

# Verifica pratica Git e GitHub

---

## ISTRUZIONI

### Screenshot richiesti

Produrre screenshot da **Git Bash** ove indicato

Ogni screenshot deve mostrare: - prompt del terminale - percorso completo con directory CognomeNome - comando digitato e output

### Consegna screenshot

1. Inserire tutti gli screenshot, via via richiesti, in **UN SOLO file Word indicando i punti a cui fanno riferimento e il nome indicato**
  2. Convertire il file in **PDF**
  3. Salvare come:  
verifica\_git\_CognomeNome.pdf
  4. **(IMPORTANTE)** Inserire il PDF **dentro il repository** prima del push
- 

## CONSEGNA:

### 1 Init

1. Effettuare il **fork** del repository indicato in consegna su campus
  2. Clonare **il proprio fork** in locale tramite Git Bash (creare la cartella cognome\_nome\_verifica-gpi).
    1. fare screenshot della finestra git-bash dell'output del comando usato ed incollarlo nel file word. Scrivere prima dell'immagine ES.1.
- 

### 2 Creazione directory personale

All'interno del repository creare **OBBLIGATORIAMENTE** una directory con:

CognomeNome

Esempio:

RossiMario

Entrare nella directory appena creata (il percorso deve essere visibile nel terminale).

---

### 3 Creazione file HTML (testo OBBLIGATORIO) e primo commit

Creare il file `index.html` **copiando ESATTAMENTE** il seguente codice (si può usare il comando Linux `nano` seguito dal *nome file* o altro comando di vostra conoscenza) :

```
<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <title>Verifica Git</title>
</head>
<body>
    <h1>Verifica Git e GitHub</h1>
    <p>Questo file serve per testare i comandi Git.</p>
</body>
</html>
```

Eseguire nell'ordine:

- 1- Comando git che mostra lo stato dei file nell'indice
2. fare screenshot della finestra git bash con output del comando ed incollarlo nel file word. Scrivere prima dell'immagine ES.2.
3. Eseguire il commit del file `index.html` con messaggio “Creato file HTML iniziale – by Nome”  
3.1 fare screenshot della finestra git bash con output del comando ed incollarlo nel file word. Scrivere prima dell'immagine ES.3.

2- \_\_\_\_\_

### 4 Creazione branch personale

1. Creare un branch con nome:

`branch-Cognome`

Esempio:

`branch-Rossi`

2. Spostarsi sul branch appena creato  
2.1 fare screenshot unico della finestra git bash con output dei comandi lanciati ed incollarlo nel file word. Scrivere prima dell'immagine ES.4.

### 5 Modifica HTML sul branch e uso di git diff

Sul **branch personale**, modificare `index.html` come segue:

- aggiungere **una lista non ordinata** con almeno 3 elementi
- aggiungere un secondo paragrafo con il proprio **nome e cognome**

Esempio di modifica:

```
<ul>
  <li>Git</li>
  <li>Branch</li>
  <li>Merge</li>
</ul>
<p>Studente: Rossi Mario</p>
```

1 - Lanciare il comando git che mostri le modifiche al file, le differenze dal precedente salvataggio.

- 1.1 fare screenshot unico della finestra git bash con output del comando lanciato ed incollarlo nel file word. Scrivere prima dell'immagine ES.5.1.
  - 1.2 – eseguire commit del file index.html con messaggio “Modifiche HTML su branch personale by Cognome Nome”. Fare screenshot unico della finestra git bash con output del comando lanciato ed incollarlo nel file word. Scrivere prima dell'immagine ES.5.2.
- 

## 6 Merge

1. Tornare sul branch main (o eventualmente master)
  2. Eseguire il **merge del branch personale** (possibilmente con parametro *-no-ff*)
    - 2.1 fare screenshot unico della finestra git bash con output dei comandi lanciati ed incollarlo nel file word. Scrivere prima dell'immagine ES.6.
- 

## 7 Creazione tag (annotato)

Dopo il merge sul branch main:

1. Creare un **tag annotato** con nome:

v1.0-CognomeNome

e annotazione

“Effettuato merge tra branch main e personale”

Esempio:

v1.0-RossiMario

2. Verificare la presenza del tag, **lanciando il comando che mostri anche l'annotazione**
    - 2.1 fare screenshot unico della finestra git bash con output del/i comando/i lanciato/i ed incollarlo nel file word. Scrivere prima dell'immagine ES.7.
-

**8 Push su GitHub**

1. Pushare **tutti i branch** sul proprio repository GitHub
  2. Pushare anche il **tag**  
2.1 fare screenshot unico della finestra git bash con output dei comandi lanciati ed incollarlo nel file word. Scrivere prima dell'immagine ES.8.
- 

**9 Pull Request (CONSEGNA)**

1. Aprire **UNA Pull Request** verso il repository del docente
2. Titolo:  
Cognome Nome - Pentamestre - Verifica 1
3. Descrizione: > “Verifica GIT - LAB”

**⚠️ La consegna è valida solo se la Pull Request è visibile nel repository del docente.**

---

**Nota finale**

“In questa verifica viene valutato il processo di lavoro, non solo il risultato finale.”