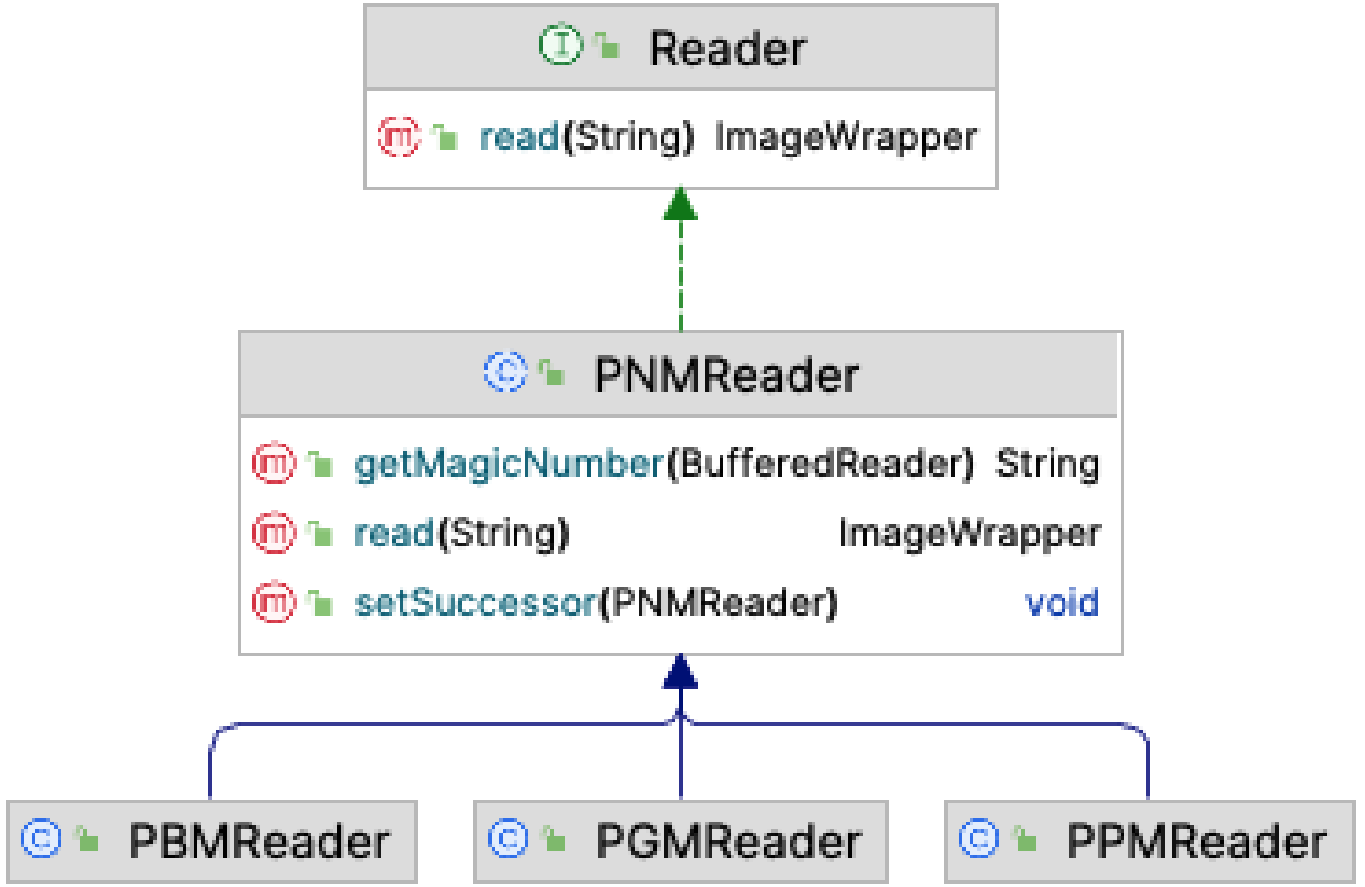
Filtri – Strategy Pattern



STUDENTSUPSI

UML – Class diagram

Reader* – Strategy + Chain of Responsibility + Template Method

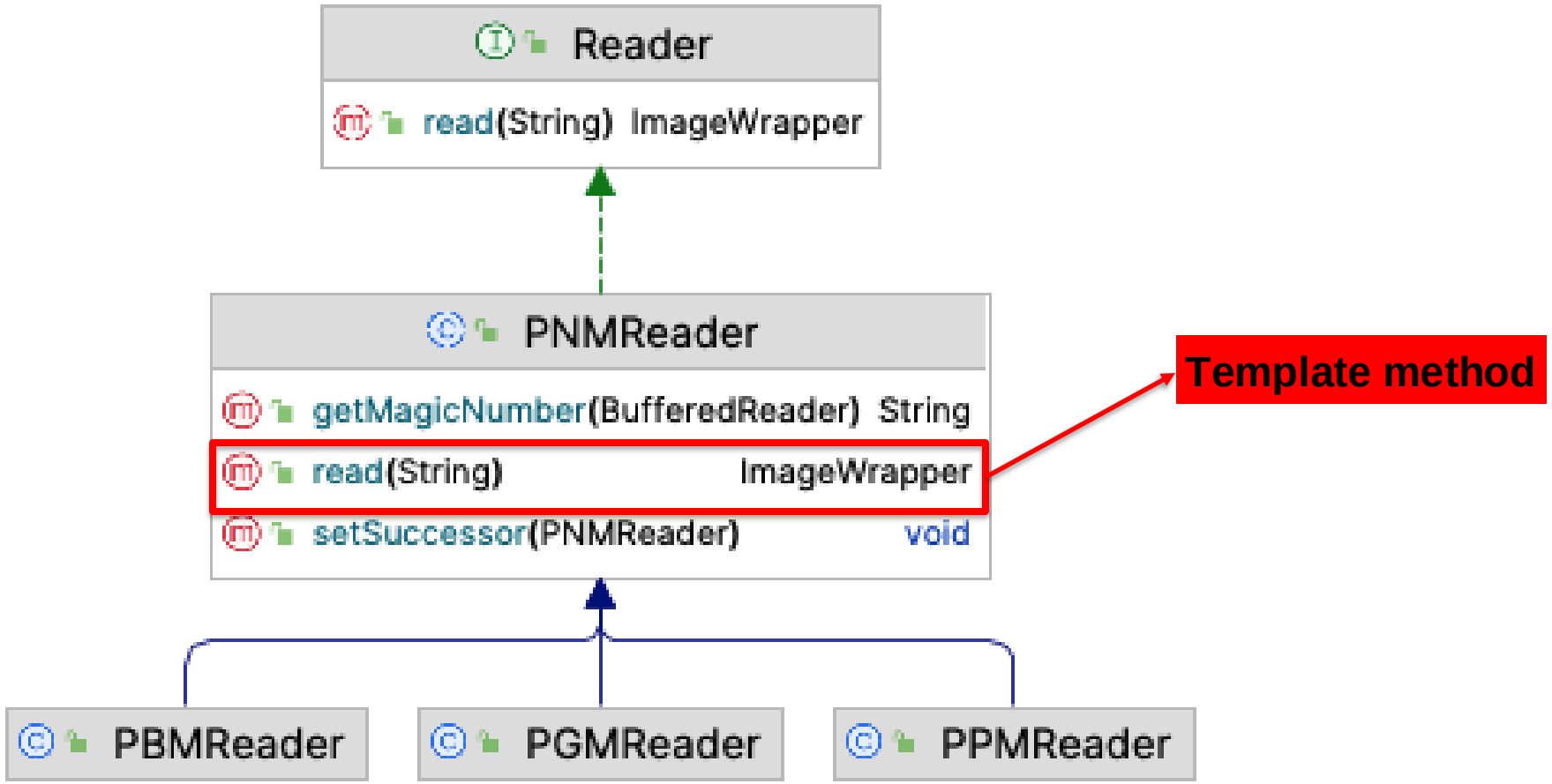


* UML in questo caso non presenta tutti i metodo all'interno delle classi

STUDENTSUPSI

UML – Class diagram

Reader* – Strategy + Chain of Responsibility + Template Method



* UML in questo caso non presenta tutti i metodo all'interno delle classi

STUDENTSUPSI

Approccio

Reader* – Strategy + Chain of Responsibility + Template Method

```java
public ImageWrapper read(final String path) throws FileReadingException, IOException {
    try(BufferedReader reader = new BufferedReader(new FileReader(path))){
        final String magicNumber = getMagicNumber(reader);
        if(canHandle(magicNumber)){
            return helperReader(reader,magicNumber);
        }
        else if(successor!=null){
            return successor.read(path);
        }else{
            throw new FileReadingException(translationsController.translate( key: "label.cantHandle"));
        }
    }
}
```

Template method

* UML in questo caso non presenta tutti i metodo all'interno delle classi

STUDENTSUPSI

Approccio

Reader* – Strategy + Chain of Responsibility + Template Method

```
public ImageWrapper read(final String path) throws FileReadingException, IOException {
    try(BufferedReader reader = new BufferedReader(new FileReader(path))){
        final String magicNumber = getMagicNumber(reader);
        if(canHandle(magicNumber)){
            return helperReader(reader,magicNumber);
        }
        else if(successor!=null){
            return successor.read(path);
        }else{
            throw new FileReadingException(translationsController.translate( key: "label.cantHandle"));
        }
    }
}
```

Template method

La richiesta viene propagata lungo la catena

* UML in questo caso non presenta tutti i metodo all'interno delle classi

STUDENTSUPSI

Approccio

Reader* – Strategy + Chain of Responsibility + Template Method

```java
public ImageWrapper read(final String path) throws FileReadingException, IOException {
    try(BufferedReader reader = new BufferedReader(new FileReader(path))){
        final String magicNumber = getMagicNumber(reader);
        if(canHandle(magicNumber)){
            return helperReader(reader,magicNumber);
        }
        else if(successor!=null){
            return successor.read(path);
        }else{
            throw new FileReadingException(translationsController.translate( key: "label.cantHandle"));
        }
    }
}
```

Template method

```java
protected abstract boolean canHandle(final String magicNumber);
```

* UML in questo caso non presenta tutti i metodo all'interno delle classi

STUDENTSUPSI

Approccio

Reader* – Strategy + Chain of Responsibility + Template Method

```
public ImageWrapper read(final String path) throws FileReadingException, IOException {
    try(BufferedReader reader = new BufferedReader(new FileReader(path))){
        final String magicNumber = getMagicNumber(reader);
        if(canHandle(magicNumber)){
            return helperReader(reader,magicNumber);
        }
        else if(successor!=null){
            return successor.read(path);
        }else{
            throw new FileReadingException(translationsController.translate( key: "label.cantHandle"));
        }
    }
}
```

Template method

```
private ImageWrapper helperReader(BufferedReader reader, final String magicNumber) throws IOException {
    String[] header = getHeader(reader);
    PixelWrapper[][] pixels = getPixels(reader,header);
    return createImageWrapper(header, pixels);
}
```

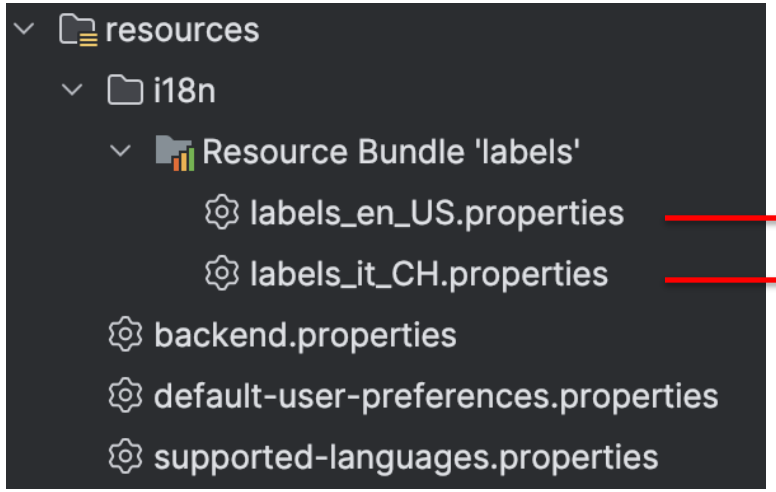
```
protected abstract ImageWrapper createImageWrapper(final String[] header, PixelWrapper[][] pixels);
```

* UML in questo caso non presenta tutti i metodo all'interno delle classi

STUDENTSUPSI

Internationalization

Labels




```
String currentLanguage = preferencesModel.getCurrentLanguage();
this.translationsModel.changeLanguage(currentLanguage);
```

TranslationController

```
label.pipelineCleared=The pipeline has been emptied
```

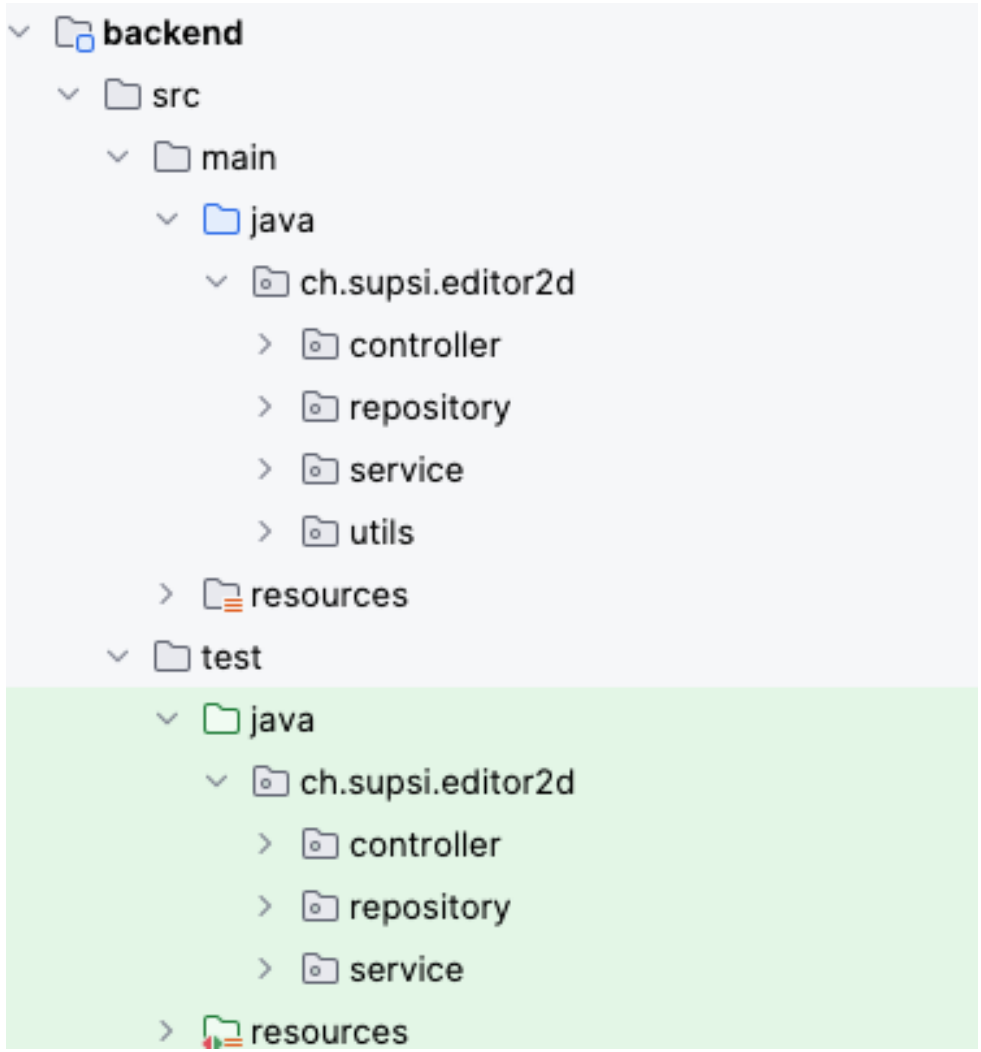
```
label.pipelineCleared=La pipeline è stata svuotata
```

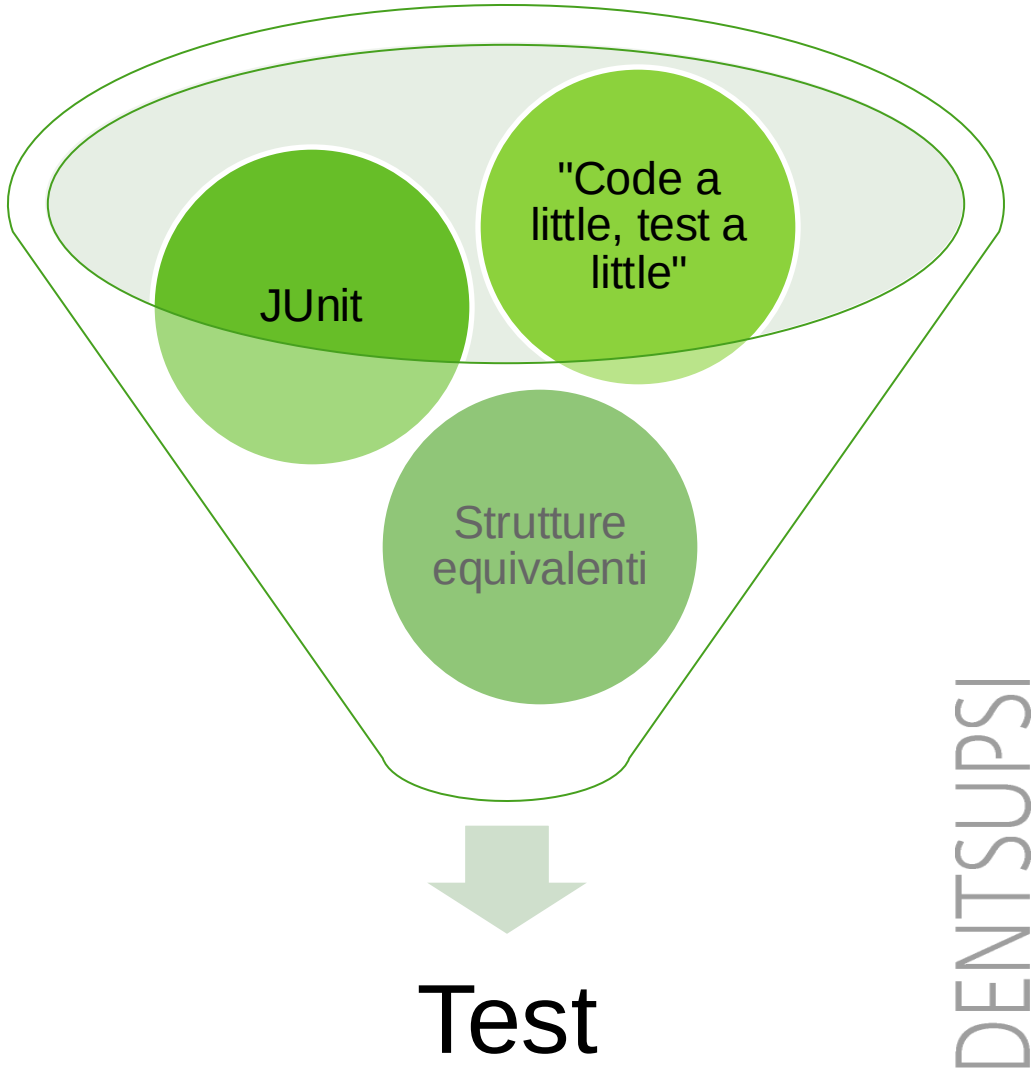
```
notifyFeedbackObservers(translationsController.translate( key: "label.pipelineCleared"));
return text.getText().contains(translationsController.translate( key: "label.pipelineCleared"));
```

STUDENTSUPSI

Test

Backend



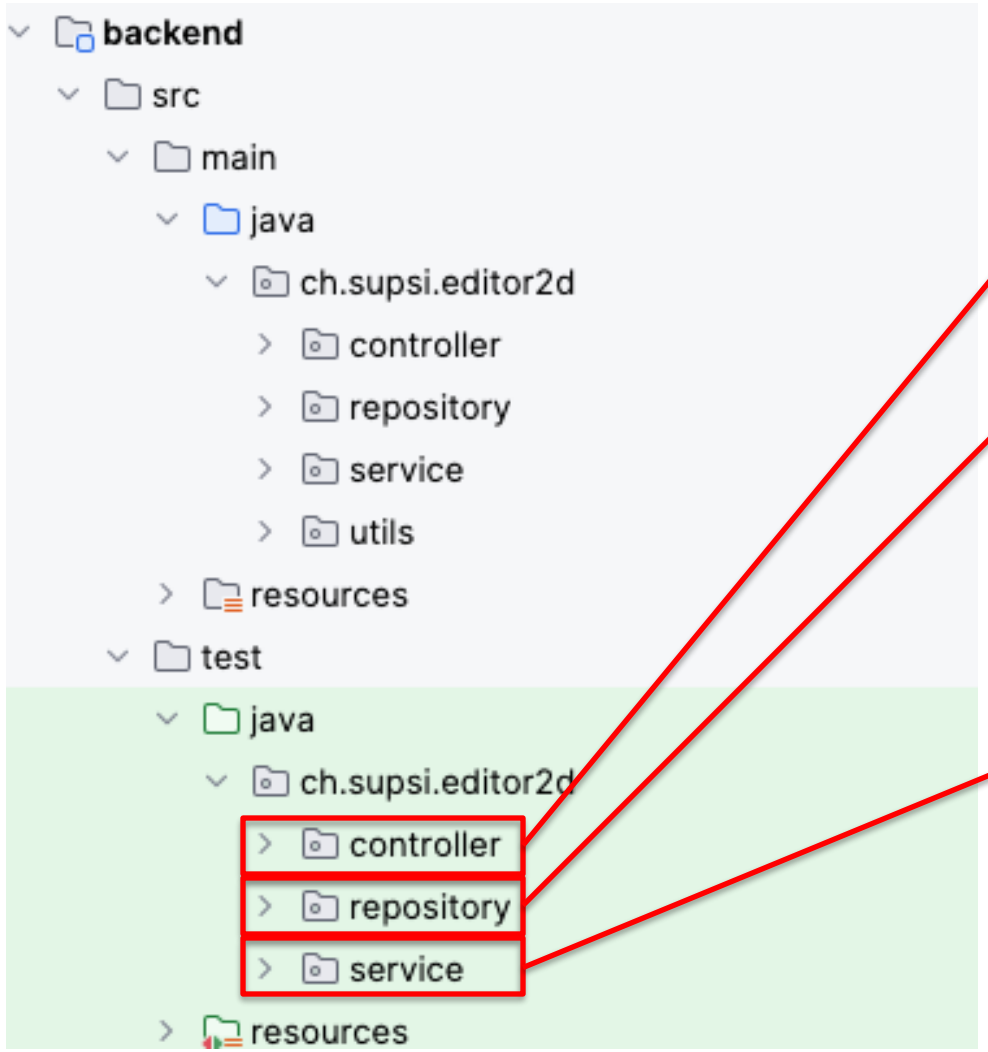


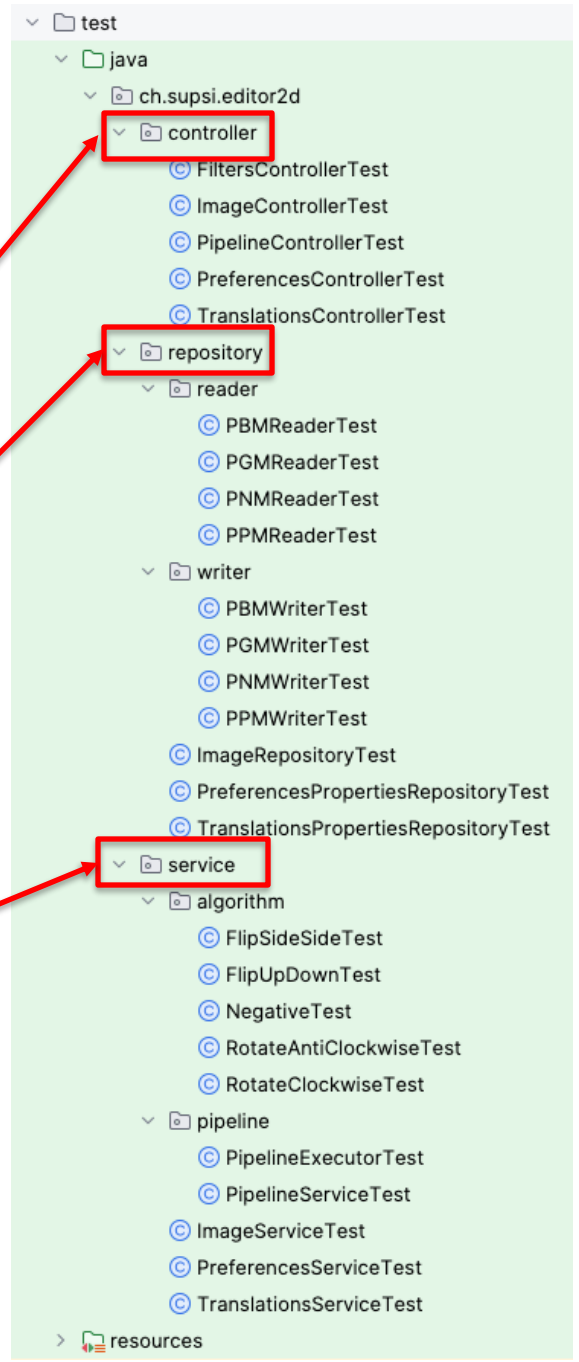
STUDENTSUPSI

SUPSI

Test

Backend





STUDENTSUPSI

Approccio

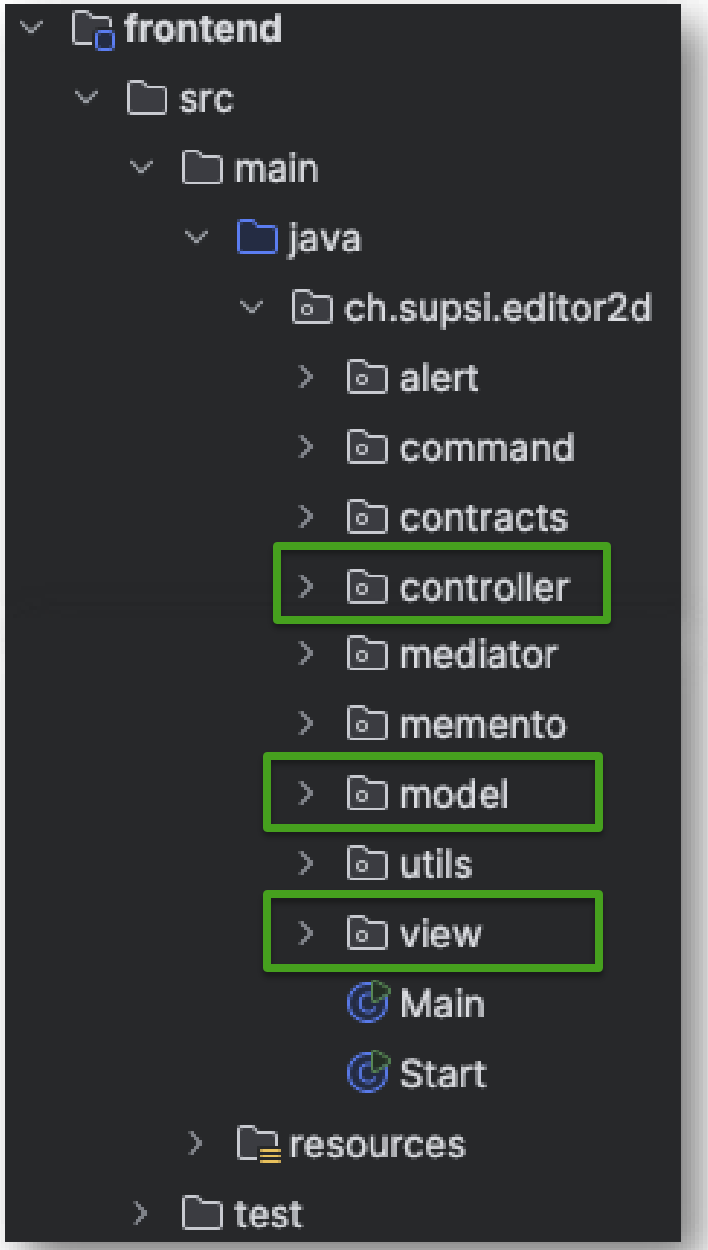
Frontend - Struttura

- **Frontend** -> MVC

  - Interfaccia grafica
  - Apertura/Salvataggio immagine
  - Editing immagine
  - Help & About

- **Dipendenze**

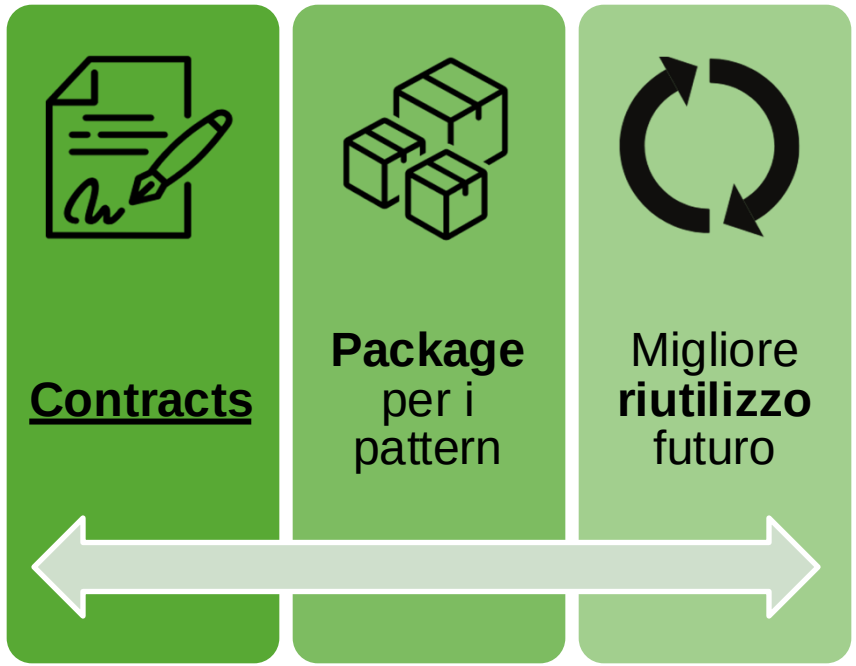
  - Backend
  - JavaFX
  - Junit
  - TestFX
  - Mockito

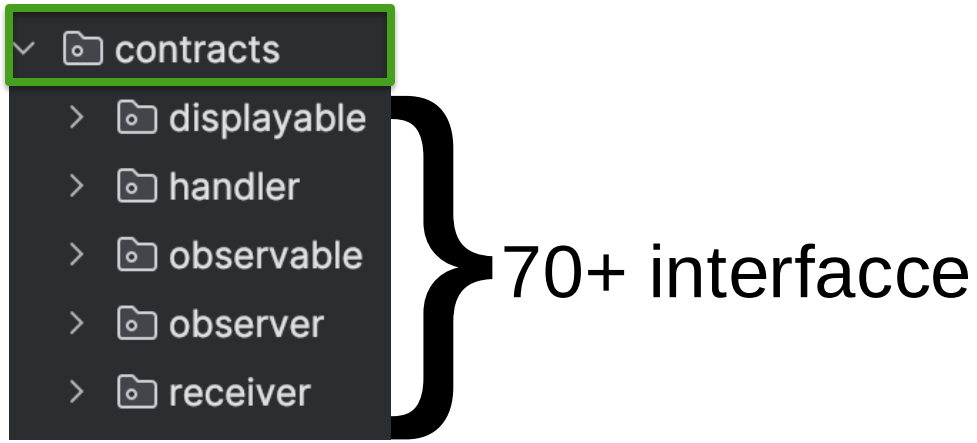


STUDENTSUPSI

Approccio

Coesione + Coupling ridotto

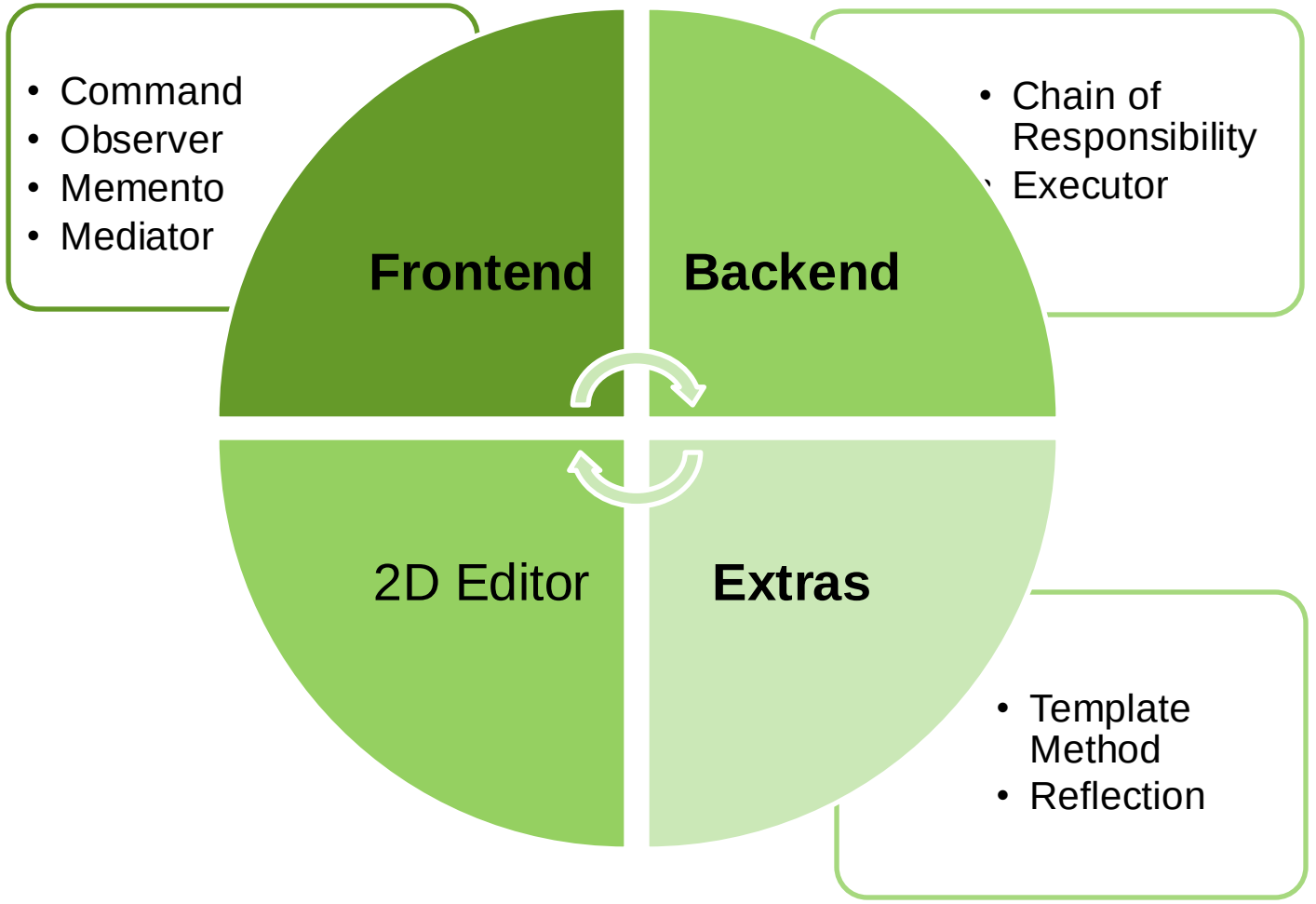




STUDENTSUPSI

Approccio

Design Patterns

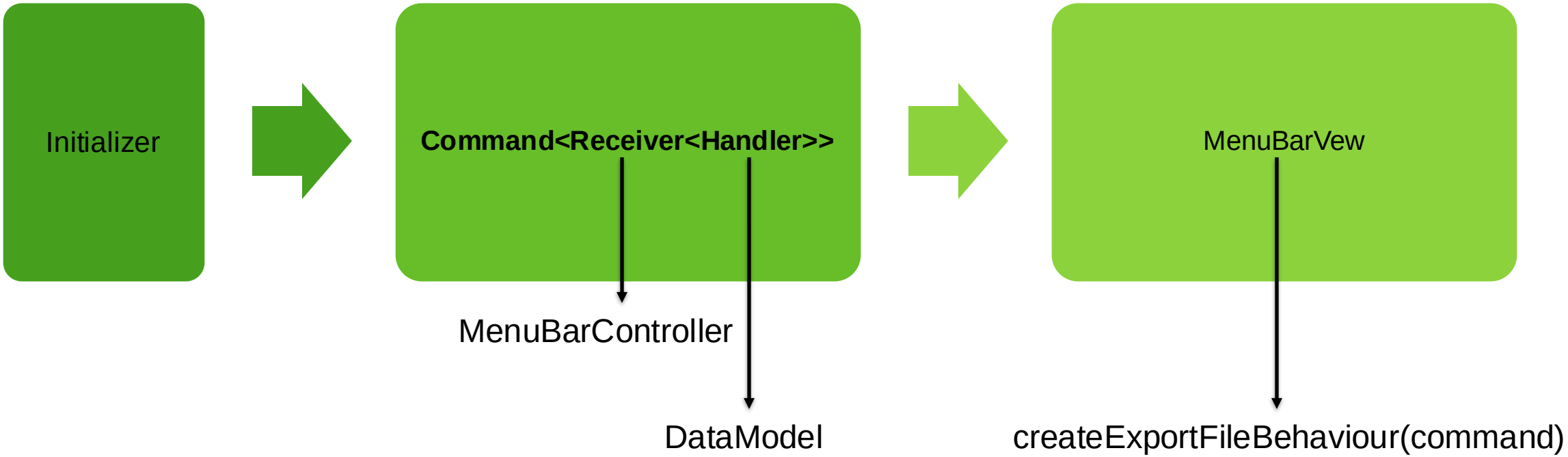


STUDENTSUPSI

Approccio

Export Immagine

- Come viene associata **l'azione dell'export dell'immagine?**

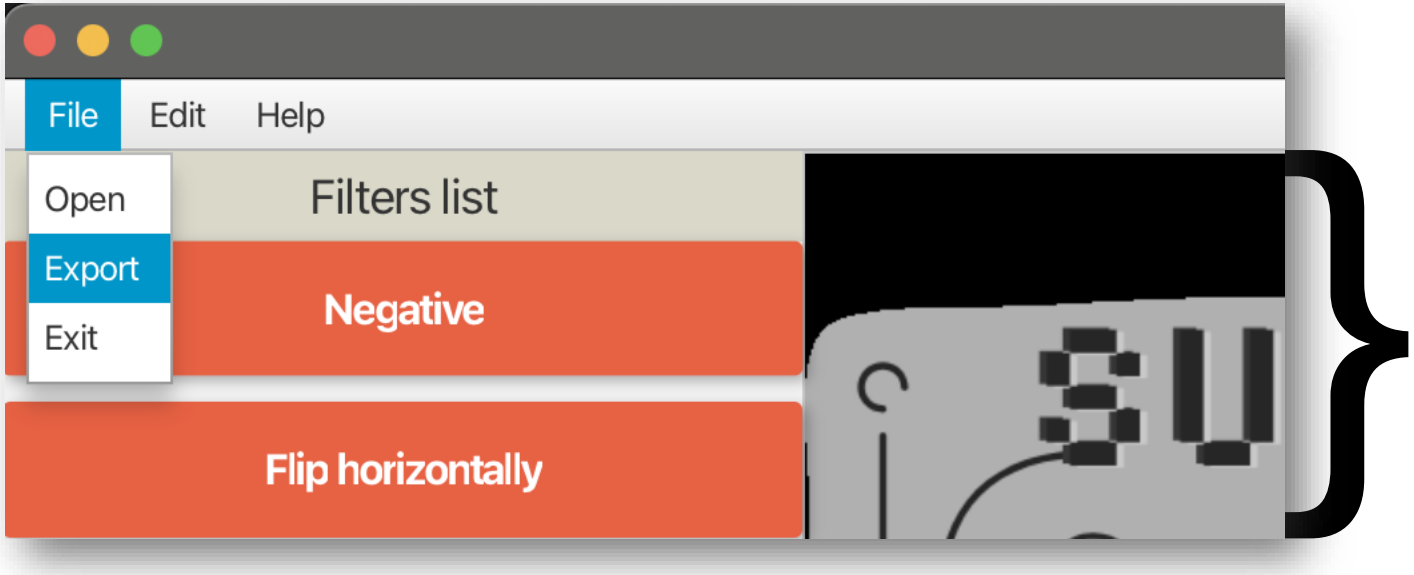


STUDENTSUPSI

Approccio

Export Immagine

- **Come avviene l'export?**



- **PPM**
- **PGM**
- **PBM**

- Esempio **Parte 1: UML sequence diagram** per l'export in **PGM**

STUDENTSUPSI

Frontend

Backend

USER

MenuBarController

DataModel

ImageController

Export

exportModel.exportFile(Filechooser)

filechooser.showSaveDialog()

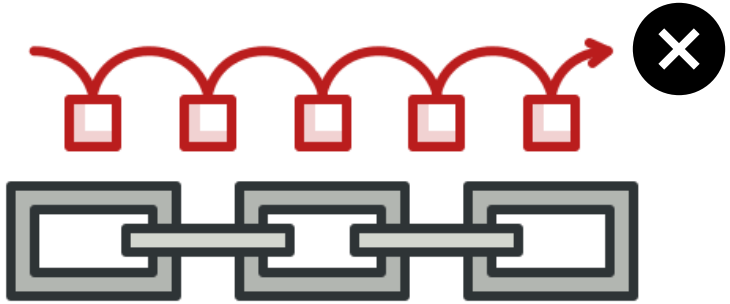
Save

imageController.exportImage(file.toURI().getPath(), filenameAndExtension, currentImage)

STUDENTSUPSI

Approccio

Export Immagine

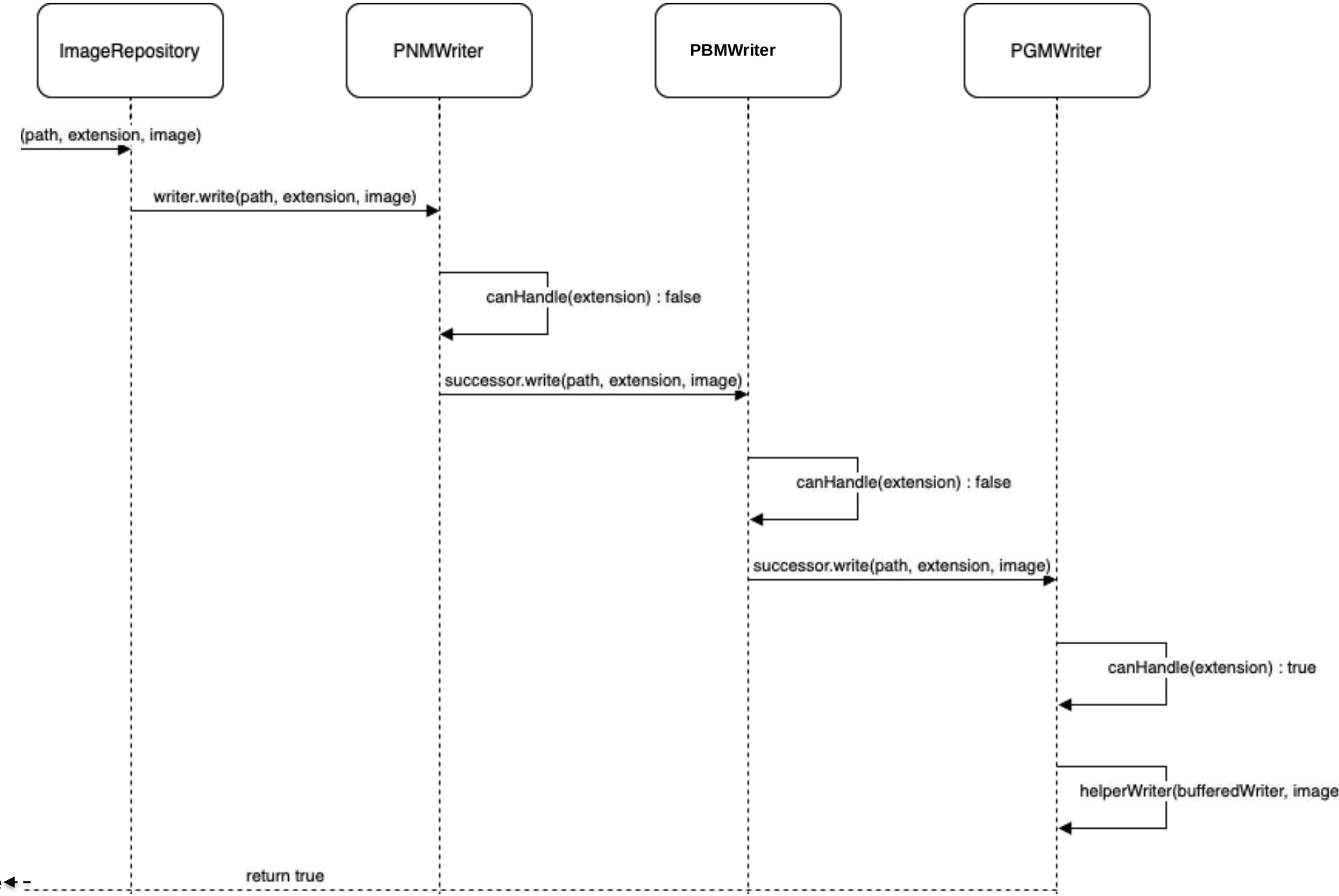


- Come viene associata la **chain of responsibility dei writer?**

ImageRepository → Imposta la **catena** di sucessione **da PBM → PGM → PPM** → **Campo writer** : Writer di partenza assegnato al PBMWriter

- Esempio **Parte 2: UML sequence diagram** per l'export in **PGM**

STUDENTSUPSI

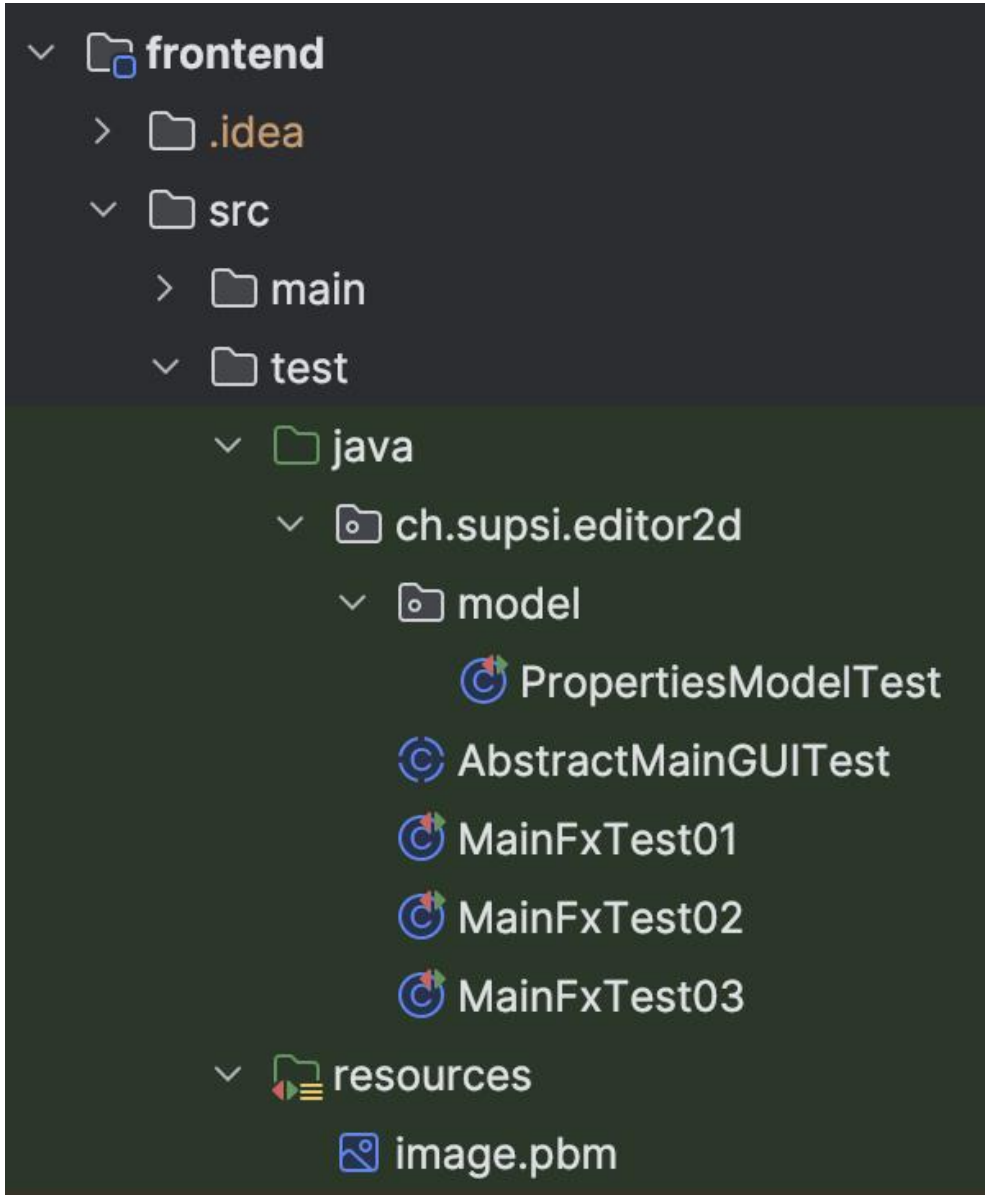


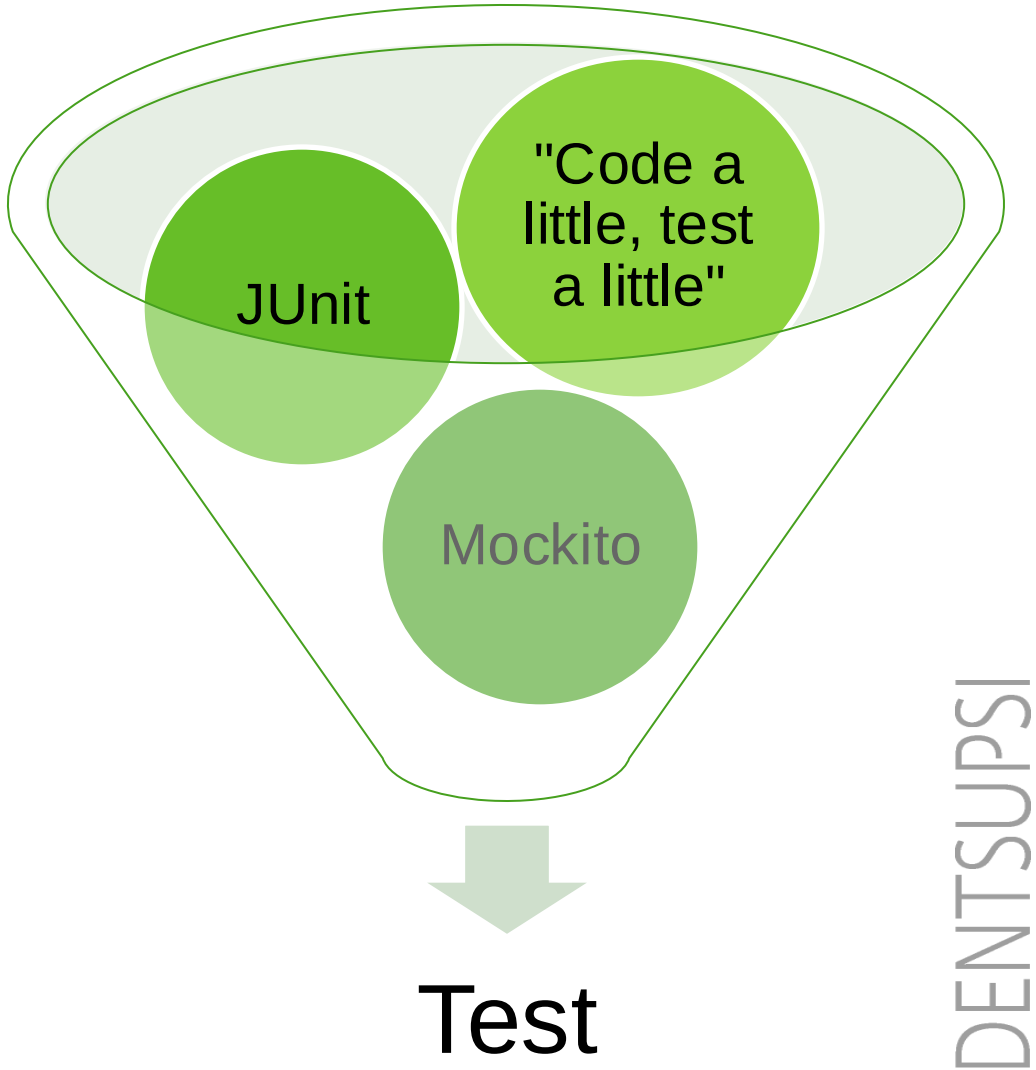
← ImageController ← ImageService

STUDENTSUPSI

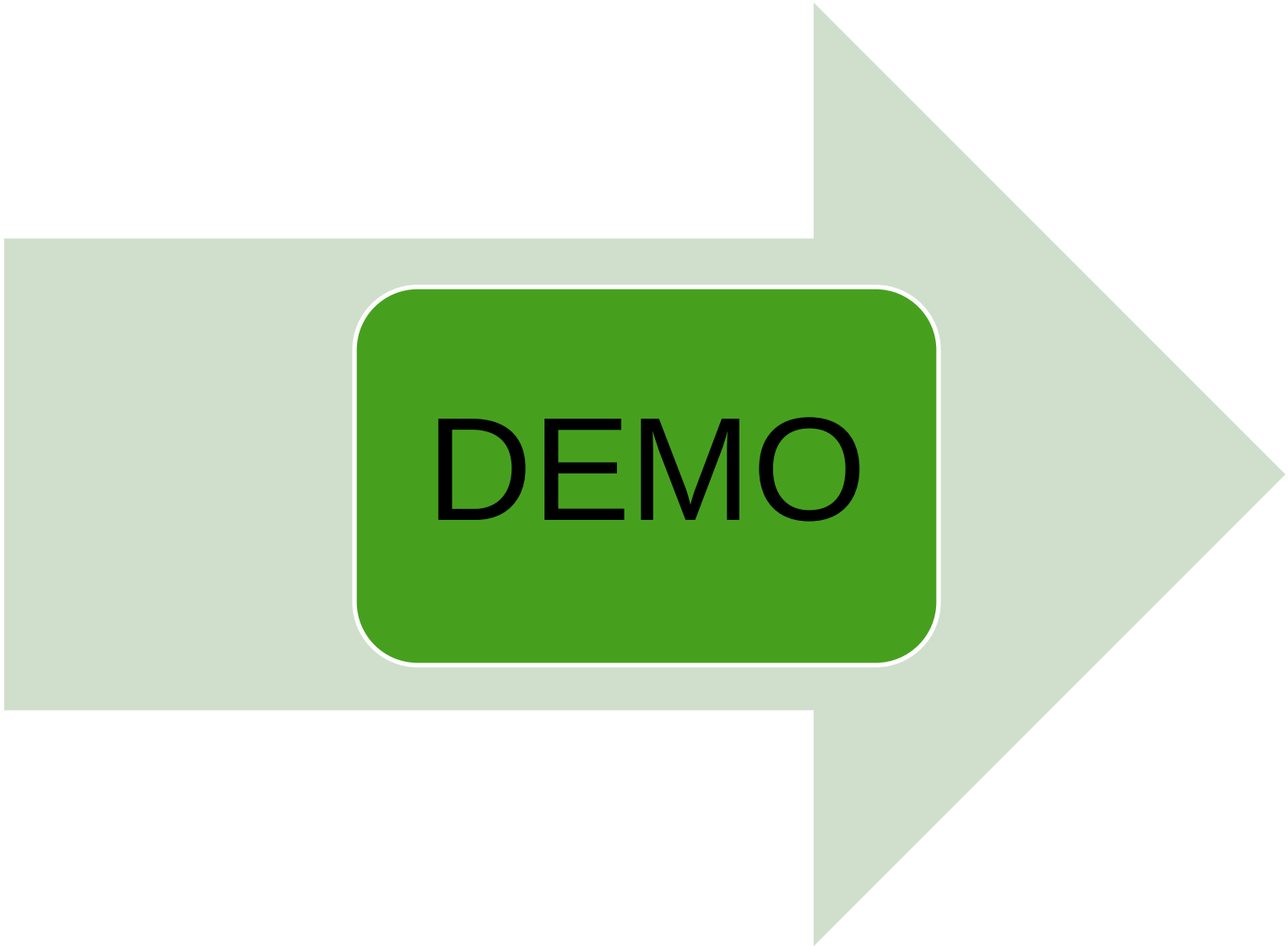
Test

Frontend

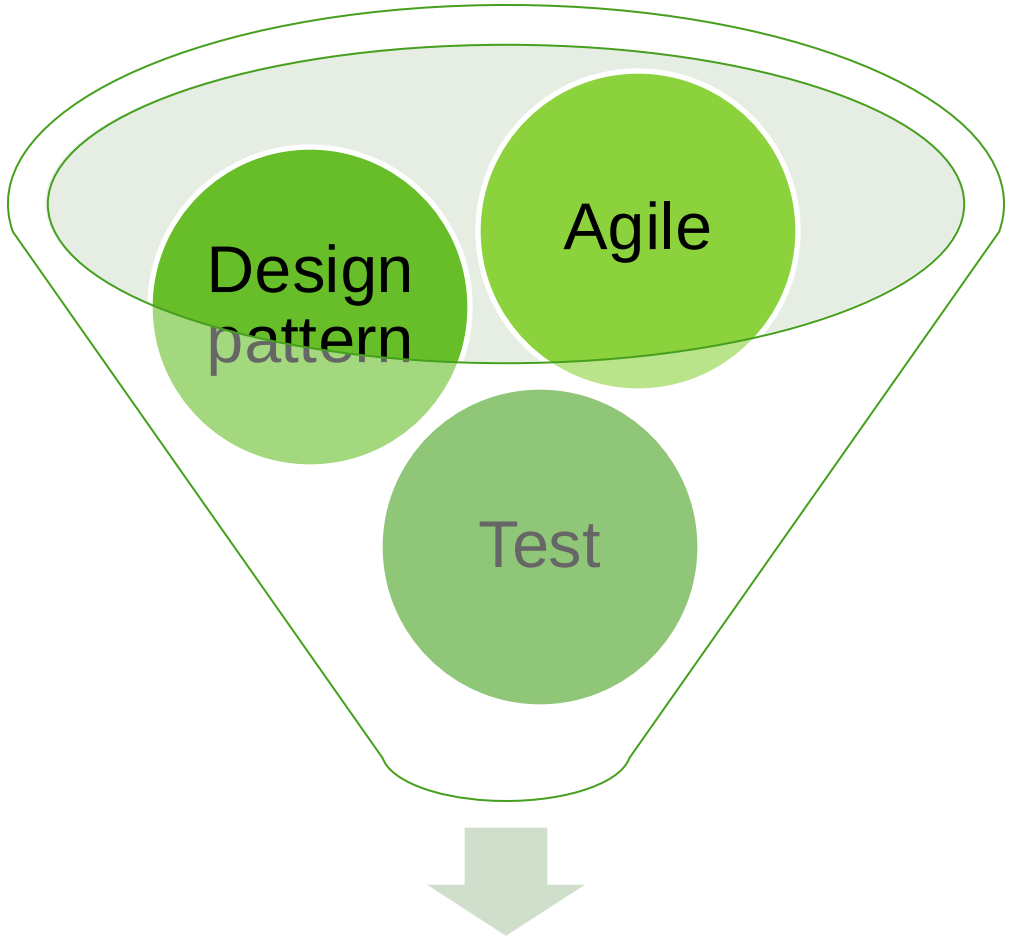




STUDENTSUPSI


DEMO

Conclusione



STUDENTSUPSI