

Esercitazioni

Prova Finale 2019

Part 1 & 2 - Recap

Docente Gianpaolo Cugola

Esercitatore Mario Scrocca

Code: <https://github.com/marioscrocca/ingsoft-prova-finale-19>

Git

- Salvare lo stato di ogni file del progetto (come se si facesse una foto del progetto in un certo istante.)
- Tali istantanee sono detti **commit**
- Albero dei commit, dove lo storico di sviluppo può separarsi in più branch
- Gli utenti di un repository possono inviare modifiche **simultaneamente** a patto che le modifiche non siano sulle stesse linee di codice
- Componenti di un team possono condividere tra loro le modifiche fatte indipendentemente

Aree di lavoro

- **Working directory**

- le cartelle di lavoro, che modificate come volete (IntelliJ, Eclipse, vim..)

- **Staging area:**

- per le **modifiche** che potrebbero diventare definitive:
i.e., codice scritto ma non ancora testato
- Modifiche := modifiche a un file esistente, aggiunta di un file/cartella, rimozione di un file/cartella

- **Repository (locale):**

- per le modifiche definitive: avete testato e funziona tutto
- Append-only: dopo una aggiunta di modifiche, **non potete cancellarle**; ma potete scrivere sopra altre modifiche che annullano “di fatto” le precedenti

- **Repository (remoto):**

- sequenza di modifiche su una macchina remota
- stessa filosofia di quella locale
- il gestore del repo deve darvi i permessi di accesso e scrittura

Advice

- Pullare sempre prima di iniziare a lavorare.
- Committare, pullare e poi pushare.
- In caso di conflitto risolvere conflitti, committare e pushare (nel caso di dubbi coordinandosi con chi ha fatto l'altra modifica).
- Don't Panic! (usa Google).

Wrong Commit

- **git revert HEAD** La history avanza SEMPRE: git revert crea un NUOVO commit che annulla gli effetti del (dei) precedente
- <https://stackoverflow.com/questions/4114095/how-to-revert-a-git-repository-to-a-previous-commit>

Don't give in to the dark side



IntelliJ + Maven

- Download and install Java SE JDK from Oracle website (ok da Java 8 in su)
- Download IntelliJ
- Community Edition ok, come studenti potete ottenere gratuitamente anche la versione a pagamento
- Maven dependencies: Use Maven central <https://search.maven.org> and copy the dependency in your pom
- Maven install for local dependencies

IntelliJ + Maven + Sonar

- al pom.xml aggiungere

```
<properties>  
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
  <sonar.language>java</sonar.language>  
  <sonar.host.url> http://localhost:9000/ </sonar.host.url>  
</properties>
```

- Avviare Sonar server
- Run with goal: “sonar:sonar”, e poi andate alla pagine web
- Aggiungere *Jacoco* al pom per esportare anche la coverage dei tests
- ***SonarLint***: Plugin per Eclipse che si connette a Sonar e mostra i risultati dell’analisi nella finestra del codice. Potete vedere le annotazioni più importanti nel progetto e altre info in una finestra a parte

Testing

- Folder *test* nel progetto Maven
- Unit, Integration, System



OO Programming - *My two cents*

- Must-see:

Polly want a message by Sandi Metz

https://www.youtube.com/watch?v=XXi_FBrZQiU

- Must-read:

*Clean Code: A Handbook of Agile Software
Craftsmanship*