PIEMONTE
FONDI STRUTTURALI E DI INVESTIMENTO EUROPEI 2014/2020
fondo sociale europeo
ITS TECNOLOGIE DELL'INFORMAZIONE E DELLA COMUNICAZIONE

Fondamenti di Version Control

Michele Leone

SVN contro GIT

in collaborazione con:
ISTITUTI TECNICI SUPERIORI
Ministero dell'Istruzione
UNIONE EUROPEA
REGIONE PIEMONTE
per una crescita intelligente, sostenibile ed inclusiva
www.regione.piemonte.it/europa2020
INIZIATIVA CO-FINANZIATA CON FSE

Fondamenti di Version Control - What?

- Non esiste un sistema migliore in assoluto rispetto ad un altro, ci sono pregi e difetti che vanno soppesati al momento di decidere quale adottare

Fondamenti di Version Control - What?

- Non esiste un sistema migliore in assoluto rispetto ad un altro, ci sono pregi e difetti che vanno soppesati al momento di decidere quale adottare
- A oggi Git è il sistema di Version Control più utilizzato per questioni di rapidità, efficienza, sicurezza

Fondamenti di Version Control - What?

- Non esiste un sistema migliore in assoluto rispetto ad un altro, ci sono pregi e difetti che vanno soppesati al momento di decidere quale adottare
- A oggi Git è il sistema di Version Control più utilizzato per questioni di rapidità, efficienza, sicurezza
- Vedremo a titolo di esempio le differenze principali tra un sistema di version control "old school" come SVN e uno "new school" come Git

Fondamenti di Version Control - What?

- SVN è uno strument di Version Control **NON DISTRIBUITO**

Fondamenti di Version Control - What?

- SVN è uno strument di Version Control **NON DISTRIBUITO**
  - **NON DISTRIBUITO** = il repository esiste solo in remoto, si può scaricare il progetto e per verificare ogni sorgente sarà necessaria una sincronizzazione online col repository

Fondamenti di Version Control - What?

- SVN è uno strument di Version Control **NON DISTRIBUITO**
  - **NON DISTRIBUITO** = il repository esiste solo in remoto, si può scaricare il progetto e per verificare ogni sorgente sarà necessaria una sincronizzazione online col repository
- Git è uno strument di Version Control **DISTRIBUITO**

Fondamenti di Version Control - What?

- SVN è uno strumento di Version Control **NON DISTRIBUITO**
  - **NON DISTRIBUITO** = il repository esiste solo in remoto, si può scaricare il progetto e per verificare ogni sorgente sarà necessaria una sincronizzazione online col repository
- Git è uno strumento di Version Control **DISTRIBUITO**
  - **DISTRIBUITO** = chiunque può **CLONARE** il repository centrale ed avere sulla propria macchina l'intera storia di quel progetto

Fondamenti di Version Control - What?

GIT	SVN
Git è open source; è stato sviluppato da Linus Torvalds nel 2005. ha la sua forza nella rapidità e nell'integrità dei dati	Apache Subversion è open source sotto Apache license.
strumento Distribuito	strumento Centralizzato
ogni utente ha la propria "copia" dei sorgenti in locale	esiste un repository centrale al quale bisogna collegarsi per scaricare il progetto, confrontare il codice, committare

Fondamenti di Version Control - What?

GIT	SVN
NON serve una connessione per effettuare le operazioni	è necessaria una connessione per effettuare le operazioni
più difficile da imparare (ha una curva di apprendimento Maggiore), ha più concetti e comandi di SVN	molto più semplice da padroneggiare rispetto a git.
non ha un'interfaccia utente nativa	ha un'interfaccia utente semplice

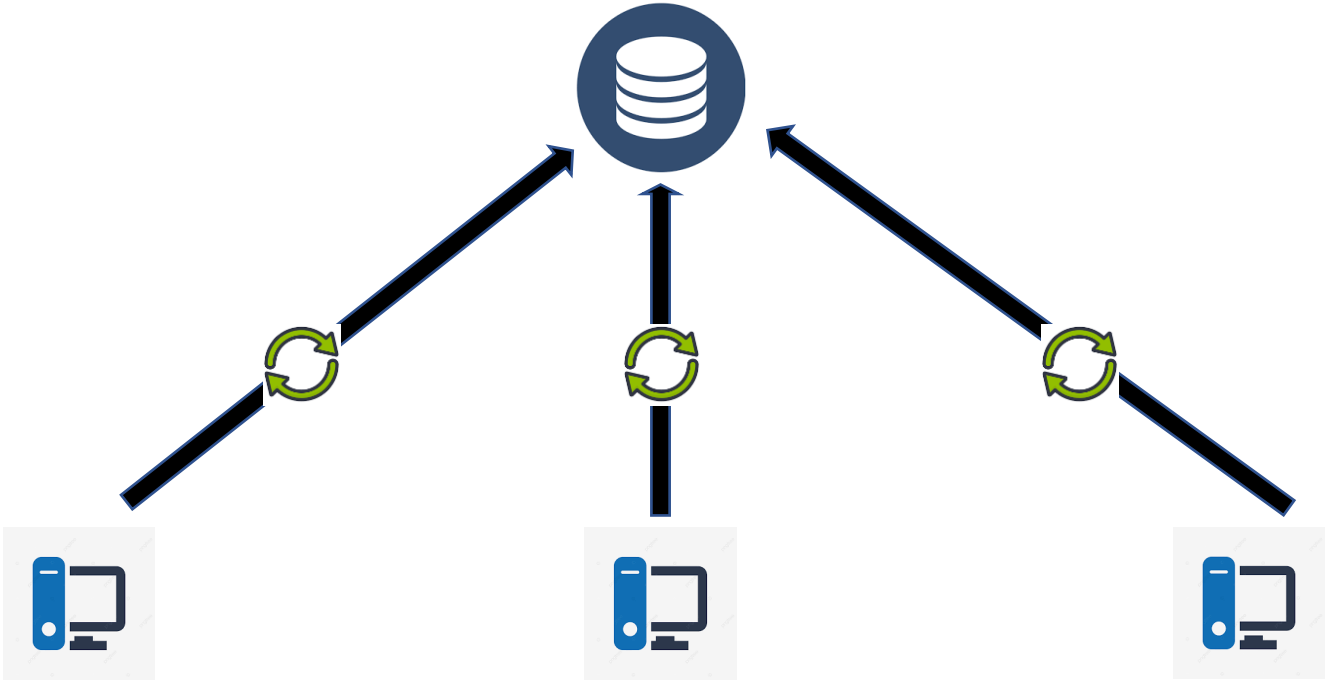
Fondamenti di Version Control - What?

- ... e allora perché Git???

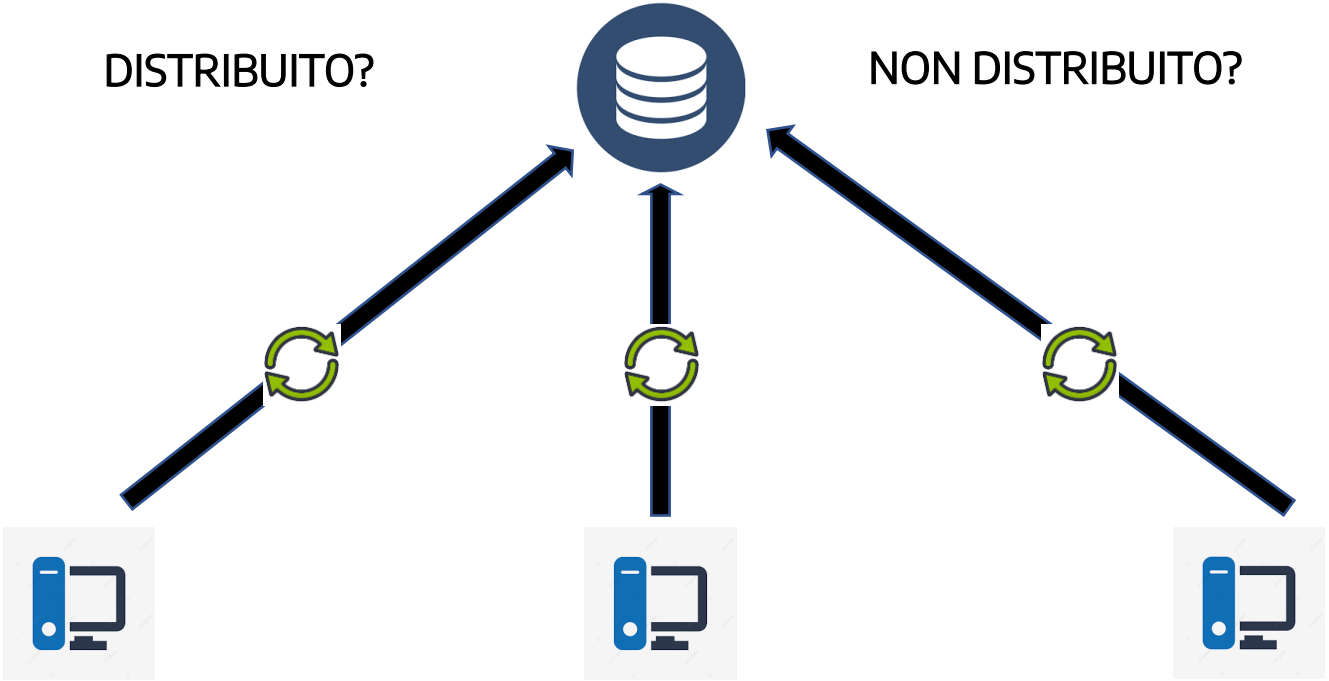
Fondamenti di Version Control - What?

- Riprendiamo lo schema visto in precedenza...

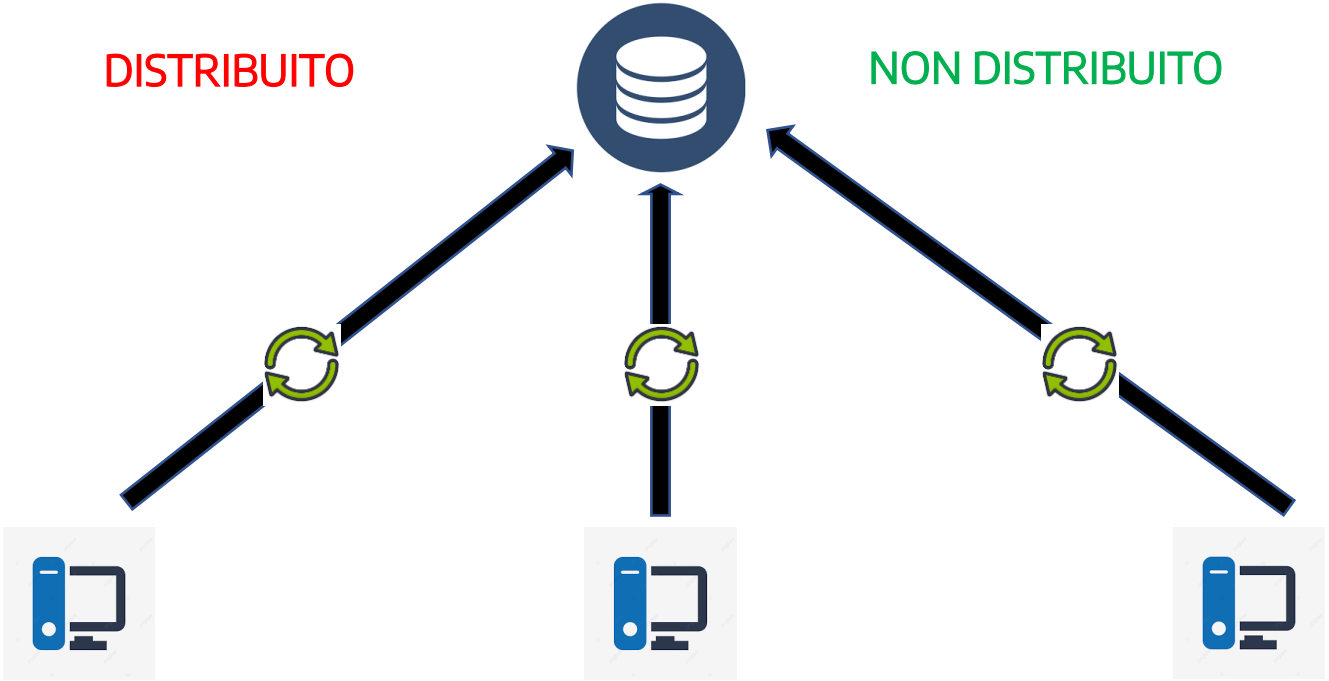
Fondamenti di Version Control - What?



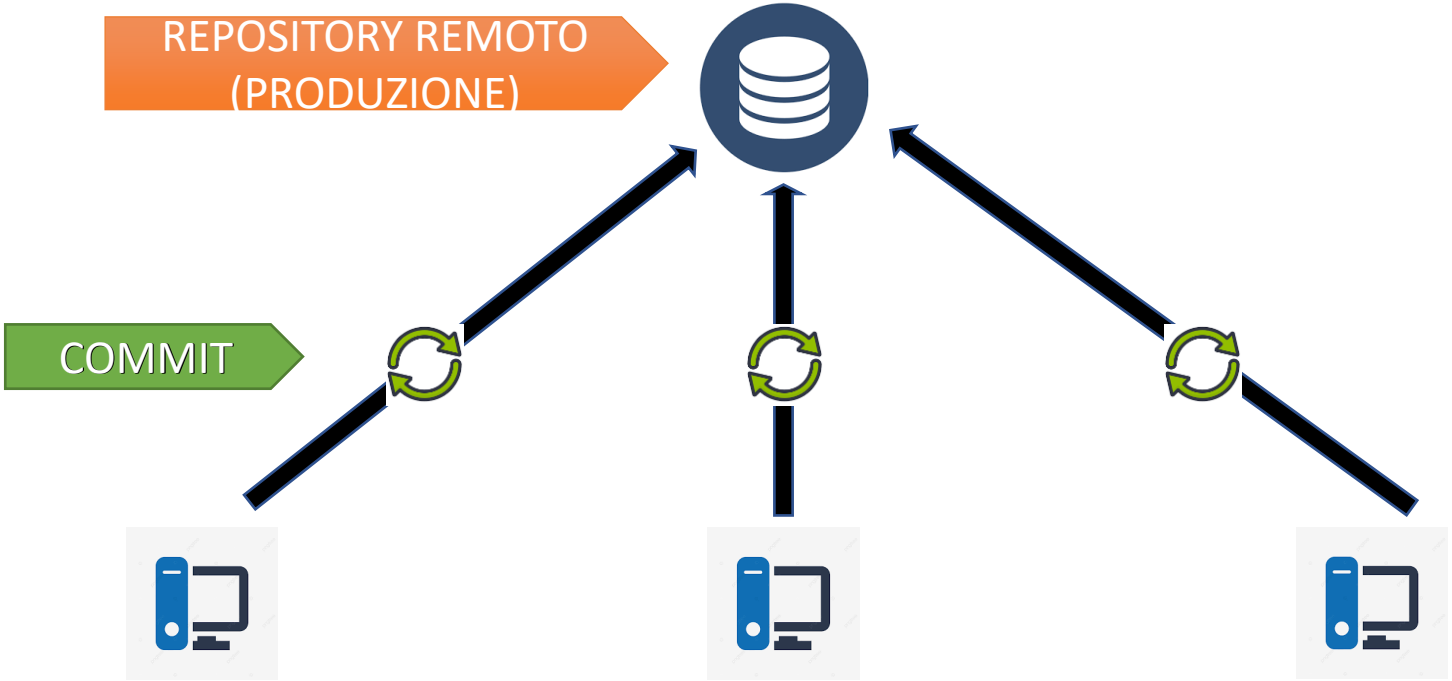
Fondamenti di Version Control - What?



Fondamenti di Version Control - What?



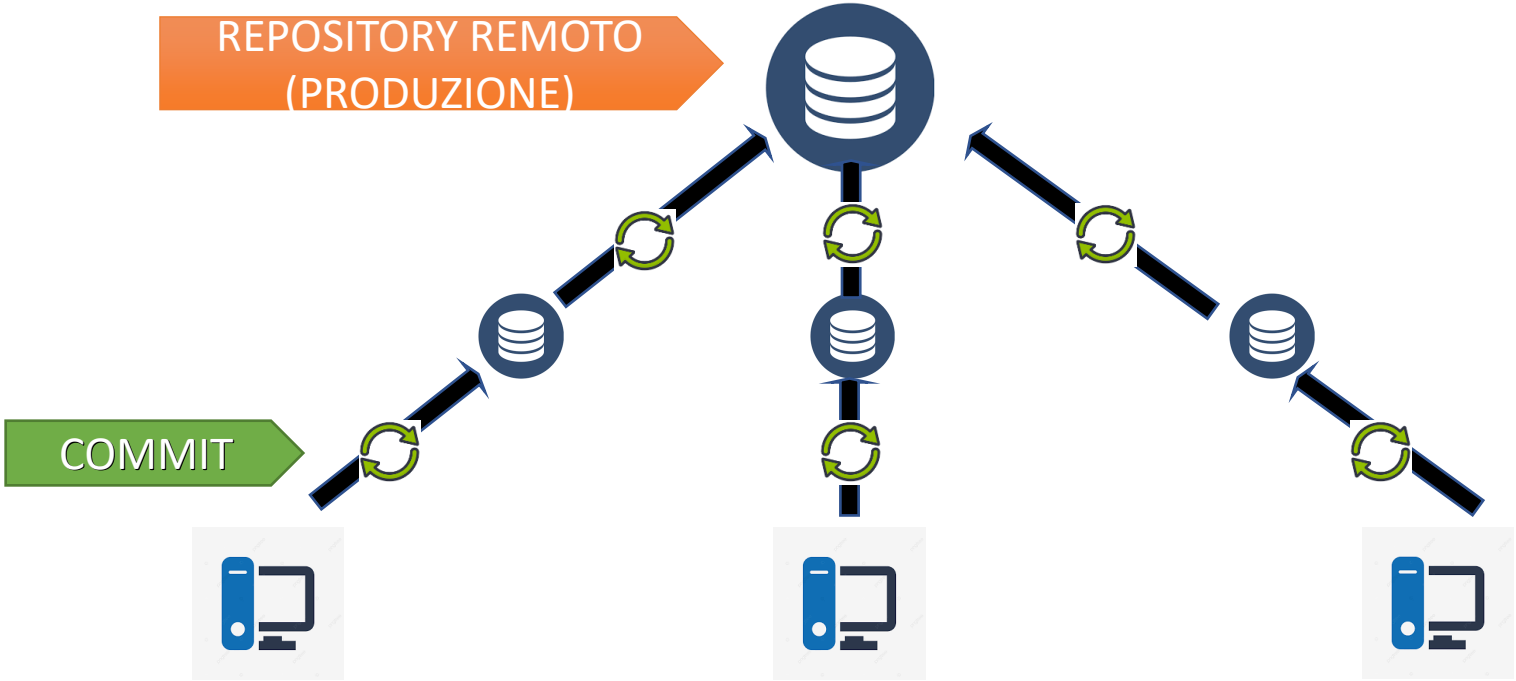
Fondamenti di Version Control - What?



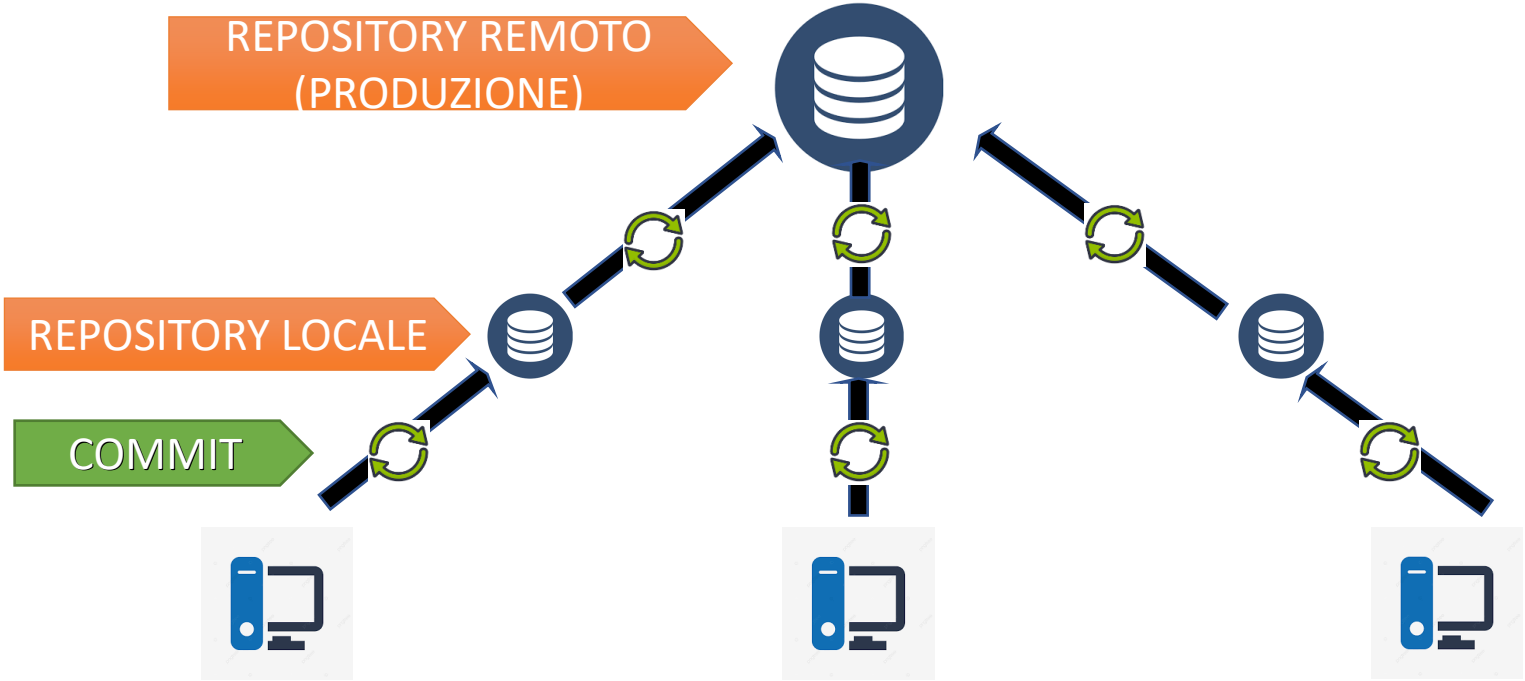
Fondamenti di Version Control - What?

- Ogni PC si è sincronizzato col repository remoto per confrontare la propria versione con quella remota e apportare le modifiche, appunto, direttamente in remoto (COMMITTARE)

Fondamenti di Version Control - What?



Fondamenti di Version Control - What?



Fondamenti di Version Control - What?

- Ogni PC ha un repository **LOCALE** che viene generato **CLONANDO** il repository remoto (**PRODUZIONE**)

Fondamenti di Version Control - What?

- Ogni PC ha un repository **LOCALE** che viene generato **CLONANDO** il repository remoto (**PRODUZIONE**)
- Le proprie modifiche vengono verificate e **COMMITTATE** sul repository **LOCALE**

Fondamenti di Version Control - What?

- Ogni PC ha un repository **LOCALE** che viene generato **CLONANDO** il repository remoto (**PRODUZIONE**)
- Le proprie modifiche vengono verificate e **COMMITTATE** sul repository **LOCALE**
- A questo punto, la situazione in **LOCALE** è definita, bisogna portare le modifiche sul repository remoto (**PRODUZIONE**)

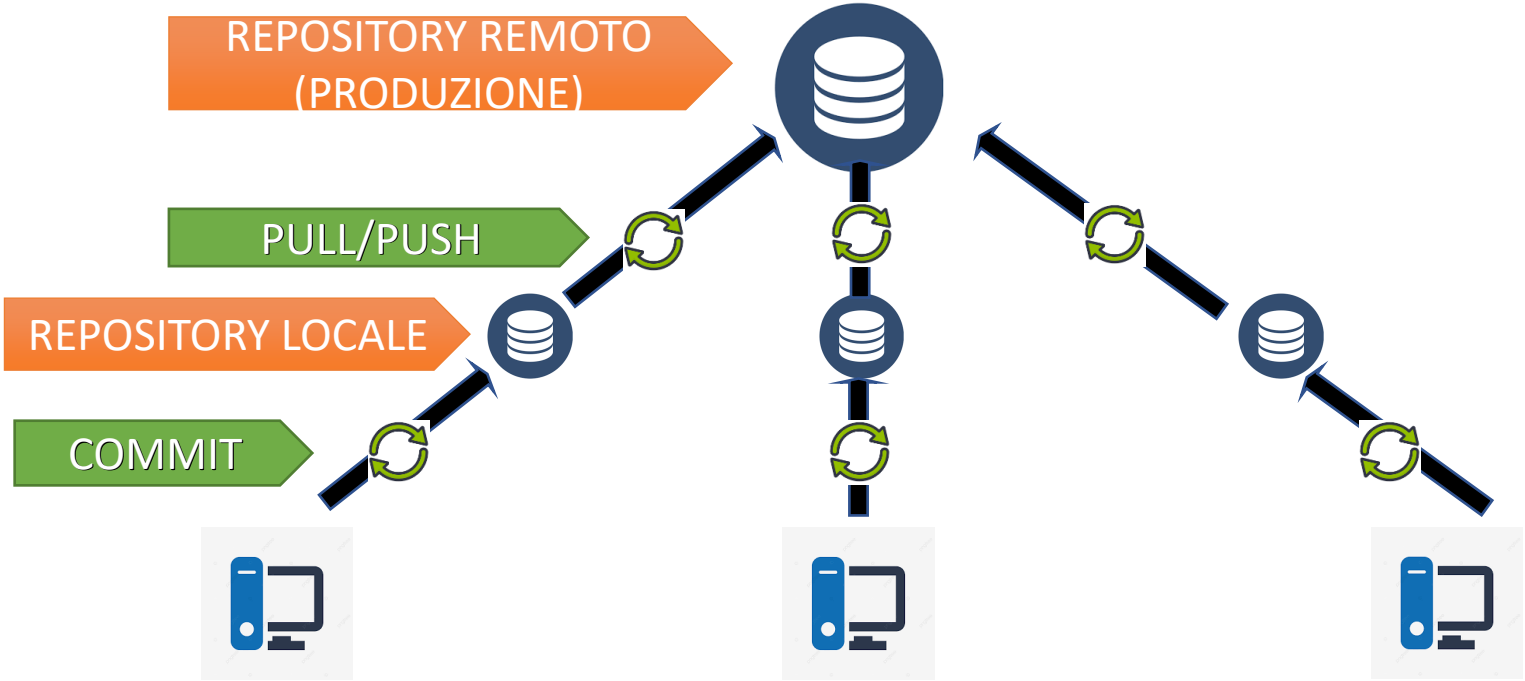
Fondamenti di Version Control - What?

- Ogni PC ha un repository **LOCALE** che viene generato **CLONANDO** il repository remoto (**PRODUZIONE**)
- Le proprie modifiche vengono verificate e **COMMITTATE** sul repository **LOCALE**
- A questo punto, la situazione in **LOCALE** è definita, bisogna portare le modifiche sul repository remoto (**PRODUZIONE**)
- L'operazione di "riversare" il repository **LOCALE** con quello **REMOTO** viene detta **PUSH**

Fondamenti di Version Control - What?

- Ogni PC ha un repository **LOCALE** che viene generato **CLONANDO** il repository remoto (**PRODUZIONE**)
- Le proprie modifiche vengono verificate e **COMMITTATE** sul repository **LOCALE**
- A questo punto, la situazione in **LOCALE** è definita, bisogna portare le modifiche sul repository remoto (**PRODUZIONE**)
- L'operazione di "riversare" il repository **LOCALE** con quello **REMOTO** viene detta **PUSH**
- Può (anzi DEVE, ma ne parleremo più avanti) essere effettuata anche l'operazione inversa, cioè "riversare" il repository **REMOTO** in quello **LOCALE**; questa operazione viene detta **PULL**

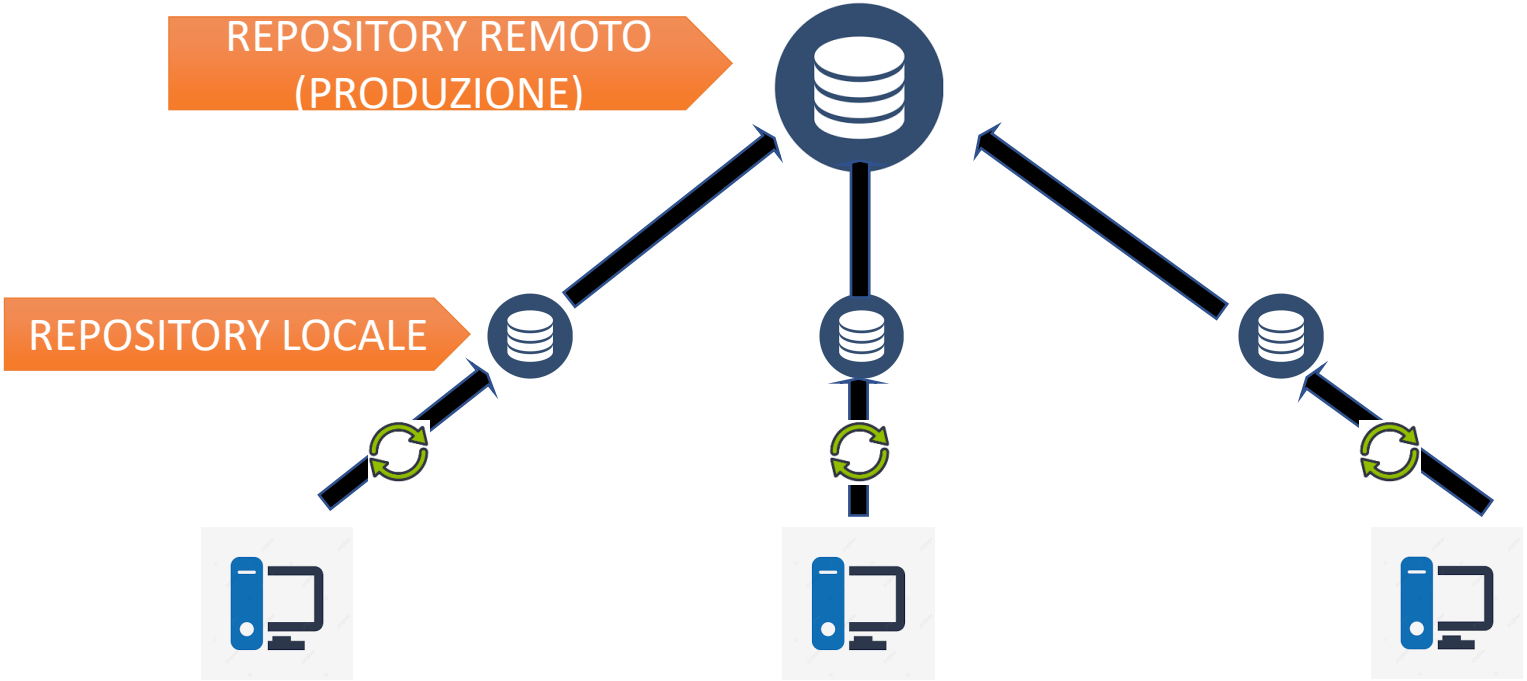
Fondamenti di Version Control - What?



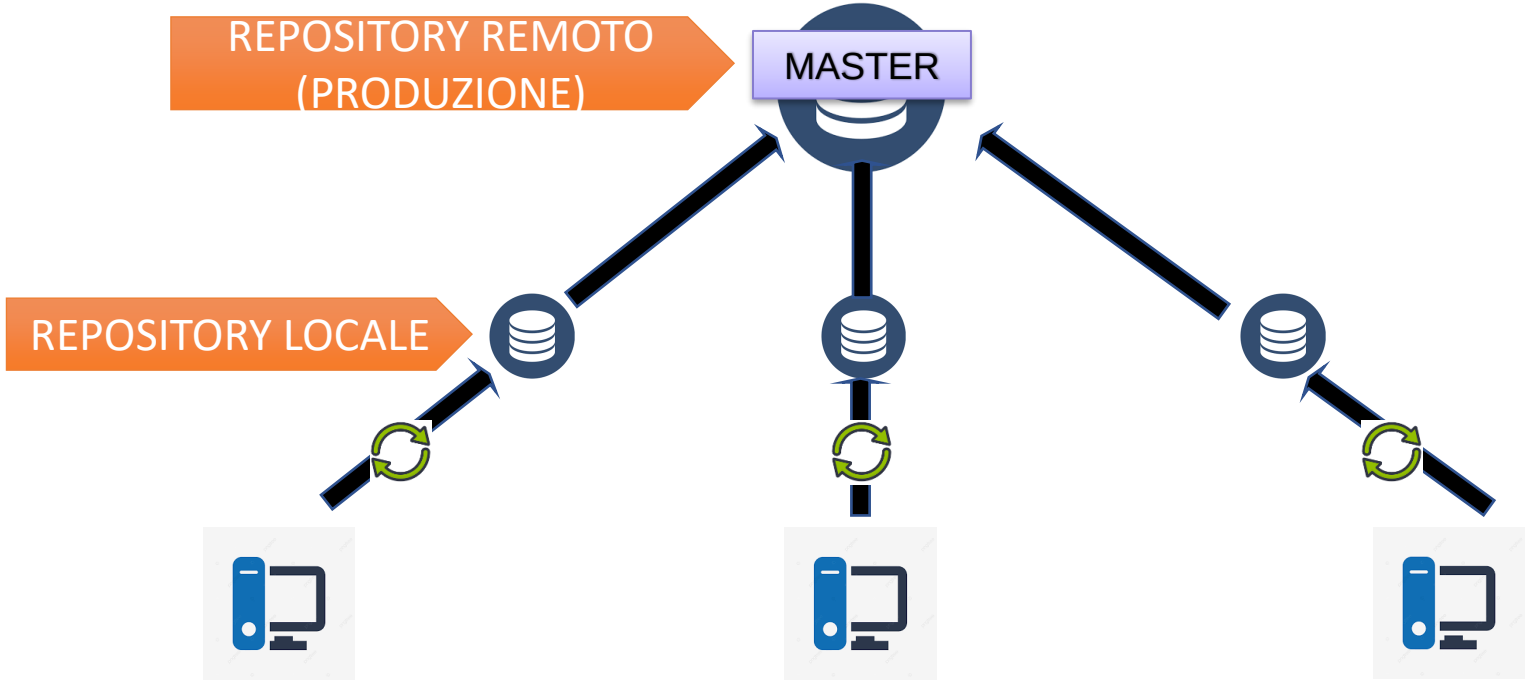
Fondamenti di Version Control - What?

- **... e allora perché Git???** (aridaje)

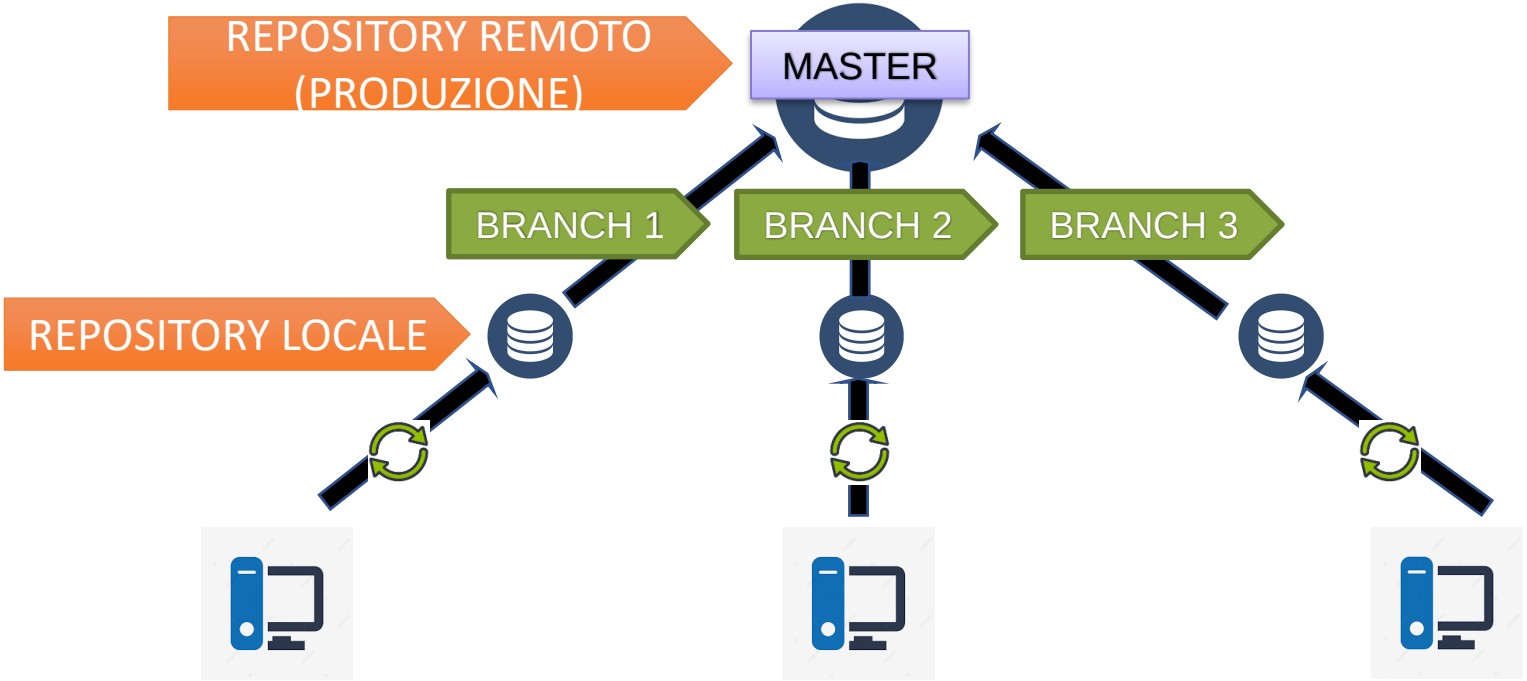
Fondamenti di Version Control - What?



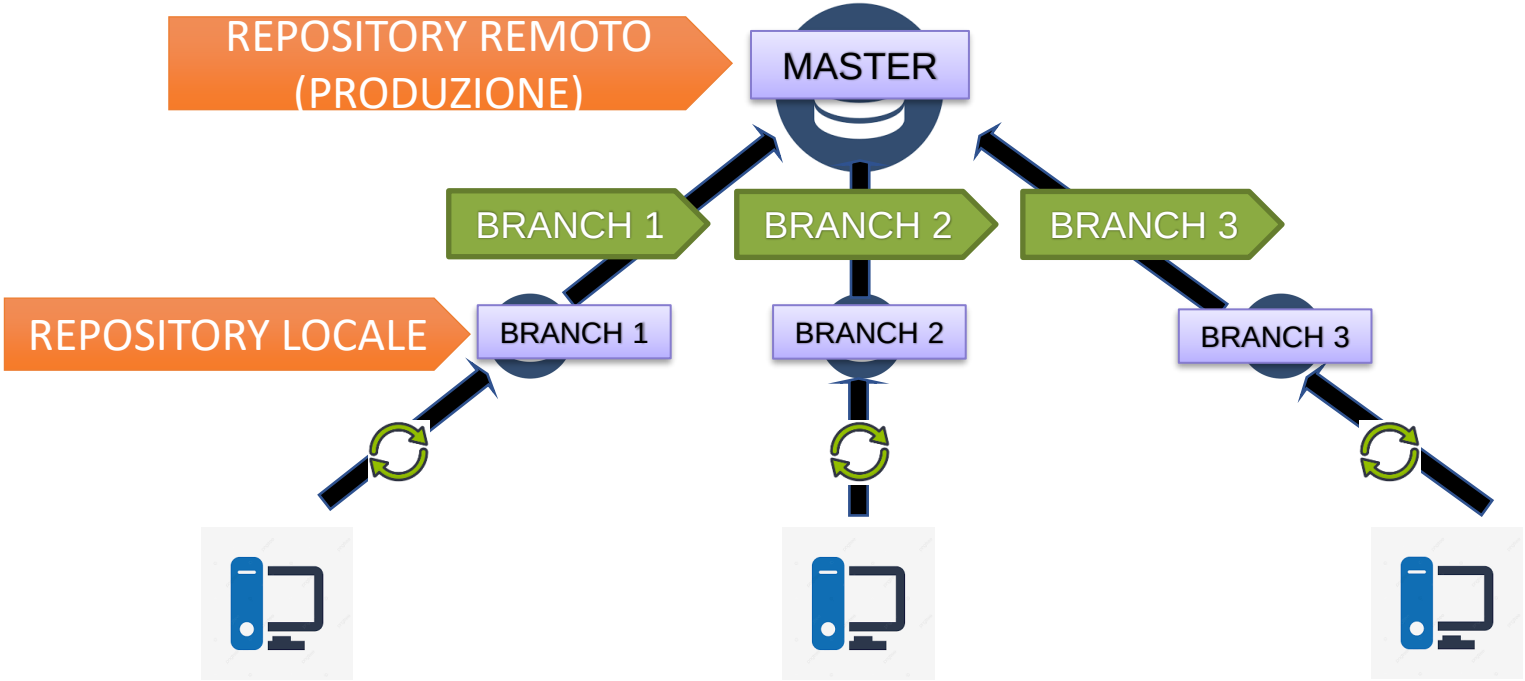
Fondamenti di Version Control - What?



Fondamenti di Version Control - What?

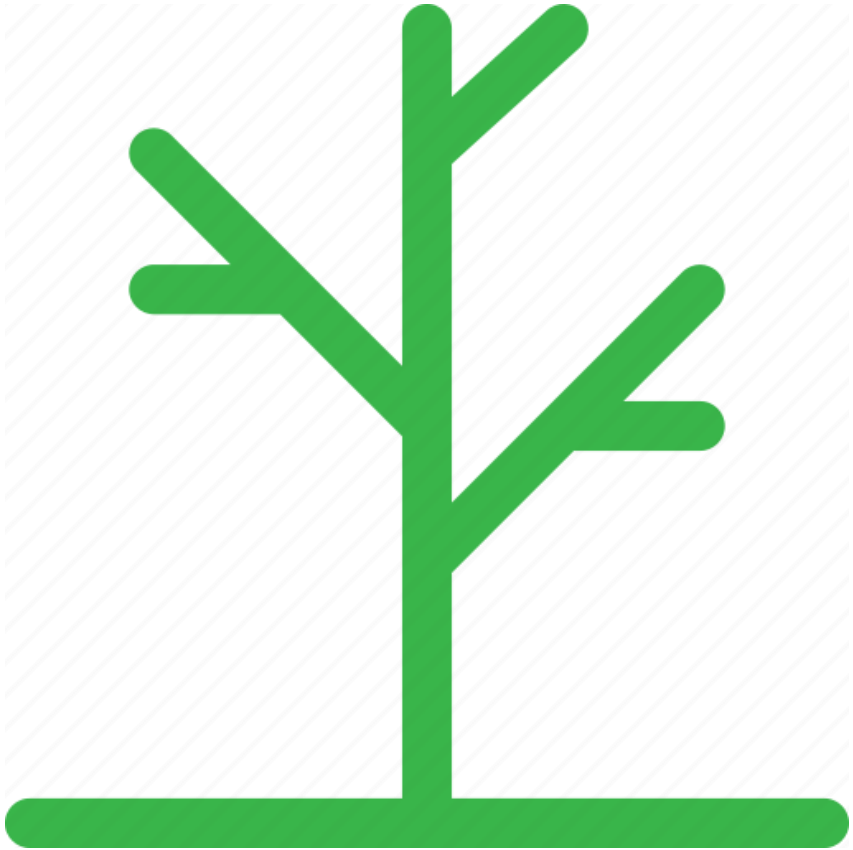


Fondamenti di Version Control - What?



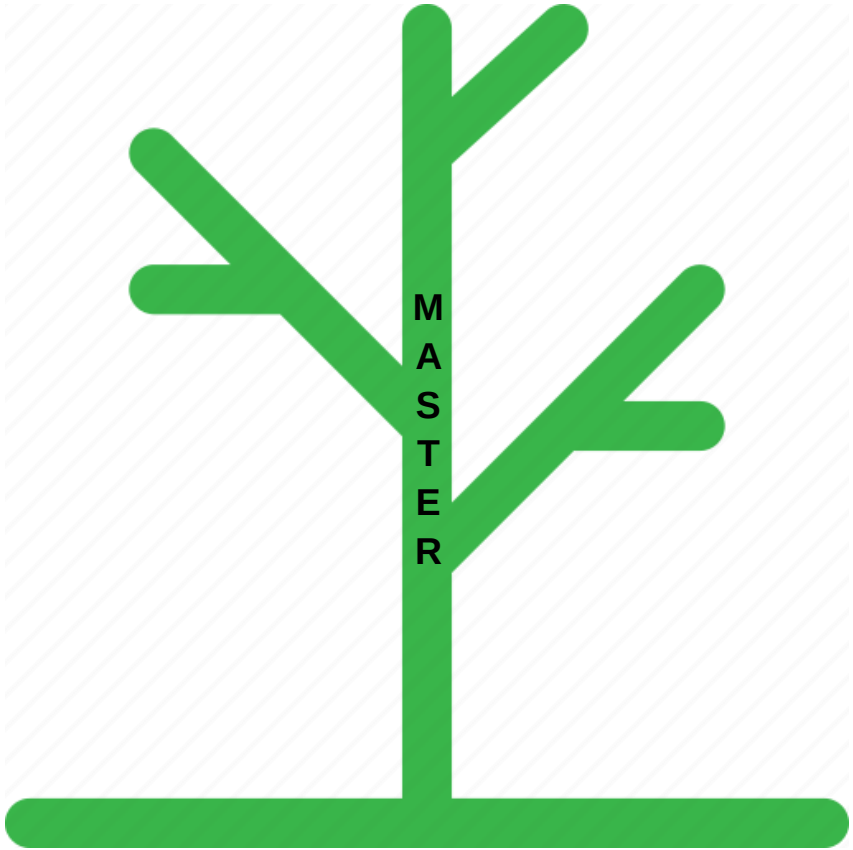
Fondamenti di Version Control - What?

- ... e allora perché Git???
- Git introduce il concetto dei branches (letteralmente "rami") che permettono di creare delle ramificazioni del progetto



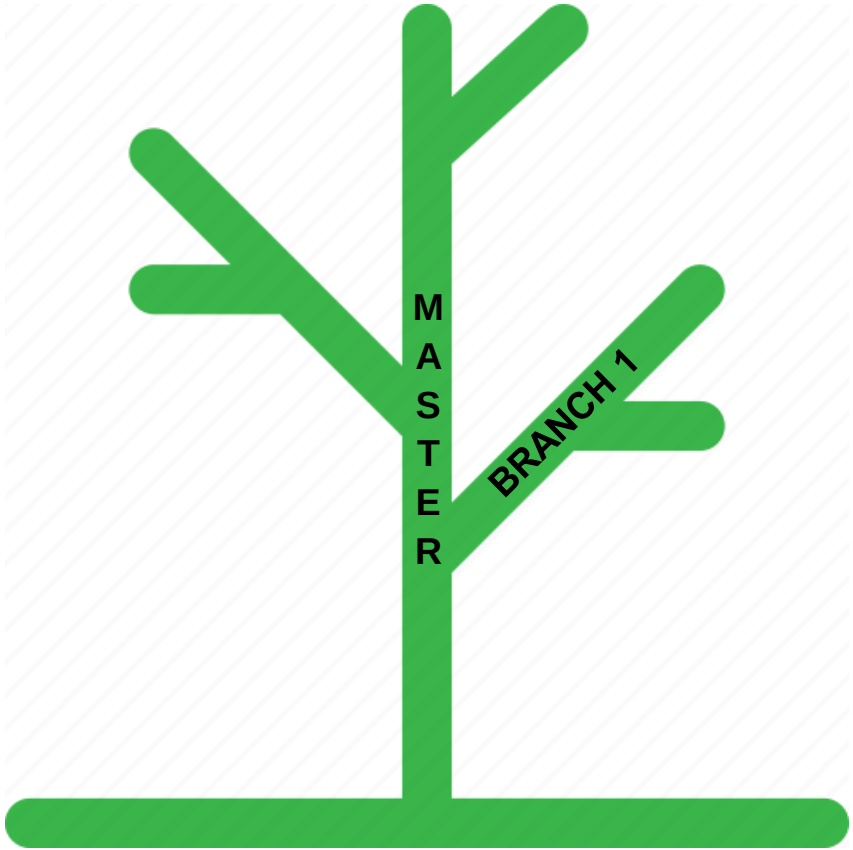
ITS TECNOLOGIE DELL'INFORMAZIONE E DELLA COMUNICAZIONE

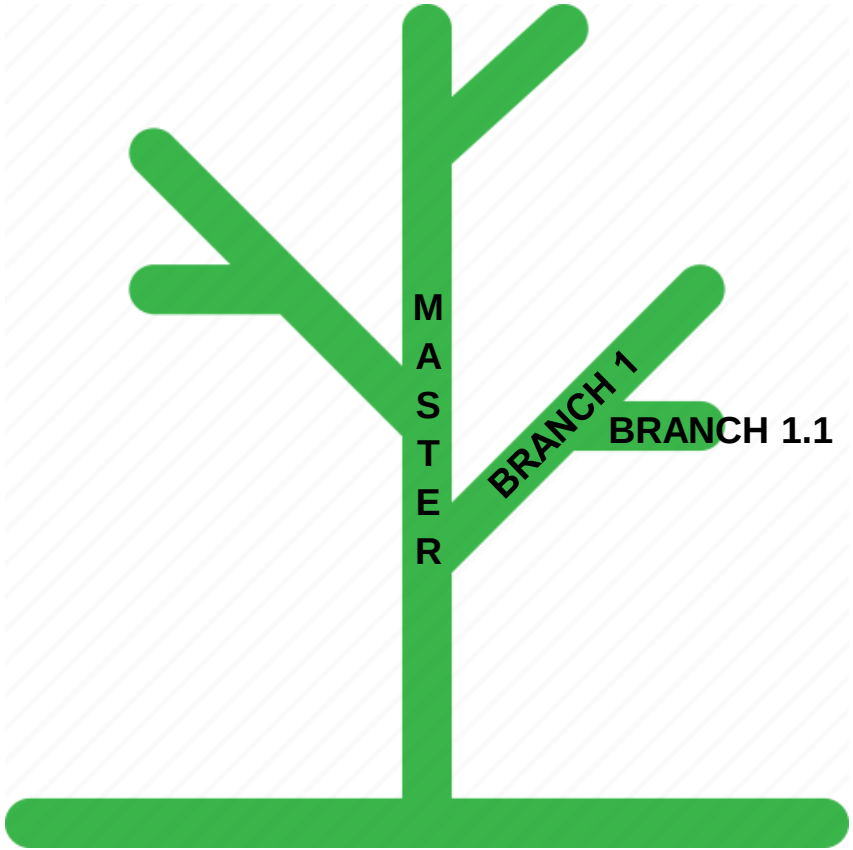
POR Piemonte FSE 2014-2020

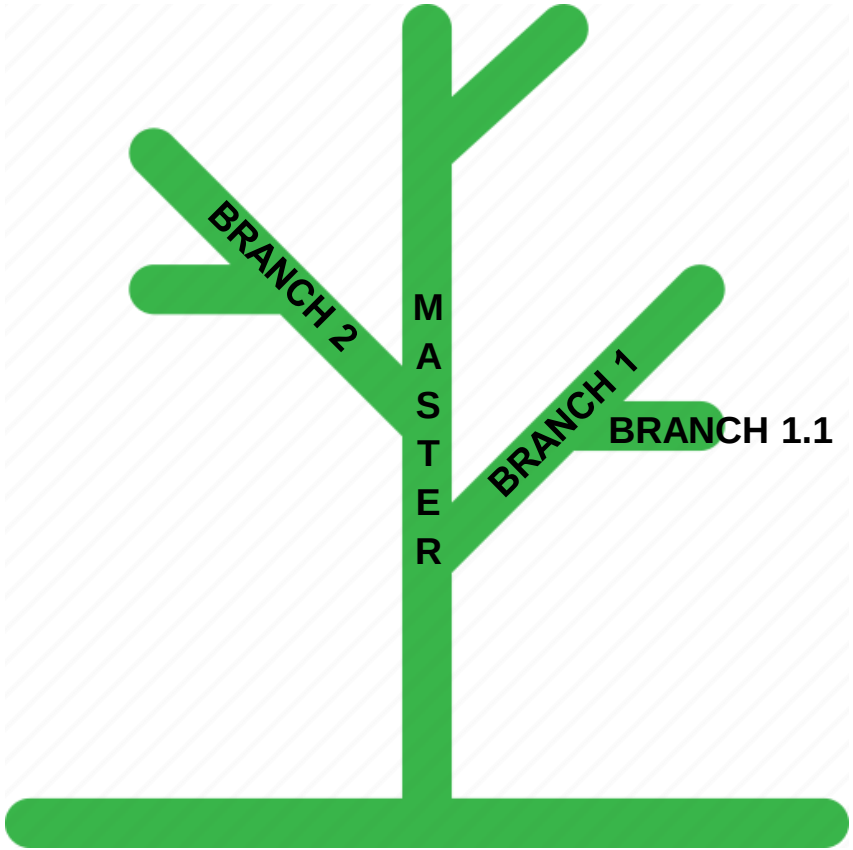


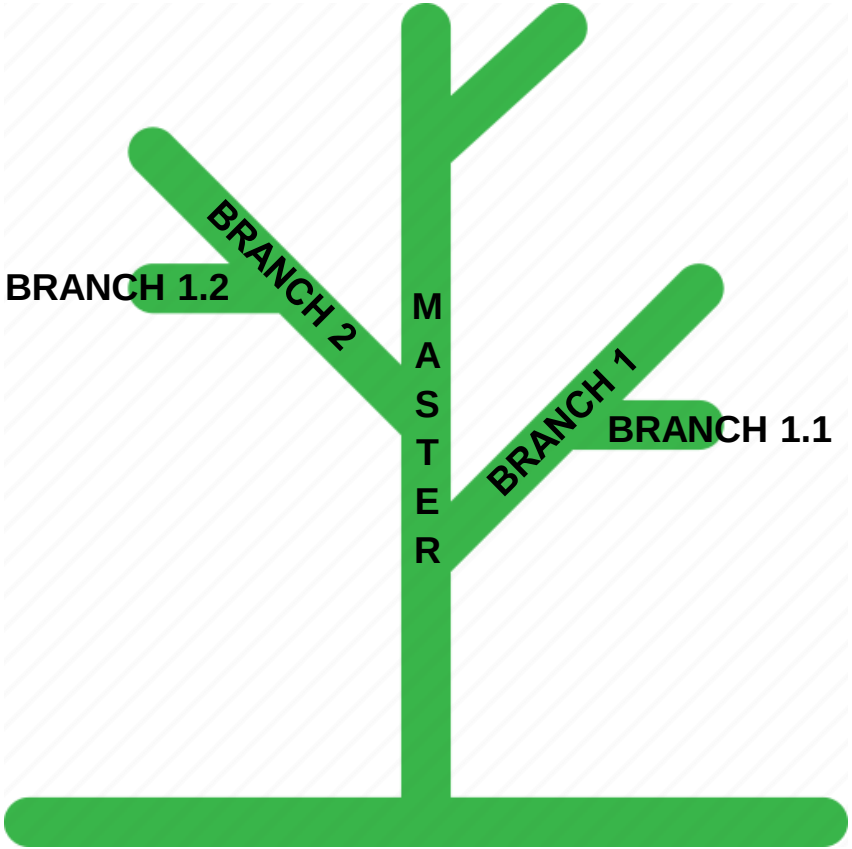
ITS TECNOLOGIE DELL'INFORMAZIONE E DELLA COMUNICAZIONE

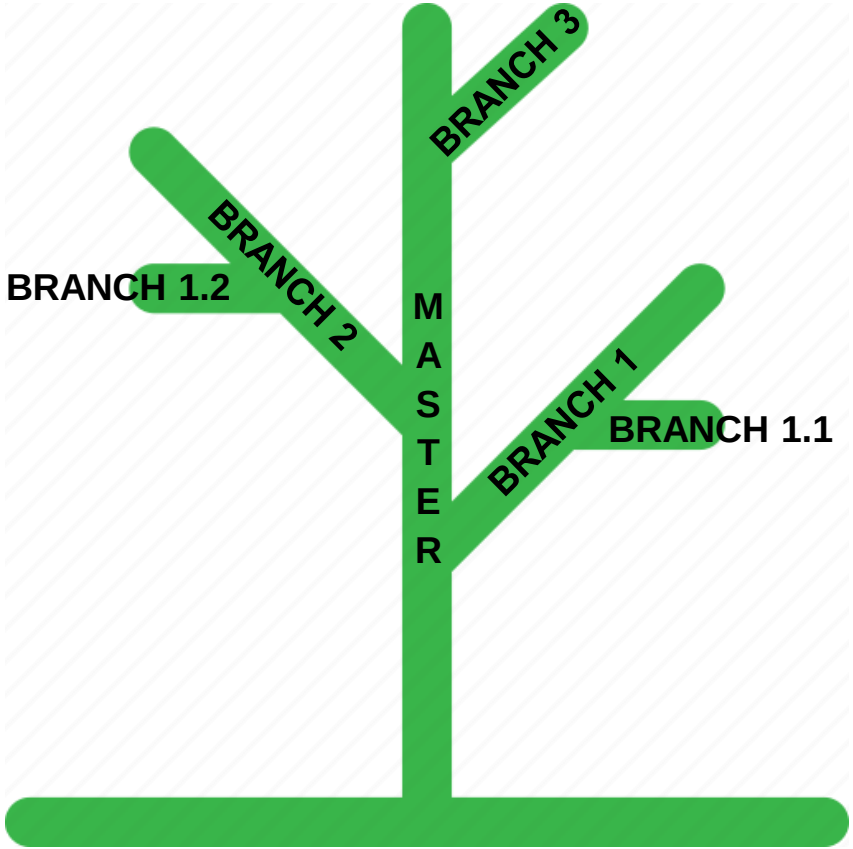
POR Piemonte FSE 2014-2020


MASTER
BRANCH 1


MASTER
BRANCH 1
BRANCH 1.1


BRANCH 2
MASTER
BRANCH 1
BRANCH 1.1


BRANCH 1.2
BRANCH 2
MASTER
BRANCH 1
BRANCH 1.1


BRANCH 3
BRANCH 2
BRANCH 1.2
MASTER
BRANCH 1
BRANCH 1.1

Fondamenti di Version Control - What?

- ... e allora perché Git???
- Git introduce il concetto dei branches (letteralmente "rami") che permettono di creare delle ramificazioni del progetto
- La struttura va vista proprio come un albero

Fondamenti di Version Control - What?

- ... e allora perché Git???
- Git introduce il concetto dei branches (letteralmente "rami") che permettono di creare delle ramificazioni del progetto
- La struttura va vista proprio come un albero
    - Le radici sono il master

Fondamenti di Version Control - What?

- ... e allora perché Git???
- Git introduce il concetto dei branches (letteralmente "rami") che permettono di creare delle ramificazioni del progetto
- La struttura va vista proprio come un albero
  - Le radici sono il master
  - Dal master si diramano i branches, 1 per ogni sviluppo (sviluppatore) diverso

Fondamenti di Version Control - What?

- ... e allora perché Git???
- Git introduce il concetto dei branches (letteralmente "rami") che permettono di creare delle ramificazioni del progetto
- La struttura va vista proprio come un albero
  - Le radici sono il master
  - Dal master si diramano i branches, 1 per ogni sviluppo (sviluppatore) diverso
  - Capita che si crei un branch da un branch e non da master, ma non è la norma

Fondamenti di Version Control - What?

Breve glossario:

- **Repository Remoto**: "contenitore ufficiale" del codice del progetto, normalmente situato su un server aziendale o remoto
- **Repository Locale**: "contenitore ufficioso" del codice del progetto, esiste sulla macchina su cui si sviluppa e viene normalmente creato clonando (vedi sotto) dal repository remoto
- **Clone**: operazione con cui viene creato un repository locale facendo una sorta di "copia-incolla" dal repository remoto; questo permette di avere il collegamento tra repository locale e remoto senza dover fare altri passaggi (e quindi poter fare push, pull, eccetera)
- **Branch**: letteralmente "ramo", in effetti è una diramazione del master. Se si immagina il master come il tronco di un albero, ogni branch diventa un ramo che parte da quel tronco e si sviluppa

ITS TECNOLOGIE DELL'INFORMAZIONE E DELLA COMUNICAZIONE

GRAZIE!

POR Piemonte
FSE 2014-2020