

Approfondimento TPSI: GitHub e Sviluppo Software

Un'analisi approfondita degli strumenti e delle metodologie chiave per lo sviluppo software moderno, focalizzandosi su GitHub, TDD e JUnit 5.



GitHub Remote: Collaborazione e Sicurezza



GitHub Remote è la versione online del tuo repository Git locale, essenziale per la collaborazione e il backup del codice.

- Collaborazione con altri sviluppatori.
- Backup sicuro del progetto nel cloud.
- Sincronizzazione del lavoro su più dispositivi.





Rapidi per GitHub Remote

1

Visualizzare Remote

```
git remote -v
```

2

Aggiungere un Nuovo Remote

```
git remote add <nome> <url>
```

3

Modificare URL Remote

```
git remote set-url <nome> <nuovo-url>
```



git-scm.com



Git - git-remote Documentation

Manage the set of repositories ("remotes") whose branches you track.

Utilizzo delle GitHub Issues



Segnalare Bug

Descrivere errori e malfunzionamenti nel software.



Richiedere Funzionalità

Proporre miglioramenti o nuove idee per il progetto.



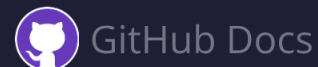
Gestire Task

Suddividere il lavoro in attività più piccole e specifiche.



Fornire Supporto

Coordinare il team e rispondere alle domande degli utenti.



GitHub Docs



About issues - GitHub Docs

Learn how you can use GitHub Issues to track ideas, feedback, tasks, or bugs.

GitHub Issues: Gestione del Progetto

Funzionalità Chiave

- Tracciamento e definizione delle priorità.
- Integrazione con codice e commit.
- Automazione dei flussi di lavoro.
- Collaborazione continua e memoria storica.



- <https://docs.github.com/en/issues>
- <https://docs.github.com/en/issues/planning-and-tracking-with-projects>

Test Driven Development (TDD)

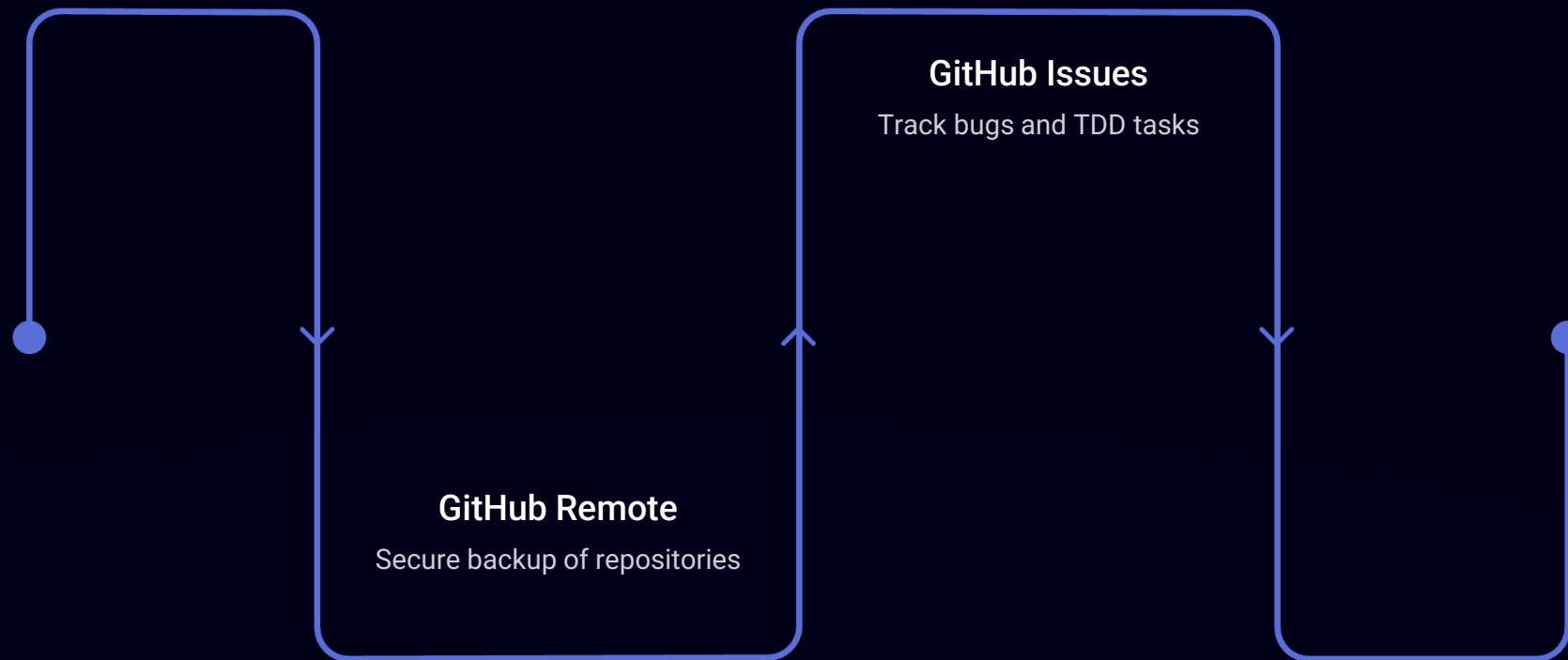
Il TDD è una filosofia di sviluppo che trasforma il modo in cui il software viene progettato e scritto, garantendo che ogni funzionalità sia verificata fin dall'inizio.

"Adottare il TDD significa cambiare l'approccio alla programmazione, passando da una scrittura del codice immediata a una guidata dai test."

- <https://martinfowler.com/bliki/TestDrivenDevelopment.html>
- <https://www.agilealliance.org/glossary/tdd/>



Integrazione TDD con GitHub



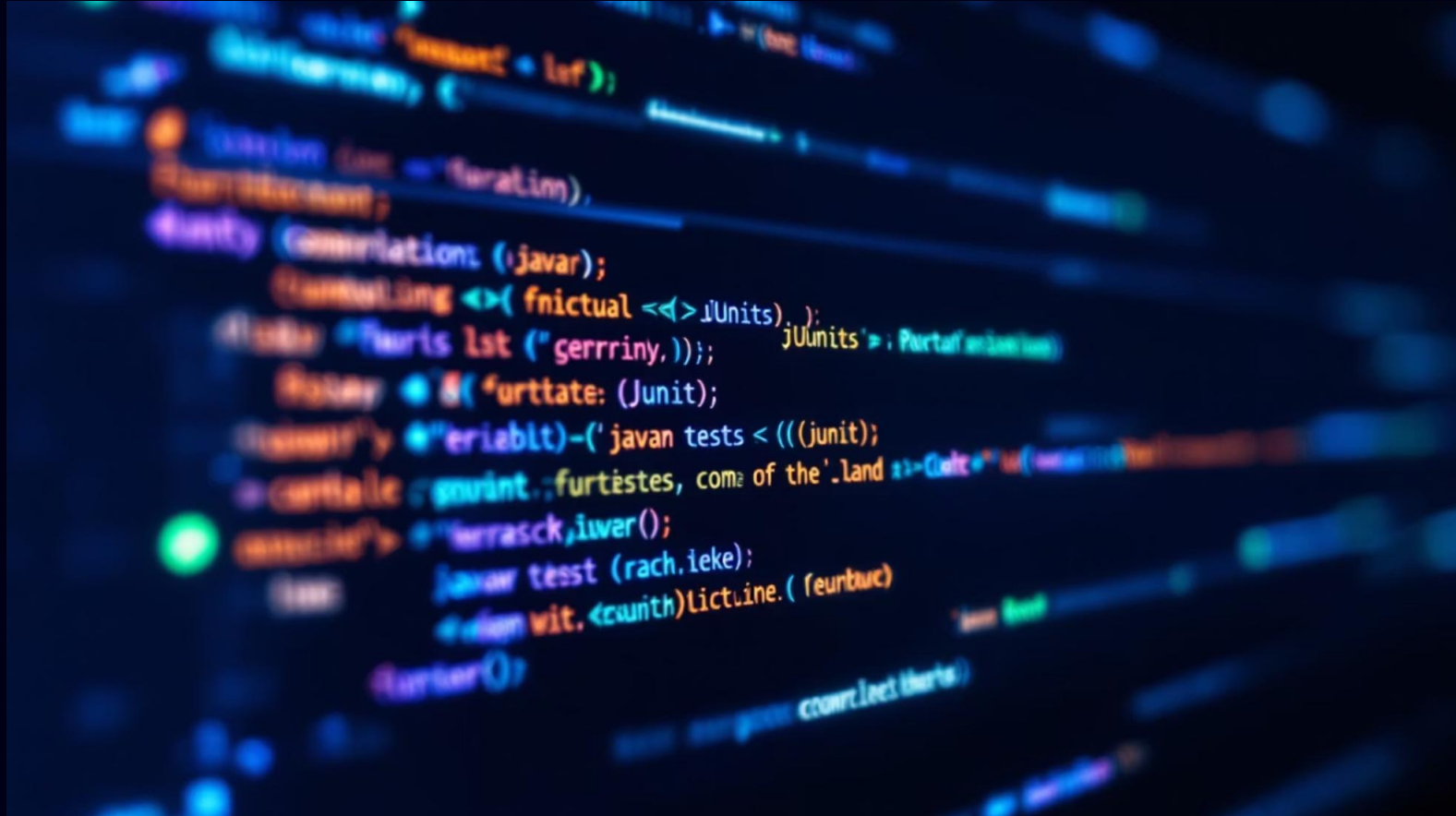
Per gestire correttamente lo sviluppo basato su JUnit e TDD, gli strumenti di collaborazione sono cruciali:

- **GitHub Remote:** Mantiene un backup sicuro e sincronizzato del codice di test e dell'applicazione.
- **GitHub Issues:** Traccia i bug, gestisce i task di sviluppo e crea una "memoria storica" del progetto.

GitHub Remote e Issues sono fondamentali per un flusso di lavoro TDD efficace.

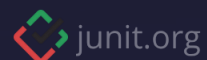
- <https://docs.github.com/en/get-started/quickstart/github-flow>
- <https://docs.github.com/en/issues/tracking-your-work-with-issues/linking-a-pull-request-to-an-issue>

JUnit 5: Il Framework di Test



JUnit 5 è il framework tecnico per implementare test in Java, utilizzato in IntelliJ IDEA per progetti OOP di media complessità.

- Configurazione in IntelliJ IDEA (senza tool di build esterni).
- Applicazione su progetti OOP con UML e Git.
- Focus sulla qualità dei test e gestione dei potenziali bug nei test stessi.



Overview :: JUnit User Guide

The goal of this document is to provide comprehensive reference documentation for programmers writing tests, extension authors, and engine authors as well as build tool and IDE vendors.



Sviluppo Integrato: Visione d'Insieme



L'integrazione di TDD, JUnit 5 e GitHub crea un ecosistema di sviluppo robusto e collaborativo.

- <https://docs.github.com/>
- <https://junit.org/junit5/>
- <https://martinfowler.com/bliki/TestDrivenDevelopment.html>