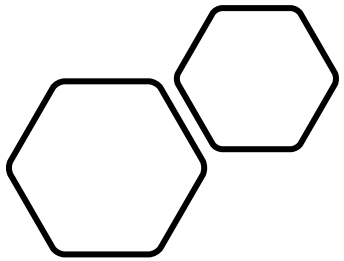
PROGETTO DI INGEGNERIA DEL SOFTWARE

ATM BANCA D'ITALIA



Bottagisi Luca

Cazzaniga Matteo

Ubbiali Paolo

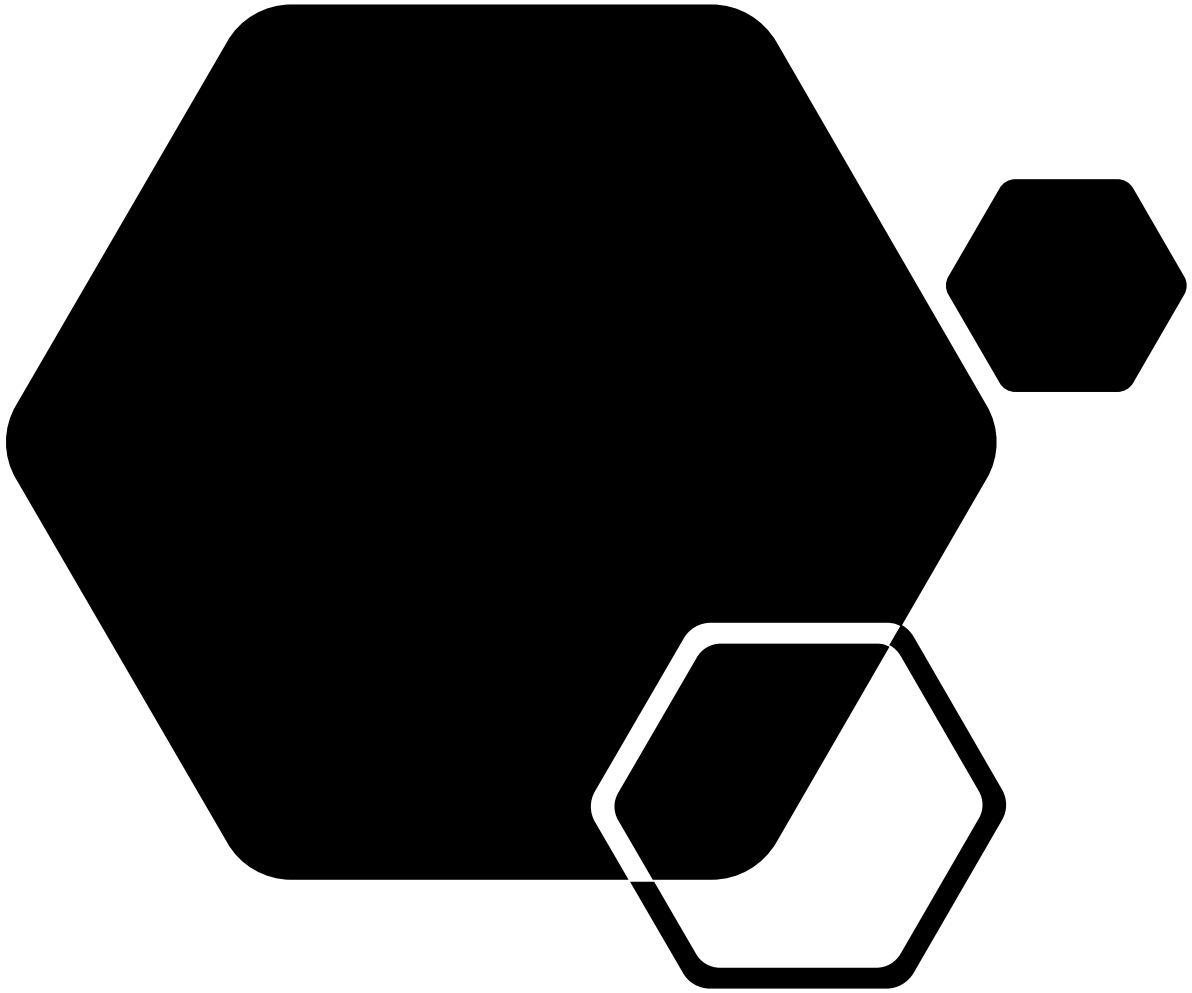
OBIETTIVO

Realizzazione di un software per la gestione degli ATM di una banca.

Ogni ATM è fornito di un lettore di tessere magnetiche, un piccolo display a colori, una tastiera ed una stampante.

L'utente attraverso l'uso di una tessera magnetica ed un PIN personali, accede al singolo ATM con cui può effettuare diverse operazioni come:

- prelievo del contante
- stampa dell'estratto conto
- visualizzazione saldo...



DIFFICOLTÀ INCONTRATE

Non avendo mai creato nulla di simile le difficoltà sono state molteplici tra cui ricordiamo:

- Comunicazione tra diversi linguaggi di programmazione
- Utilizzo di github (inizialmente)
- Implementazione di diverse funzionalità

PARADIGMA DI PROGRAMMAZIONE/ MODELLAZIONE UTILIZZATO E TOOLS

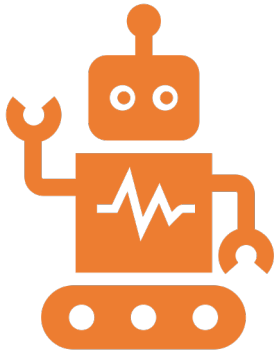
Programmazione: programmazione ad oggetti

Modellazione: UML

Java, SQL, HTML

Eclipse, StarUML, GitHub, mySQL

SOFTWARE CONFIGURATION MANAGEMENT



GitHub

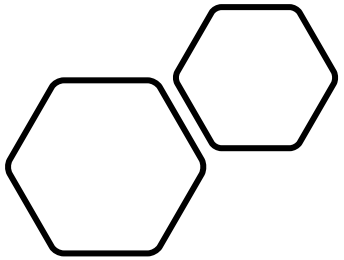
Issues, branches, pull request…

SOFTWARE LIFE CYCLE

METODO SCRUM

- **Analisi requisiti:** esigenze e specifiche software
- **Progettazione:** creazione soluzione per il soddisfacimento requisiti
- **Sviluppo:** scrittura codice
- **Testing:** test di ogni parte del codice e del suo insieme

REQUISITI



I requisiti sono stati specificati nel file ”ANALISI DEI REQUISITI.docx” e sono stati divisi in:

- Funzionali
- Non funzionali
- Tecnologici
- Inversi

Per ricavare questi requisiti, non avendo un vero cliente a disposizione, abbiamo preso sia le parti del cliente che dei programmatori cercando di pensare cosa avrebbe dovuto avere questo progetto per soddisfare le nostre esigenze da cliente.

ARCHITETTURA

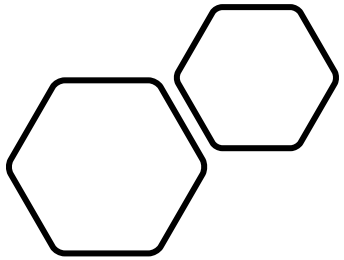
Architettura a tre livelli

1. Strato di presentazione: questo strato gestisce l'interfaccia utente dell'ATM, ovvero il display, la tastiera e gli altri dispositivi di input/output.
2. Strato di business logic: questo strato gestisce la logica di business dell'ATM, ovvero la gestione delle transazioni finanziarie, la validazione dei dati, il controllo degli accessi e la sicurezza.
3. Strato di accesso ai dati: questo strato gestisce l'accesso ai dati dell'ATM, ovvero le informazioni sui conti degli utenti, le transazioni e i registri di attività

DESIGN PATTERN

- MVC (Model-View-Controller): questo pattern prevede la suddivisione dell'applicazione in tre componenti principali: il modello (transazioni), la vista (l'interfaccia utente) e il controller (gestione accessi).
- Metriche di qualità del progetto

MODELLAZIONE



Abbiamo utilizzato i seguenti diagrammi UML:

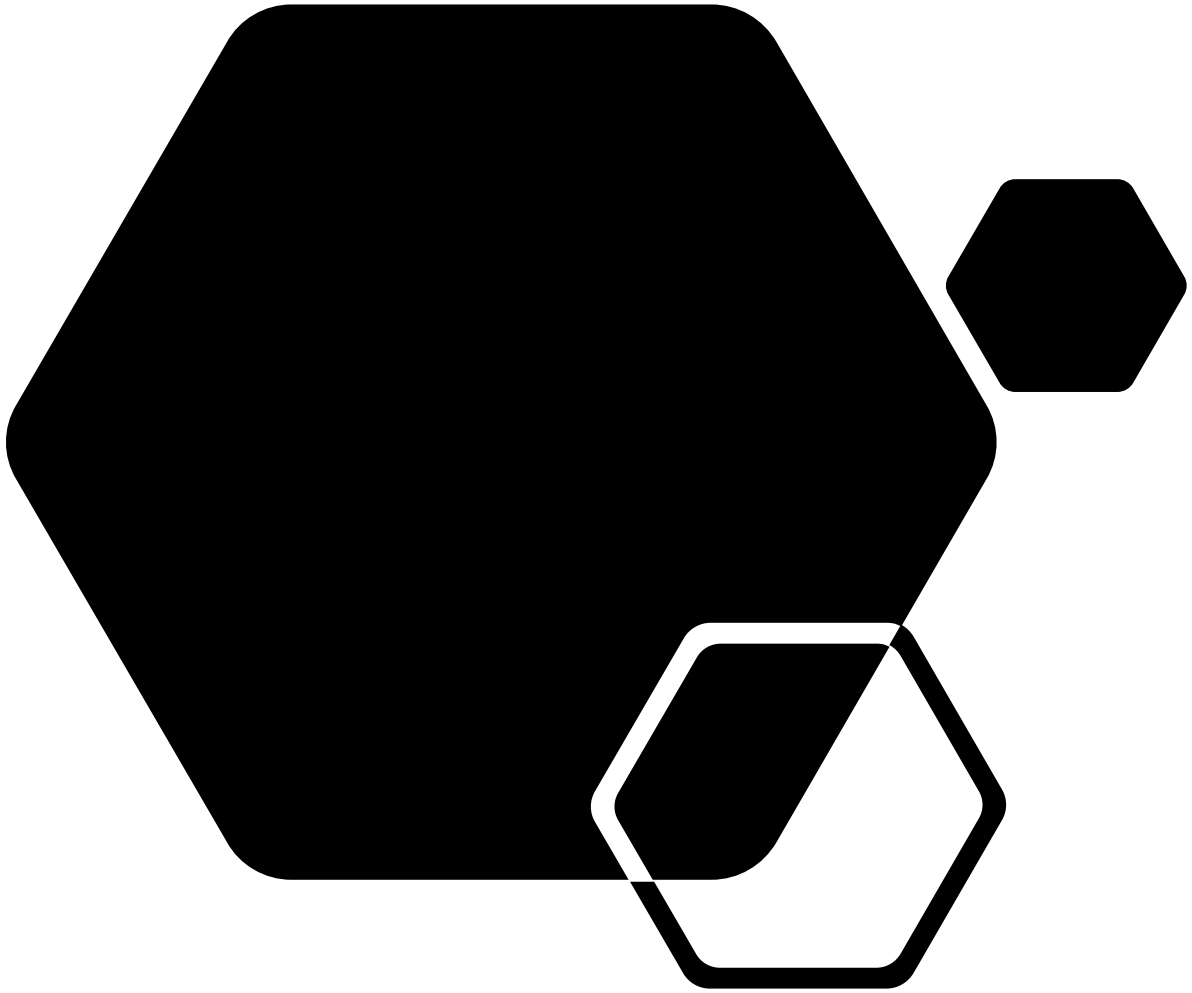
- Casi d’uso
- Attività
- Classi
- Sequenze
- Package
- Stati

IMPLEMENTAZIONE

- Abbiamo utilizzato Eclipse come IDE creando il codice in java.
- mySQL per la creazione e gestione del database.

DEMO

Procediamo con una breve dimostrazione del funzionamento dell’applicazione creata.



TESTING

- Per la creazione dei test ci siamo serviti del test JUnit automatico di Eclipse e abbiamo poi implementato manualmente i vari test.
- *Quali risultati avete ottenuto con l'attività di testing*