



**POLITECNICO**  
MILANO 1863

# Introduzione a GIT

19 Marzo 2019

Scaglione San Pietro

RESPONSABILI

Giovanni Meroni  
Amarildo Likmeta

TUTOR

Valentina Deda  
Marco Bacis

# Presentazione di GIT

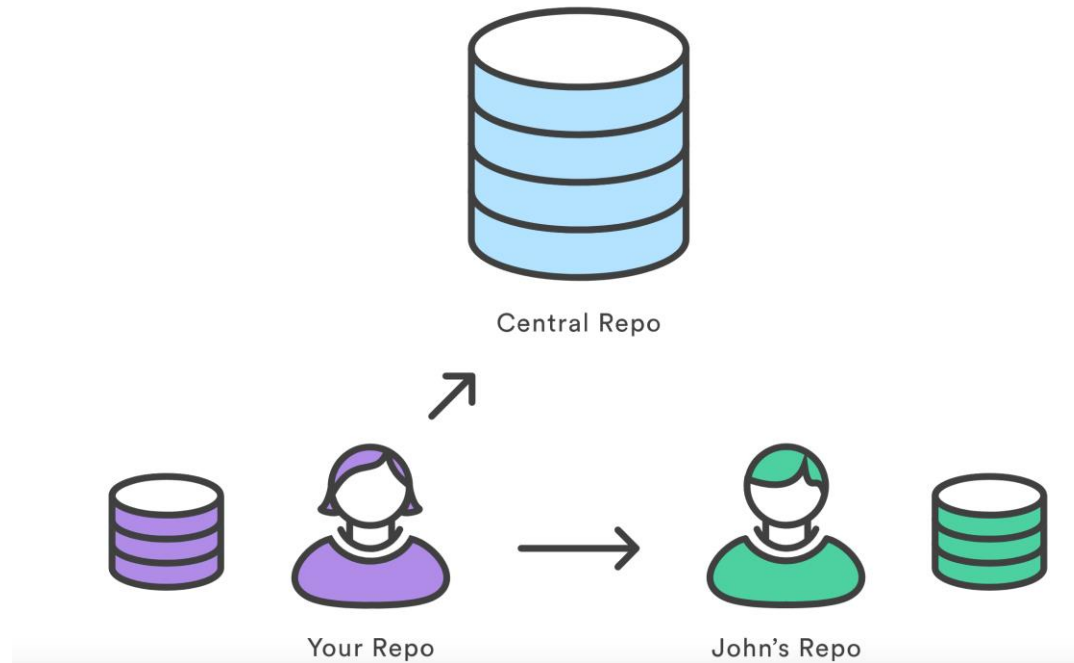
- Git è un Version Control System (VCS) decentralizzato
- Chiamasi VCS un software che serve a tracciare lo sviluppo di un progetto
- Ciò offre numerose comodità in fase di sviluppo:
  - È possibile salvare lo stato di ogni file del progetto (commit)
  - È possibile tornare indietro di una versione, anche per singolo file, se si riscontrano problemi (revert)
  - È possibile lavorare in parallelo, senza intralciarsi, su più funzionalità del progetto (branch)
  - È possibile fondere più branch sviluppate separatamente in una nuova versione dell'applicazione (merge)

# Presentazione di GIT

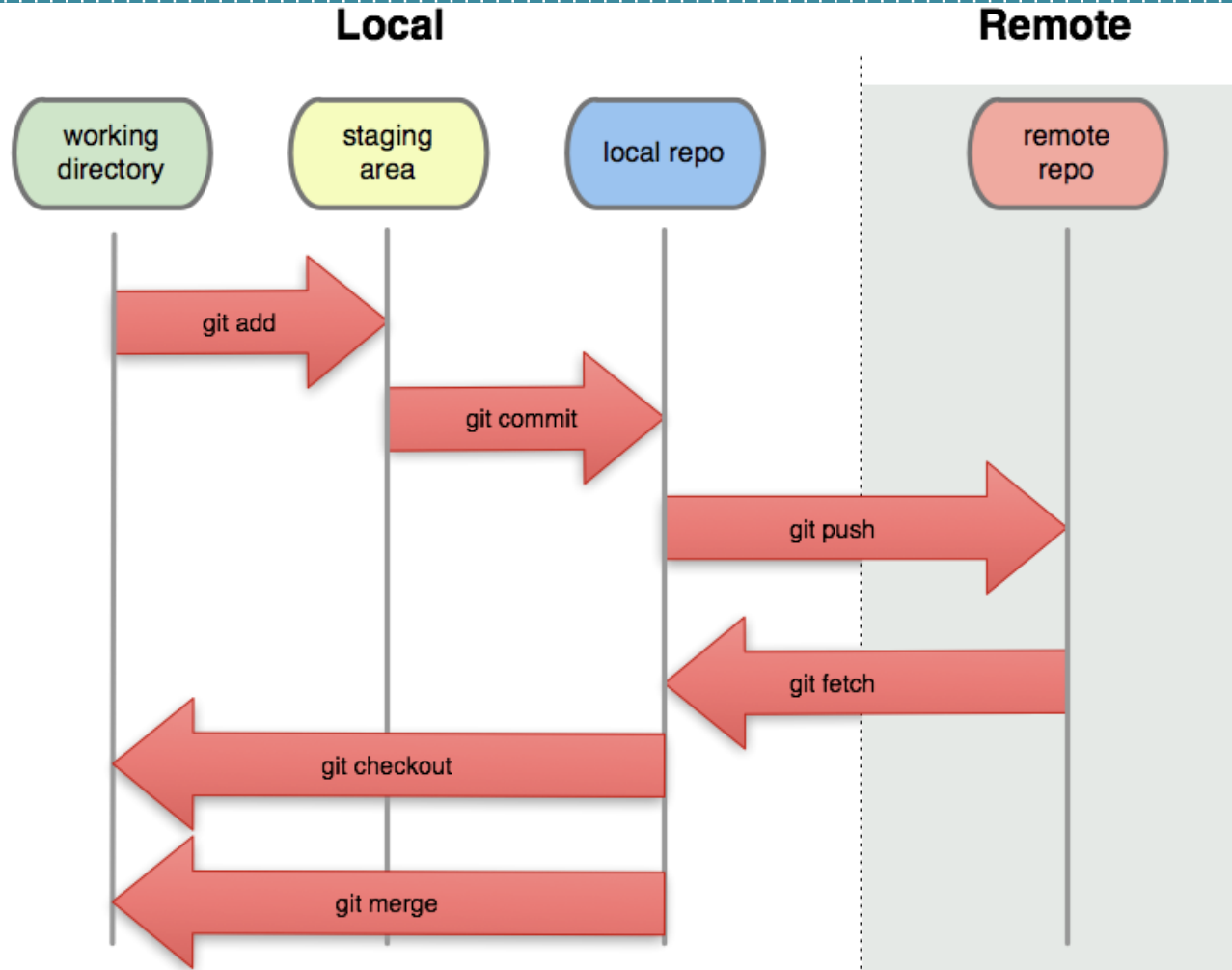
- GIT prevede tre aree di lavoro locali ed una remota:
  - **Working directory**: le cartelle di lavoro, modificabili a piacere (IntelliJ, Eclipse, vim..)
  - **Staging area**: per le modifiche che potrebbero diventare definitive (i.e., codice scritto ma non ancora testato)
  - **Repository (locale)**: per le modifiche definitive (i test hanno dato riscontro positivo)
  - **Repository (remoto)**: sequenza di modifiche su una macchina remota (stessa filosofia di quella locale, ma il gestore del repo deve darvi i permessi di accesso e scrittura)

# Workflow tipico di GIT

- Gli utenti di un repository possono inviare modifiche simultaneamente
  - A patto che le modifiche non siano sulle stesse linee di codice
- Possono condividere tra loro le modifiche fatte indipendentemente



# Workflow tipico di GIT



# Stato dei file

- In GIT ogni file nel progetto può assumere quattro stati:
  - Untracked - non tracciato, non è un vero e proprio stato
  - Modified - il file è stato modificato
  - Staged - il file è stato contrassegnato per essere salvato nel database
  - Committed - una copia del file è salvata al sicuro nel database Git
- I commit possono essere trasferiti in repo remoto (i.e. push)

# Da ricordare

- Git è un tool nato, fatto e usato da professionisti
- Quindi, se vi dà errore ... HA RAGIONE LUI!!!
- L'opzione --force di molti comandi è IL modo per fare ulteriori danni
- Cercare nella documentazione (man, git-scm.com) e su StackOverflow
- Qualora un commit introducesse errori, NON bisogna cancellarlo, ma fare revert dei file problematici, quindi di nuovo commit.
- La logica di GIT (e dei VCS in genere) è che le modifiche debbano SEMPRE essere additive rispetto allo stato corrente, e MAI scomparire

# Risorse utili

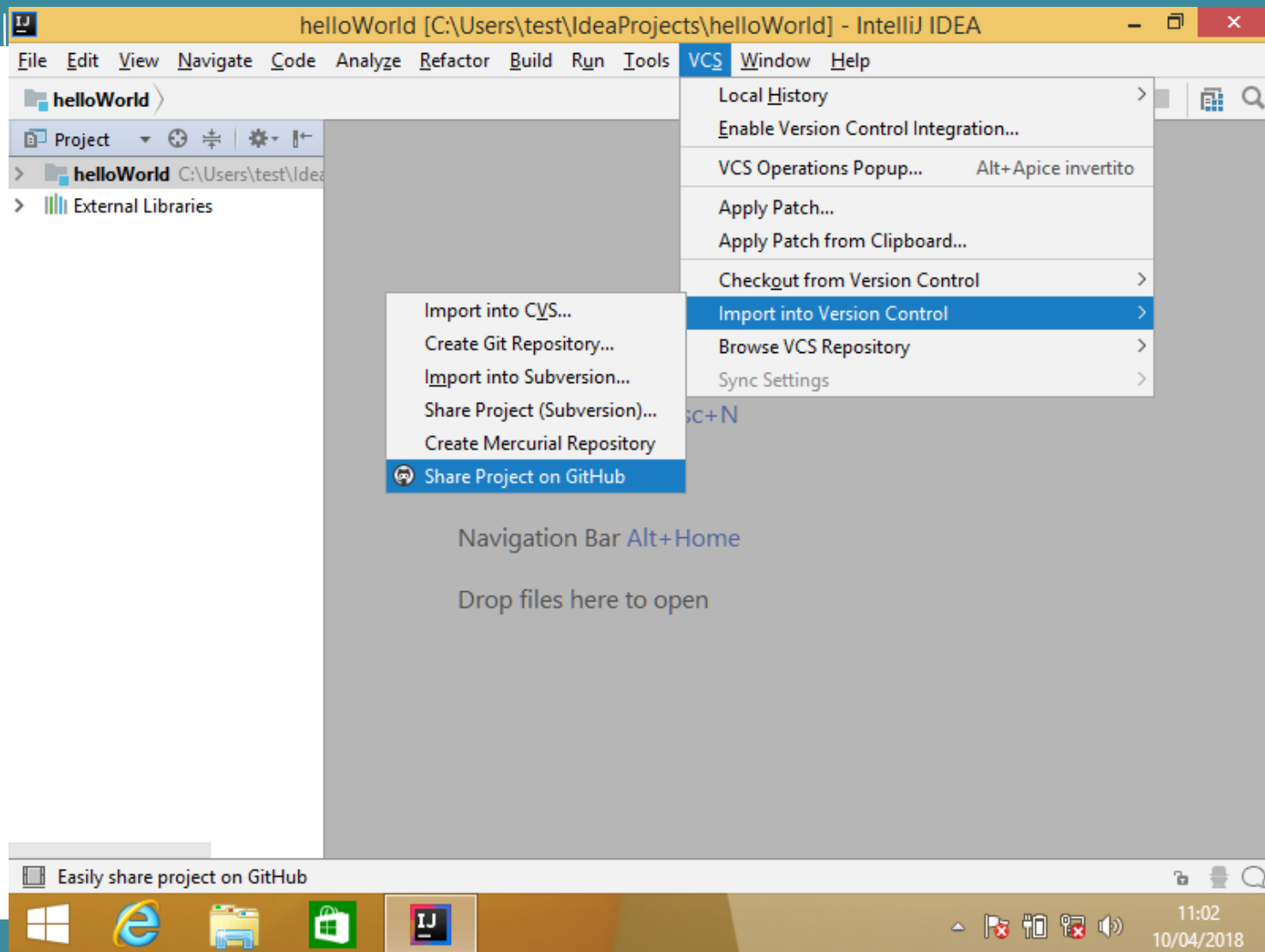
- Moltissimo materiale online, ovunque (Google, StackOverflow, ...); tra cui
- il man della linea di comando, per comando: `man git-add`, `man git-commit`, `man git-push`, ...
- sito ufficiale: <https://git-scm.com/doc>
- Tutorials Atlassian: <https://www.atlassian.com/git/tutorials>
- GitHub help: <https://help.github.com/articles/git-and-github-learning-resources/>



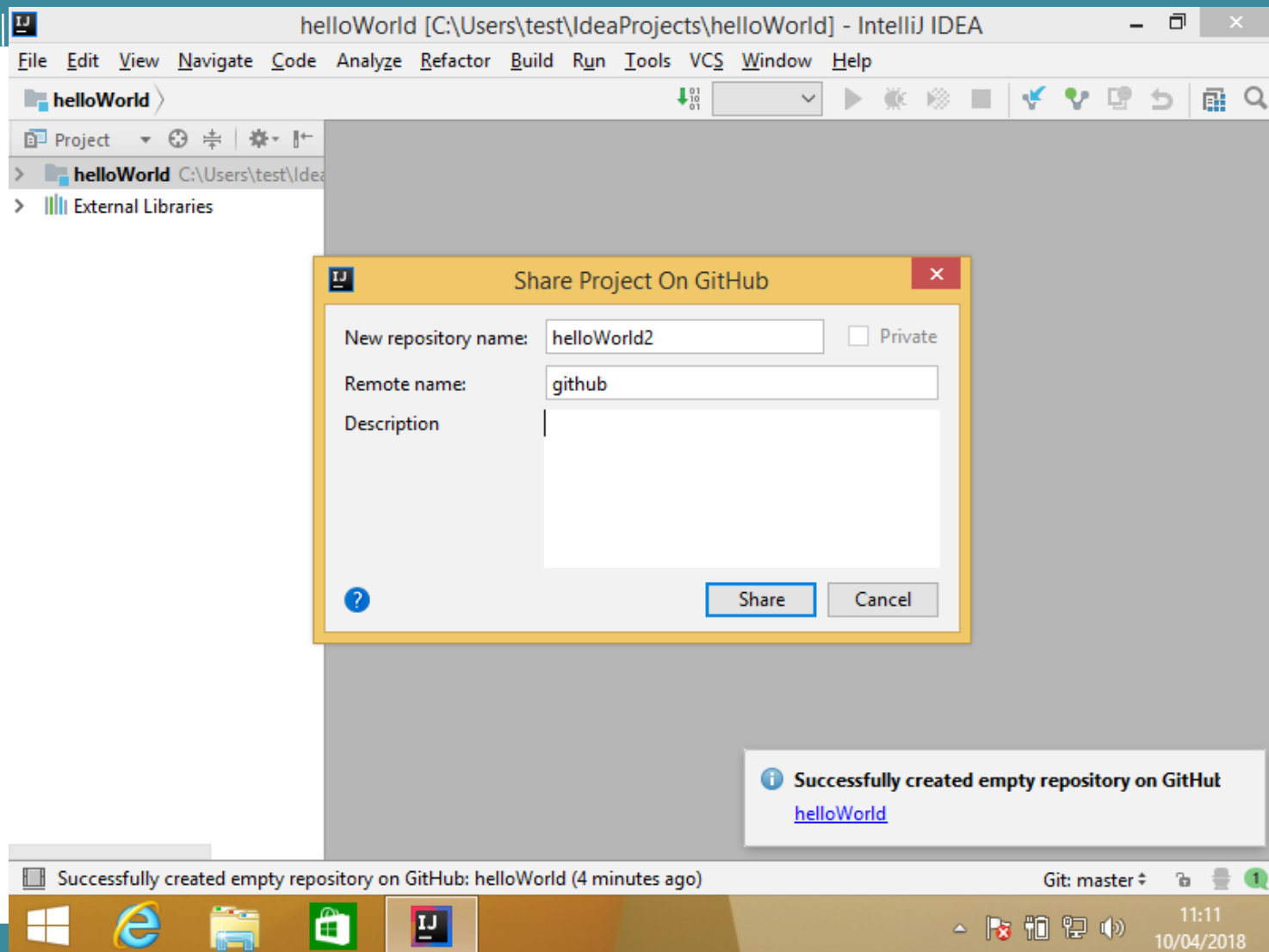
# Integrazione GitHub – IntelliJ IDEA

- **Importante: seguire questi passi solo se siete il capogruppo (e non avete ancora creato il repository su GitHub)**
- Scaricare ed installare il client GIT dal seguente indirizzo: <https://git-scm.com/download/win>
- Lasciare le impostazioni di default
- Una volta aperto il vostro progetto in IntelliJ IDEA, selezionare VCS, quindi Import into version control, poi Share project on GitHub
- In Auth type scegliere Password, quindi inserire le vostre credenziali GitHub
- Nella nuova finestra, dare il nome al vostro progetto come da nostre istruzioni, **spuntare la casella Private**, quindi fare clic su Share
- Selezionare quali file pubblicare nel commit iniziale, quindi fare clic su Ok
- Se il commit fallisce, ripeterlo andando in VCS, quindi Commit. Selezionare i file, scrivere il commento e fare clic su Ok. Indicare username ed email, quindi reinserire le credenziali GitHub

# Integrazione GitHub – IntelliJ IDEA



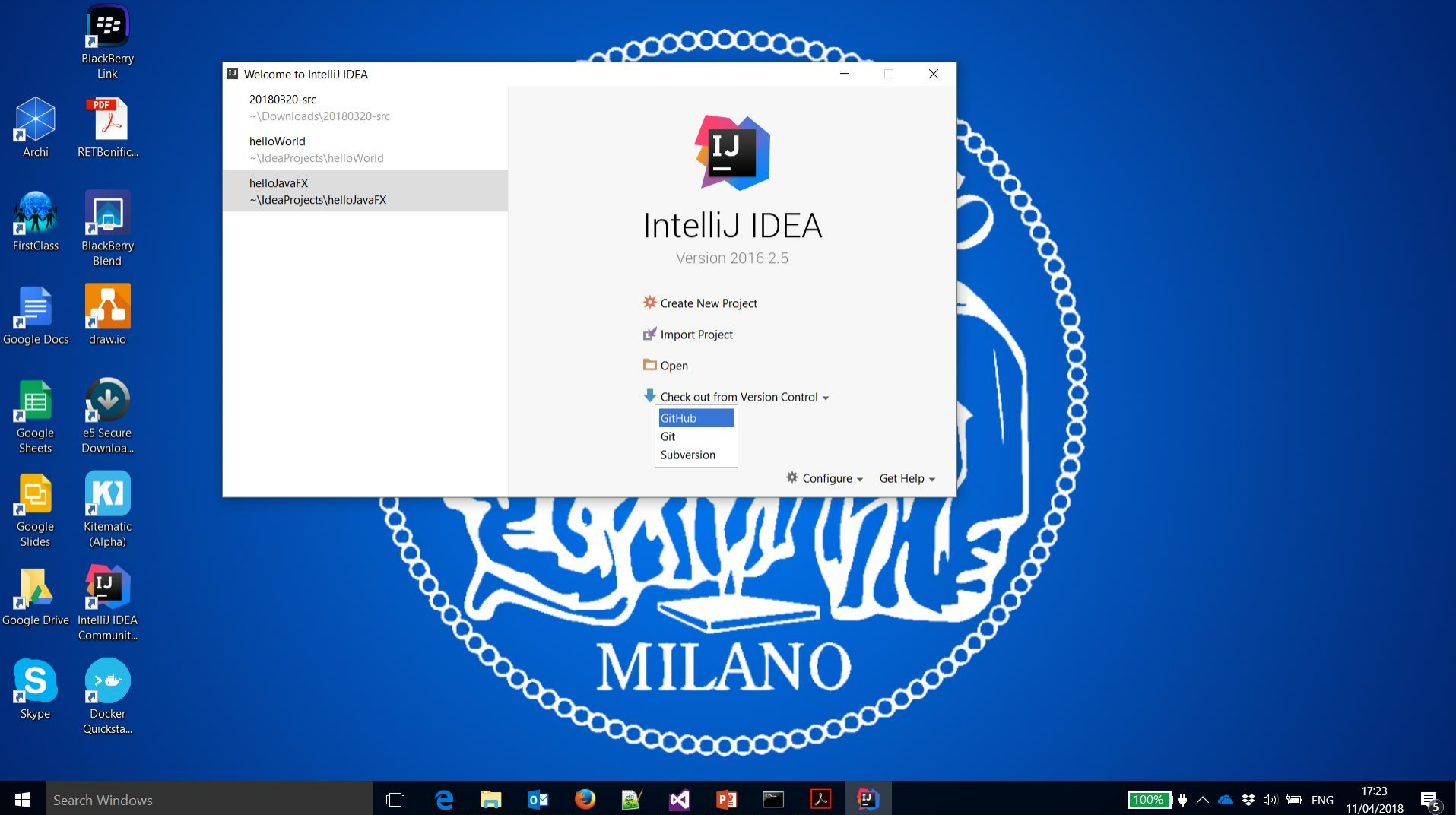
# Integrazione GitHub – IntelliJ IDEA



# Integrazione GitHub – IntelliJ IDEA

- **Importante: seguire questi passi solo se non siete il capogruppo, ed il capogruppo ha già creato il repository su GitHub e fatto il commit iniziale**
- Scaricare ed installare il client GIT dal seguente indirizzo: <https://git-scm.com/download/win>
- Lasciare le impostazioni di default
- Lanciare IntelliJ IDEA e chiudere eventuali progetti attivi
- Dalla schermata iniziale, selezionare Check out from version control, quindi GitHub
- Se richiesto, inserire le proprie credenziali GitHub
- Selezionare il repository del progetto, quindi fare clic su Clone, poi su Yes

# Integrazione GitHub – IntelliJ IDEA



# Integrazione GitHub – IntelliJ IDEA

