

# Università Degli Studi di Ferrara

Corso di Laurea in Informatica - A.A. 2023 - 2024

# Tecnologie Web

Lez. 03 - Git e Bitbucket

# In questa lezione...

- VCS
- Git
- Bitbucket

# VCS

VCS è un acronimo che sta per Version Control System, ovvero:

## **Sistema di Controllo di Versione**

Un VCS è una combinazione di tecnologie e procedure utilizzate per tenere traccia nel tempo delle modifiche fatte ad un file o ad un insieme di file.

# VCS

I VCS, detti anche sistemi di versionamento, vengono utilizzati da un singolo sviluppatore per tenere traccia delle modifiche apportate al proprio software e per recuperare una vecchia versione di un determinato (o gruppo di) file.

Ad esempio, se uno sviluppatore si rende conto che le sue ultime modifiche hanno introdotto un malfunzionamento nel software, può decidere di annullare tutte le ultime modifiche ripristinando una versione precedente del software.

# VCS

Per un singolo sviluppatore, utilizzare un sistema di controllo di versione è utile per organizzare lo sviluppo di un progetto in evoluzione.

Per questo scopo, è sufficiente avere installato il sistema di controllo di versione sul proprio computer.

# git

**git** è un moderno sistema di controllo di versione distribuito, Open Source e quindi gratuito, progettato per gestire dati in modo veloce ed efficiente.

Il suo sviluppo è stato iniziato da Linus Torvalds per gestire lo sviluppo del kernel di Linux.



# Configurazione di git

Il primo passo da compiere è scaricare ed installare **git** dal sito ufficiale, se necessario.

<https://git-scm.com/downloads>

Una volta installato **git**, è necessario eseguire una piccola configurazione.

**git** ha una interfaccia da linea di comando (CLI) è quindi necessario scrivere i comandi sul **terminale**:

```
git config --global user.name "Nome Cognome"  
git config --global user.email "email@edu.unife.it"  
git config --global core.editor "gedit"
```

# Dizionario

## **Working Directory**

Si intende la directory in cui sono presenti tutti i file su cui stiamo lavorando. È quella directory che di solito contiene i file del nostro progetto.

## **Repository**

È l'insieme di tutte le versioni dei nostri file. È gestito dal sistema di versionamento ed è solitamente “invisibile” all'utente (nel nostro caso allo sviluppatore).

## **Registrazione**

Si intende la creazione di una nuova versione di uno o più file all'interno del repository.



# Creazione di un repository

Una volta installato e configurato **git**, per iniziare ad utilizzare il VCS è necessario creare un repository per il progetto.

Attraverso i comandi del terminale ci posizioniamo nella Working directory e creiamo il repository per il nostro progetto con il comando:

```
git init
```

Da adesso in poi, possiamo utilizzare i comandi di **git** per gestire il nostro repository.

# Area di Stage

È importante tenere sempre presente la Working directory ed il Repository sono due entità separate e possono anche non coincidere.

Pensiamo ad esempio al seguente caso.

Si sta eseguendo una ricerca per raccogliere il valore delle aree di diversi rettangoli.

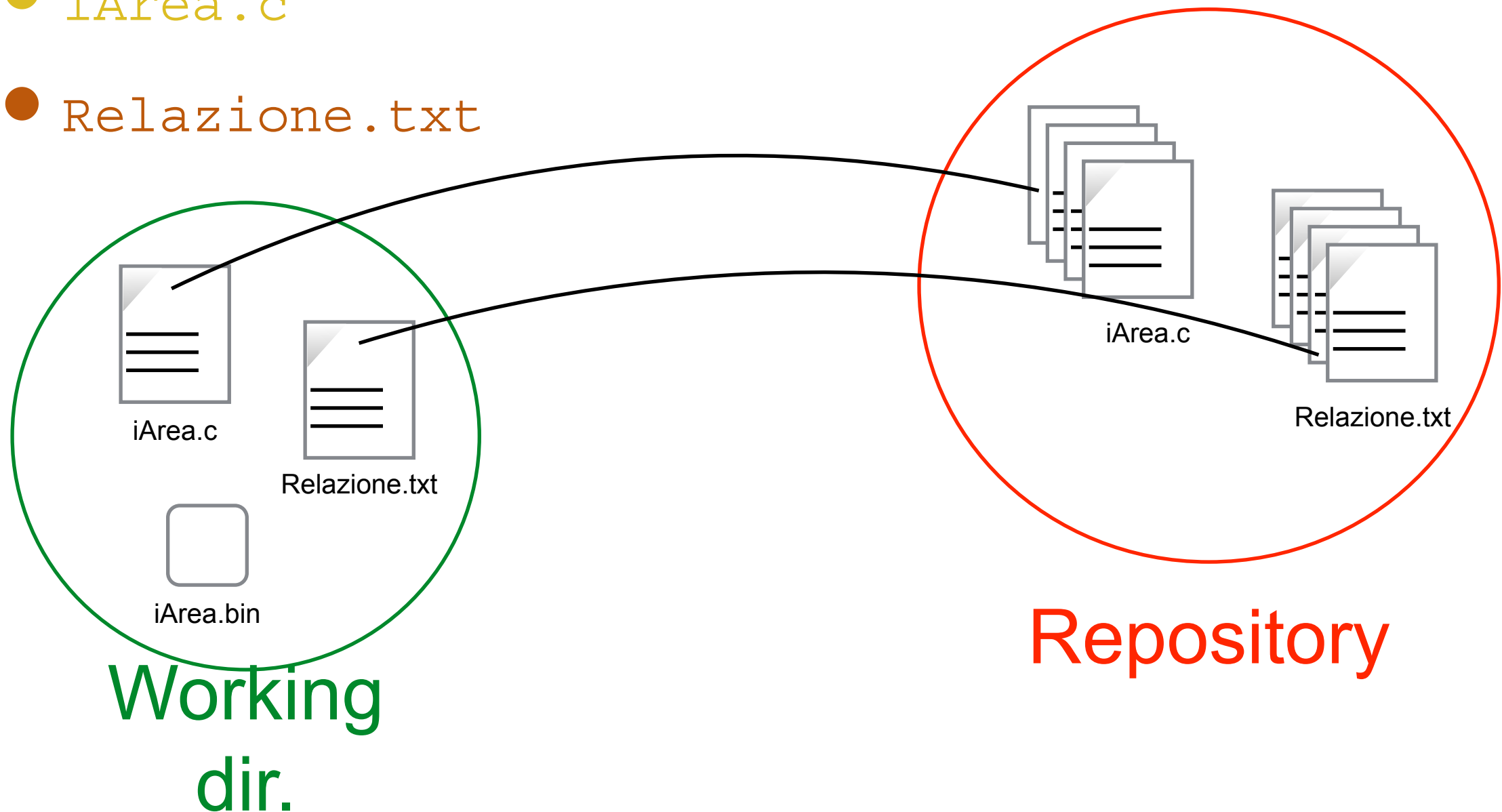
Nella mia W.D. avrò i seguenti file:

- `iArea.c` (file sorgente del mio codice)
- `iArea.bin` (file sorgente compilato)
- `Relazione.txt` (file di testo che contiene i risultati)

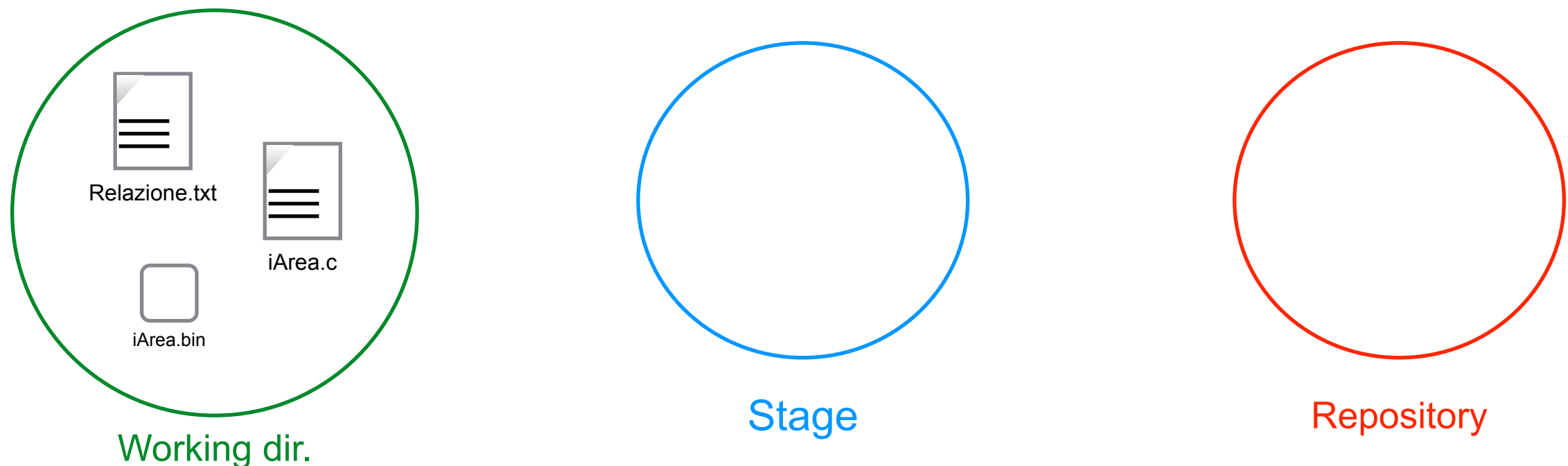
# Area di Stage

È ragionevole pensare (ed è prassi comune) che io voglia **tenere traccia dei cambiamenti** solamente dei file:

- `iArea.c`
- `Relazione.txt`

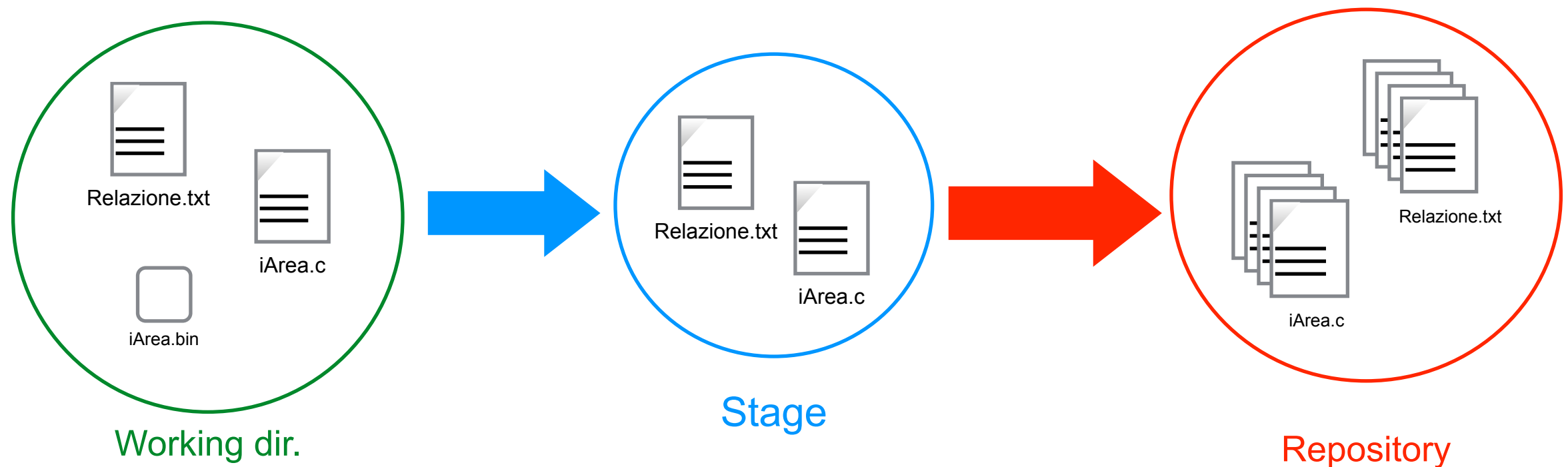


# Area di Stage



**git** prevede una terza area (detta “di Stage”) che raggruppa tutti i file pronti per essere registrati nel repository.

# Area di Stage



I file, di cui vogliamo tenere traccia delle versioni, devono essere **prima aggiunti** all'Area di Stage e **poi registrati** nel Repository

# Aree di lavoro

Working dir

La **Working directory** rappresenta la directory di lavoro, quella in cui sono presenti TUTTI i file su cui stiamo lavorando.

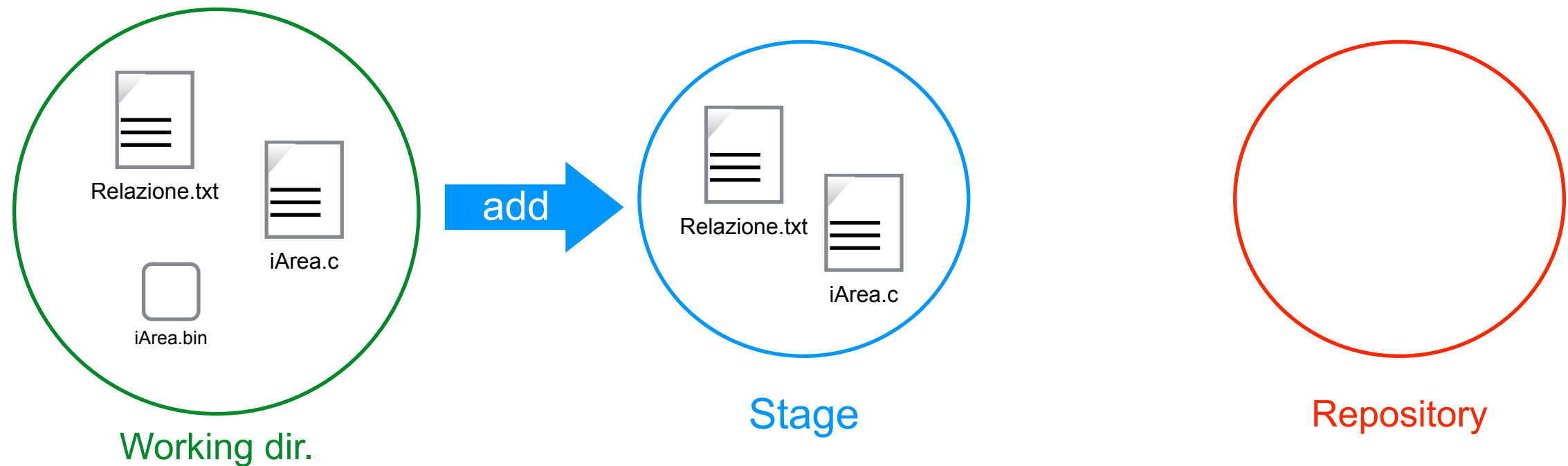
Stage

Lo **Stage** è l'insieme dei file tracciati e pronti per essere registrati nel repository.

Repository

Il **Repository** è l'insieme di tutte le versioni dei files che vengono tracciati con il VCS.

# Add



Per includere un nuovo file all'area di Stage, uso il comando:

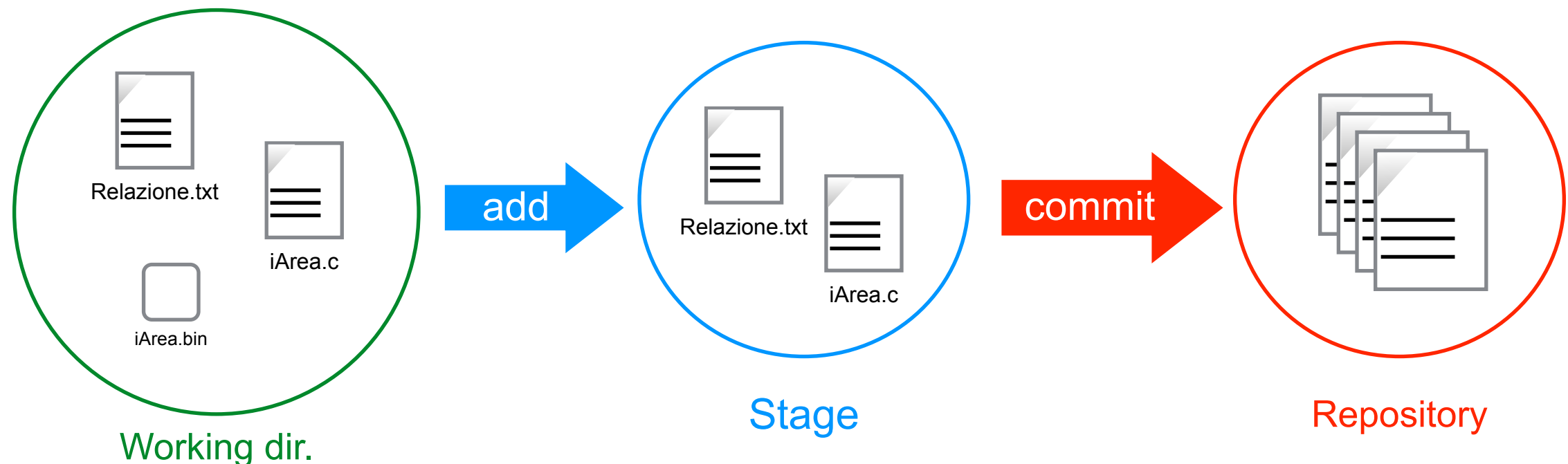
```
git add <nome_file>
```

Nel nostro esempio:

```
git add iArea.c
```

```
git add Relazione.txt
```

# Commit



Per effettuare una nuova registrazione dei file presenti all'area di Stage, uso il comando:

```
git commit
```

Allegando un messaggio che descrive i cambiamenti

Ad esempio:

```
git commit -m "Correzione della formula dell'area"
```



# Commit

Ad ogni nuovo commit (o registrazione) il VCS memorizza solamente i cambiamenti di ogni file rispetto alla loro ultima versione registrata ed associa ad ogni commit un identificativo univoco rendendo così possibile identificare ogni singola registrazione.

Attraverso il comando:

```
git log
```

Possiamo visualizzare l'elenco degli ultimi commit eseguiti e le informazioni sintetiche di ognuno, compreso l'identificatore.

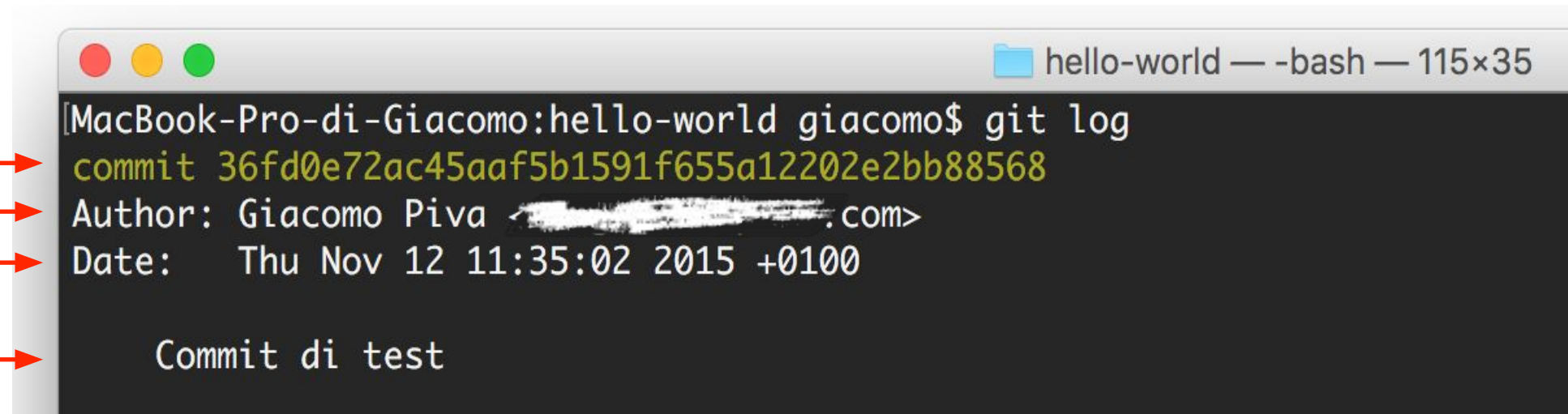
# Informazioni del Commit

Identificatore

Autore

Data

Messaggio

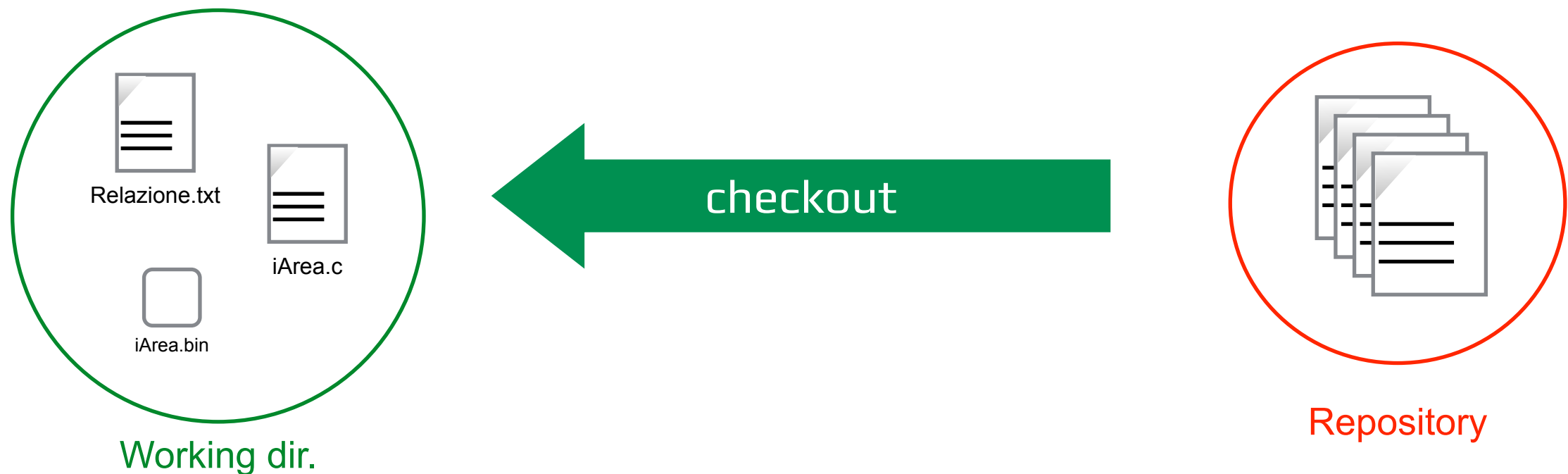


```
hello-world — -bash — 115x35
[MacBook-Pro-di-Giacomo:hello-world giacomo$ git log
commit 36fd0e72ac45aaf5b1591f655a12202e2bb88568
Author: Giacomo Piva <[REDACTED].com>
Date: Thu Nov 12 11:35:02 2015 +0100

    Commit di test
```

The image shows a terminal window with the output of the `git log` command. Four red arrows originate from labels on the left and point to specific lines in the terminal output: 'Identificatore' points to the commit hash, 'Autore' points to the author name, 'Data' points to the date, and 'Messaggio' points to the commit message.

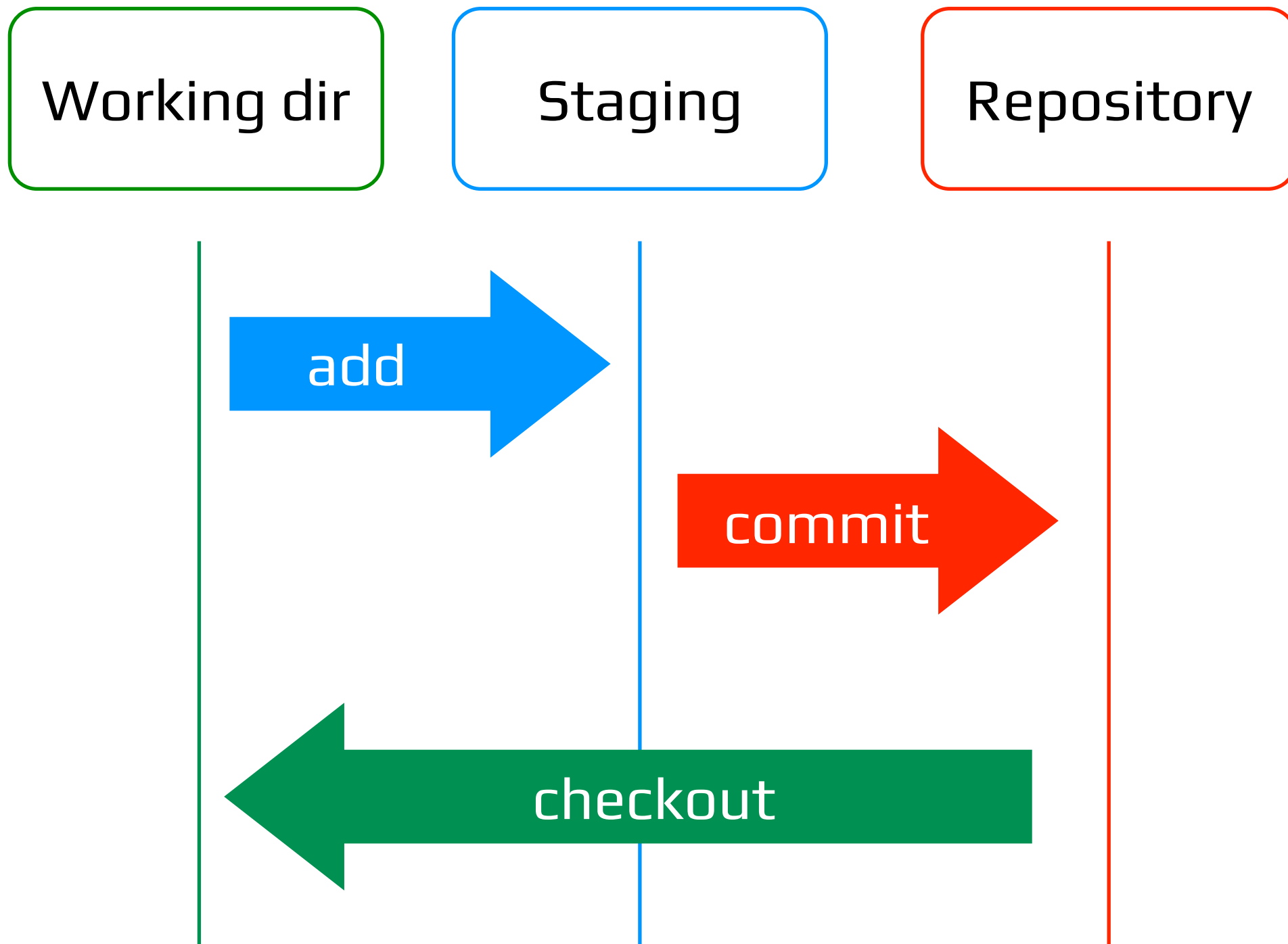
# Checkout



È possibile recuperare una precedente versione di un singolo file oppure della intera Working directory attraverso il comando:

```
git checkout <commit_id> [<nome_file>]
```

# Workflow di lavoro



# Status

In ogni momento, il comando:

```
git status
```

Ci da informazioni essenziali sullo stato del repository e della working directory, sui file modificati e non ancora registrati e sulla presenza di file che non stiamo tracciando.

# Esempio

Repository locale

# Commit

Si potrebbe, arrivati a questo punto, fare un'importante considerazione:

## **Quando eseguire una registrazione?**

L'ideale sarebbe mantenere i commit concentrati su “piccoli” step di avanzamento del progetto.

## **Cosa scrivere nel messaggio?**

Per mantenere i messaggi di commit più descrittivi possibile dell'evoluzione, è meglio descrivere il **Perché** delle modifiche invece del **Cosa** è stato modificato.

# VCS

I sistemi di controllo di versione sono diventati ormai strumenti indispensabili soprattutto per organizzare il codice prodotto da **gruppi di sviluppatori** che lavorano insieme ad uno stesso progetto.



# VCS

In questo caso, l'utilizzo di un sistema di controllo di versione diventa indispensabile per:

- Tenere traccia di chi, nel gruppo di sviluppo, ha effettuato una modifica su un file;
- Unire le modifiche di ogni membro del gruppo di sviluppo;

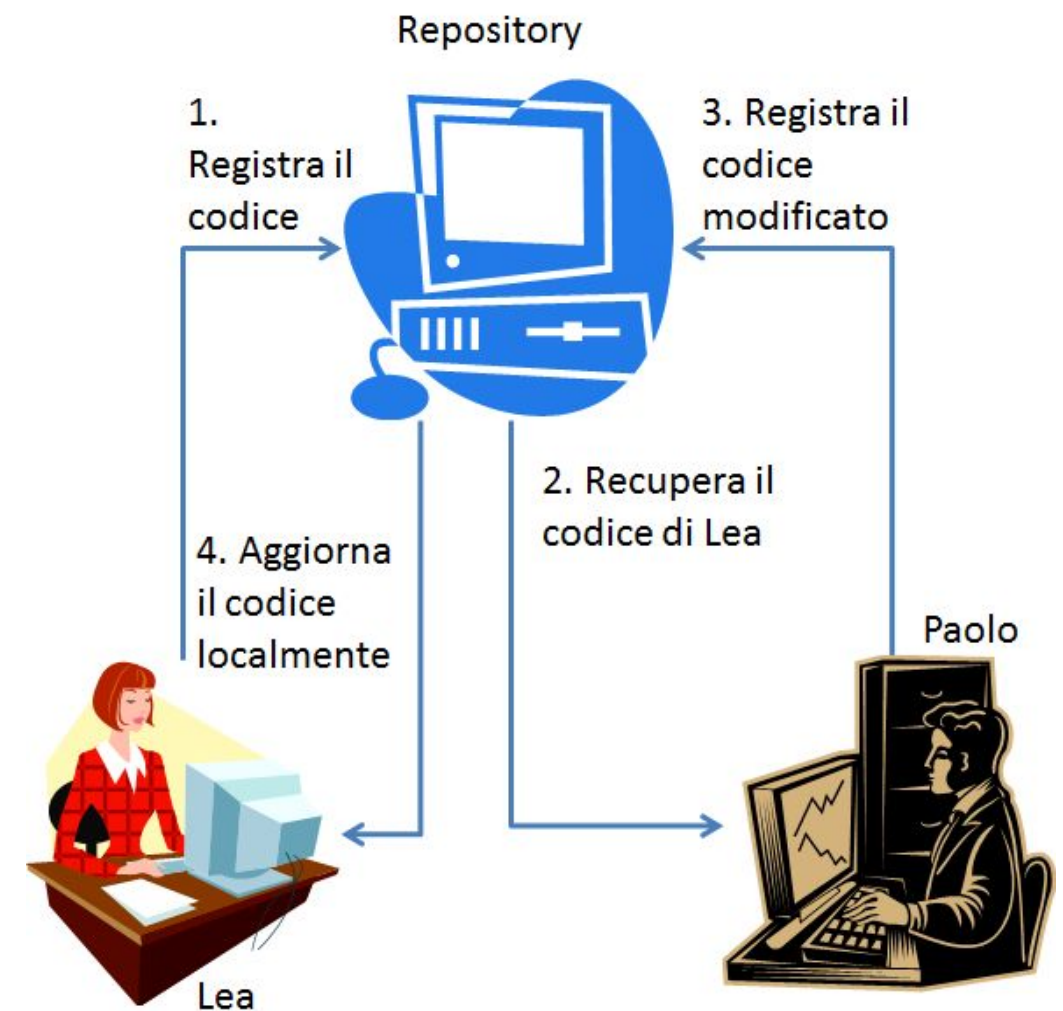
Per questi scopi, non basta più avere solamente il sistema di controllo di versione installato sul proprio computer, serve anche un “punto” centralizzato raggiungibile da tutti i membri del gruppo, che collezioni le modifiche di ogni sviluppatore e le renda disponibili a tutti gli altri.

# Come funziona un VCS

Lea e Paolo lavorano insieme a un progetto.

Ogni membro del gruppo ha una copia dell'area di lavoro sulla propria macchina chiamata **repository locale**

L'area di lavoro centralizzata (su un server) viene chiamata **repository remoto**



# Bitbucket

Bitbucket è un servizio online che offre soluzioni per la gestione di repository **git** remoti.

Bitbucket è un servizio a pagamento ma offre un piano gratuito per progetti che coinvolgono fino a 5 utenti.

Una volta registrati al servizio dal sito: <https://bitbucket.org>

Bitbucket vi assegnerà un nome utente che vi identifica all'interno del suo sistema e potrete creare repository remoti. Attraverso i comandi di **git** sarà possibile sincronizzare i vari repository locali con i corrispettivi repository remoti.



# Bitbucket

Ogni **repository remoto** è identificato, oltre che dal **nome** anche da un **URL** (che per definizione è univoco).

Bitbucket, per gli URL dei repository utilizza il seguente schema:

`https://nome-utente@bitbucket.org/nome-utente/nome-repository.git`

Dove:

- **nome-utente**: È l'utente proprietario del repository (chi lo ha creato)
- **nome-repository**: È appunto il nome del repository

# Bitbucket Dashboard

The screenshot shows the Bitbucket Dashboard interface. On the left is a blue sidebar with the Bitbucket logo and a search icon. Below these are navigation links: 'Your work' (selected), 'Repositories', 'Projects', 'Pull requests', 'Issues', and 'Snippets'. The main content area is titled 'Your work' and features a 'Repositories' section. This section has a 'Last updated' dropdown menu and lists two repositories: 'Tecnologie Web - Esempi AA16/17' by Giacomo Piva (last updated 2017-12-28) and 'portaleweb' by Porte Aperte (last updated 2017-12-07). A 'View all repositories' link is at the bottom of the list.

Bitbucket

Search

+ Your work

+ Repositories

+ Projects

+ Pull requests

+ Issues

+ Snippets

Your work

Repositories

Last updated ▾

php Tecnologie Web - Esempi AA16/17  
Giacomo Piva - Last updated 2017-12-28

php portaleweb  
Porte Aperte - Last updated 2017-12-07

[View all repositories](#)

Dalla Sidebar è possibile selezionare:

- Gli ultimi repository utilizzati
- Elenco dei repository
- Elenco dei progetti

# Bitbucket Dashboard

Cliccando sul “ + ” nella Sidebar, si viene portati alla scelta di quale entità creare.



CREATE A NEW



**Repository**



**Team**



**Project**

Team:

Insieme di utenti che collaborano a progetti e repository

Progetti:

Insieme di repository

Repository:


Singolo repository

# Bitbucket Nuovo Repository

## Create a new repository

[Import repository](#)

Owner

 giacomopiva

▼

Repository name\*

Access level

☒ This is a private repository

Include a README?

No

▼

Version control system

☒ Git

☐ Mercurial

▼ Advanced settings

Description

Forking

Allow only private forks

▼

Project management

☐ Issue tracking

☐ Wiki

Language

Select language...

▼

Integrations

☐ Enable Hipchat notifications

Create repository

Cancel




Nome nuovo repository


Descrizione repository


Linguaggio di programmazione





# Bitbucket Nuovo Repository





 **Tecnologie Web - Es...**


 Overview


 Source


 Commits


 Branches

 Pull requests

 Pipelines



 Downloads

 Boards

 Settings

Giacomo Piva / **Tecnologie Web - Esempi AA16/17**

## Overview



 **HTTPS**  <https://giacomopiva@bitbucket.org/g>

Last updated 2017-12-28


Language PHP


Access level Admin


0	6
Open PRs	Watchers
1	0
Branch	Forks


 Share 

## Recent activity

 **1 commit**  
Pushed to giacomopiva/tecnologie-web-ese...  
[920e504](#) Completate azioni CRUD  
Giacomo Piva · 2017-06-08

 **1 commit**  
Pushed to giacomopiva/tecnologie-web-ese...  
[7c680b1](#) Esempio controlelr e view  
Giacomo Piva · 2017-06-06

 **1 commit**  
Pushed to giacomopiva/tecnologie-web-ese...  
[8889a38](#) Config iniziale e Modello  
Giacomo Piva · 2017-06-05

 Edit README

## README

Esempi per il corso di Tecnologie Web



# Bitbucket

In questo momento sappiamo come:

- Creare un repository locale (git init)
- Creare un repository remoto (usando Bitbucket)

Quello che manca ora è il collegamento fra i 2 repository.

Cioè, in che modo posso informare il VCS (**git**) che un dato repository locale deve essere sincronizzato con un preciso repository remoto?

# Bitbucket

Quale delle due azioni (creare il repository locale o remoto) svolgere prima è arbitrario e varia da caso a caso, pensiamo ad esempio ai due casi:

- Invito qualcuno a lavorare ad un progetto
- Mi unisco ad un gruppo che già lavora ad un progetto.

Esiste perciò la possibilità di collegare il nostro repository locale ad un repository remoto a seconda del nostro punto di partenza.

# Bitbucket

Se abbiamo già un repository locale e vogliamo collegarlo ad un repository remoto, possiamo farlo (una volta creato il repository su Bitbucket) con il comando:

```
git remote add origin <url_repository>
```

Il comando predisporrà il nostro repository locale per la sincronizzazione con il repository remoto identificato dall'URL\*.

\* È possibile che venga richiesto username e password di Bitbucket durante il processo.

# Bitbucket

Se abbiamo prima creato il repository remoto, possiamo crearne una copia locale attraverso il comando

```
git clone <url_repository>
```

Il comando creerà nella directory corrente una copia del repository remoto inizializzando anche il repository locale\*

\* Non sarà necessario inizializzare il repository locale con il comando `git init`

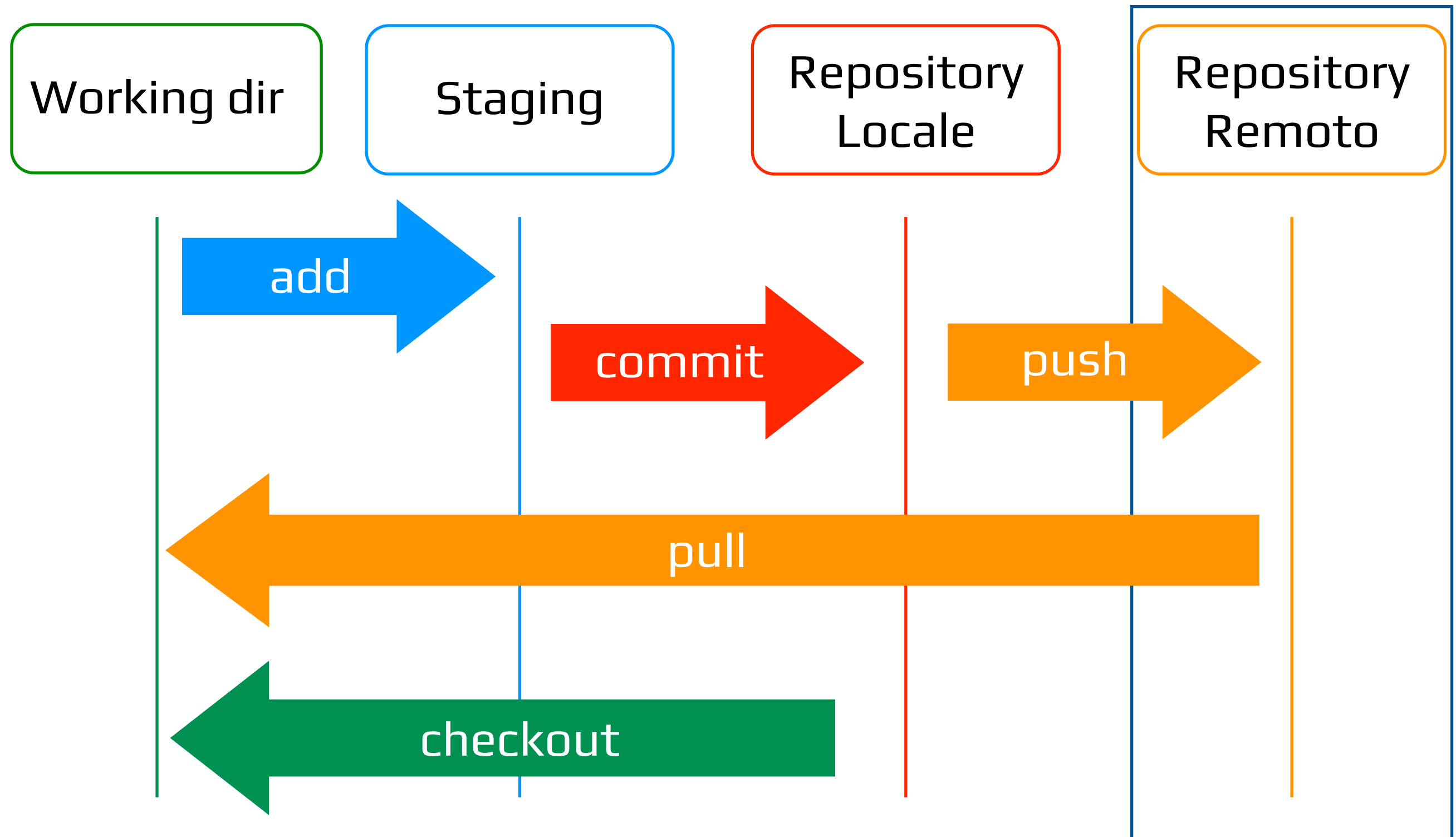
# Bitbucket

Una volta che il nostro repository locale è collegato al repository remoto, attraverso i comandi




```
git push  
git pull
```


sarà possibile mantenere le due copie del repository sincronizzate.


# Workflow di lavoro





# Gestione Repository





 **Tecnologie Web - Es...**


 Overview


 Source


 Commits


 Branches

 Pull requests

 Pipelines


 Downloads

 Boards



 Settings


Giacomo Piva / **Tecnologie Web - Esempi AA16/17**


**Overview**


 **HTTPS** <https://giacomopiva@bitbucket.org/g>


Last updated	2017-12-28	0	6
Language	PHP	Open PRs	Watchers
Access level	Admin	1	0
		Branch	Forks


 Share 

**Recent activity** 

 **1 commit**  
Pushed to giacomopiva/tecnologie-web-ese...  
[920e504](#) Completate azioni CRUD  
Giacomo Piva · 2017-06-08

 **1 commit**  
Pushed to giacomopiva/tecnologie-web-ese...  
[7c680b1](#) Esempio controlelr e view  
Giacomo Piva · 2017-06-06

 **1 commit**  
Pushed to giacomopiva/tecnologie-web-ese...  
[8889a38](#) Config iniziale e Modello  
Giacomo Piva · 2017-06-05

 Edit README

**README**

Esempi per il corso di Tecnologie Web

# Gestione Repository

php Technologie Web - Es...

Giacomo Piva / Technologie Web - Esempi AA16/17

## Overview

Download HTTPS <https://giacomopiva@bitbucket.org/g> Share

Last updated	2017-12-28	0	6
Language	PHP	Open PRs	Watchers
Access level	Admin	1	0
		Branch	Forks

Edit README

## README

Esempi per il corso di Tecnologie Web

### Recent activity

- 1 commit**  
Pushed to giacomopiva/tecnologie-web-ese...  
[920e504](#) Completate azioni CRUD  
Giacomo Piva · 2017-06-08
- 1 commit**  
Pushed to giacomopiva/tecnologie-web-ese...  
[7c680b1](#) Esempio controlelr e view  
Giacomo Piva · 2017-06-06
- 1 commit**  
Pushed to giacomopiva/tecnologie-web-ese...  
[8889a38](#) Config iniziale e Modello  
Giacomo Piva · 2017-06-05

Gestione del repository



# Gestione Repository

URL del repository

The screenshot displays a Bitbucket repository page. On the left, a sidebar contains navigation links: Overview (selected), Source, Commits, Branches, Pull requests, Pipelines, Downloads, Boards, and Settings. The main content area shows the repository name 'Tecnologie Web - Esempi AA16/17' and the owner 'Giacomo Piva'. Below this, the 'Overview' tab is active, showing a download button, a dropdown menu set to 'HTTPS', and the repository URL 'https://giacomopiva@bitbucket.org/g' which is highlighted with a red box. To the right of the URL, there are statistics: 0 Open PRs, 6 Watchers, 1 Branch, and 0 Forks. Below the statistics, there is a section for the 'README' with the text 'Esempi per il corso di Tecnologie Web'. On the right side of the page, there is a 'Recent activity' section showing three commits by Giacomo Piva, each with a commit hash and a description of the changes.

Repository Information	Open PRs	Watchers
Last updated: 2017-12-28 Language: PHP Access level: Admin	0	6
	1 Branch	0 Forks

**README**  
Esempi per il corso di Tecnologie Web

**Recent activity**

- 1 commit**  
Pushed to giacomopiva/tecnologie-web-ese...  
920e504 Completate azioni CRUD  
Giacomo Piva · 2017-06-08
- 1 commit**  
Pushed to giacomopiva/tecnologie-web-ese...  
7c680b1 Esempio controlelr e view  
Giacomo Piva · 2017-06-06
- 1 commit**  
Pushed to giacomopiva/tecnologie-web-ese...  
8889a38 Config iniziale e Modello  
Giacomo Piva · 2017-06-05

# Gestione Repository

## Gestione accessi

The screenshot displays a Bitbucket repository interface. On the left is a sidebar with navigation links: Overview (selected), Source, Commits, Branches, Pull requests, Pipelines, Downloads, Boards, and Settings. The main content area shows the repository 'Tecnologie Web - Esempi AA16/17' by user 'Giacomo Piva'. It includes an 'Overview' section with a download button, a dropdown menu set to 'HTTPS', and the repository URL 'https://giacomopiva@bitbucket.org/g...'. Below this is a table of statistics:

Last updated	2017-12-28	0	6
Language	PHP	Open PRs	Watchers
Access level	Admin	1	0
		Branch	Forks

To the right of the statistics is an 'Edit README' link. Below the statistics is a 'README' section with the text 'Esempi per il corso di Tecnologie Web'. On the right side of the page is a 'Recent activity' section showing three commits by Giacomo Piva, each with a commit hash and a description. In the top right corner, a red box highlights the 'Share' button.

# Gestione Repository

The screenshot displays a Bitbucket repository interface. On the left is a sidebar with navigation links: Overview (selected), Source, Commits, Branches, Pull requests, Pipelines, Downloads, Boards, and Settings. The main content area shows the repository 'Tecnologie Web - Esempi AA16/17' by Giacomo Piva. The 'Overview' tab is active, displaying repository statistics: Last updated 2017-12-28, Language PHP, Access level Admin, 0 Open PRs, 6 Watchers, 1 Branch, and 0 Forks. Below this is the 'README' section with the text 'Esempi per il corso di Tecnologie Web'. On the right, the 'Recent activity' section is highlighted with a red box, showing three commits by Giacomo Piva from June 2017. Each commit entry includes a commit hash, a description, and a date.

Repository Information	Open PRs	Watchers
Last updated: 2017-12-28 Language: PHP Access level: Admin	0	6
	1 Branch	0 Forks

**Recent activity**

- 1 commit**  
Pushed to giacomopiva/tecnologie-web-ese...  
920e504 Completate azioni CRUD  
Giacomo Piva · 2017-06-08
- 1 commit**  
Pushed to giacomopiva/tecnologie-web-ese...  
7c680b1 Esempio controlelr e view  
Giacomo Piva · 2017-06-06
- 1 commit**  
Pushed to giacomopiva/tecnologie-web-ese...  
8889a38 Config iniziale e Modello  
Giacomo Piva · 2017-06-05

Storico commit

# Esempio

Repository remoto

# Riferimenti

Documentazione di **git**: <https://git-scm.com/doc>

Tour di Bitbucket:

<https://confluence.atlassian.com/bitbucket/git-tutorial-keep-track-of-your-space-station-locations-759857287.html>

Elenco comandi essenziali di **git**:

<https://docs.google.com/document/d/18Z7JA96YLYmPCzRShnddRp9KHROBLO2a2wrcRUouwieg/edit?usp=sharing>

# Esempi delle lezioni

Gli esempi:

<https://bitbucket.org/giacomopiva/tecnologie-web-esempi-delle-lezioni>

Domande?