

# Fondamenti di programmazione

---

1

Configurazione ambiente  
di sviluppo

# Fondamenti di programmazione

---

- Link materiale del corso  
(accedere con l'email istituzionale `nome.cognome@community.unipa.it`)  
**[https://bit.ly/fp2021\\_2022](https://bit.ly/fp2021_2022)**
- Riferimento docente esercitazioni  
**[antonio.bordonaro@unipa.it](mailto:antonio.bordonaro@unipa.it)**
- Ricevimento  
**Edificio 6, 3° piano, Laboratorio  
di Intelligenza artificiale e Sistemi distribuiti**
- Orario Ricevimento  
**Da concordare via e-mail**
- Testo consigliato  
**Deitel, Deitel, Il linguaggio C – Fondamenti e tecniche di programmazione 8 Ed.,  
Pearson, Italia, 2016**



# SOMMARIO

- Definizione di Algoritmo
- Programma
- Linguaggi di programmazione
- Compilazione
- Ambiente di Sviluppo

# Definizione di *Algoritmo*

Un ***algoritmo*** è una sequenza ordinata di *operazioni* atte a risolvere un determinato problema

Viene eseguito da un generico *esecutore* (che non deve essere necessariamente un calcolatore)

L'esecutore, affinché possa seguire l'algoritmo, deve essere in grado di **interpretare** in modo corretto tutti i passi di cui è composto.

# Definizione di *Algoritmo*

Un algoritmo **deve** possedere le seguenti proprietà:

- **Atomicità:** le operazioni di cui è composto l'algoritmo devono essere elementari e non ulteriormente scomponibili
- **Non ambiguità:** i passi dell'algoritmo devono essere interpretabili in modo univoco e senza ambiguità
- **Finitezza:** l'algoritmo deve essere formato da un numero finito di passi
- **Effettività:** risultato univoco e deterministico

# Esempio di algoritmo umano

Un esempio di algoritmo che viene eseguito dall'uomo è quello relativo alla *preparazione del caffè*

1. **Svitare** la caffettiera
2. Riempire la parte inferiore della caffettiera con **acqua**
3. Inserire il **filtro** nella caffettiera
4. Se si dispone di **caffè macinato**:
  - a) Riempire il filtro con il caffè macinato
5. Altrimenti, se si dispone di **caffè in chicchi**:
  - a) Macinarlo e ripetere il punto 4
6. **Avvitare** la caffettiera
7. **Accendere** il fuoco a gas
8. Posizionare la moka sul fuoco
9. Attendere che il caffè sia pronto
10. Spegnerne il fuoco
11. Caffè pronto. (**Fine**)

# Programma

Un programma è la descrizione di un algoritmo che un calcolatore può interpretare ed eseguire correttamente

La programmazione è l'insieme di attività che portano alla realizzazione di un programma

In genere, le fasi principali della programmazione comprendono:

- **Analisi:** viene individuato il problema che si intende risolvere
- **Scomposizione del problema:** si scompone il problema in sotto-problemi più semplici, tramite un approccio **top-down**, arrivando alla definizione di un algoritmo
- **Codifica dell'algoritmo:** l'algoritmo viene codificato in un determinato linguaggio di programmazione
- **Test:** il programma viene testato per verificare che non siano presenti malfunzionamenti (*bug*) e che sia coerente con quanto definito in fase di analisi

# Linguaggi di programmazione

Sono linguaggi **formali** che vengono utilizzati per tradurre un algoritmo (che può anche essere espresso in modo *informale*) in un formato *più vicino* al calcolatore

Differentemente dai linguaggi non-formali (quale può essere il linguaggio naturale) non deve essere **ambiguo**

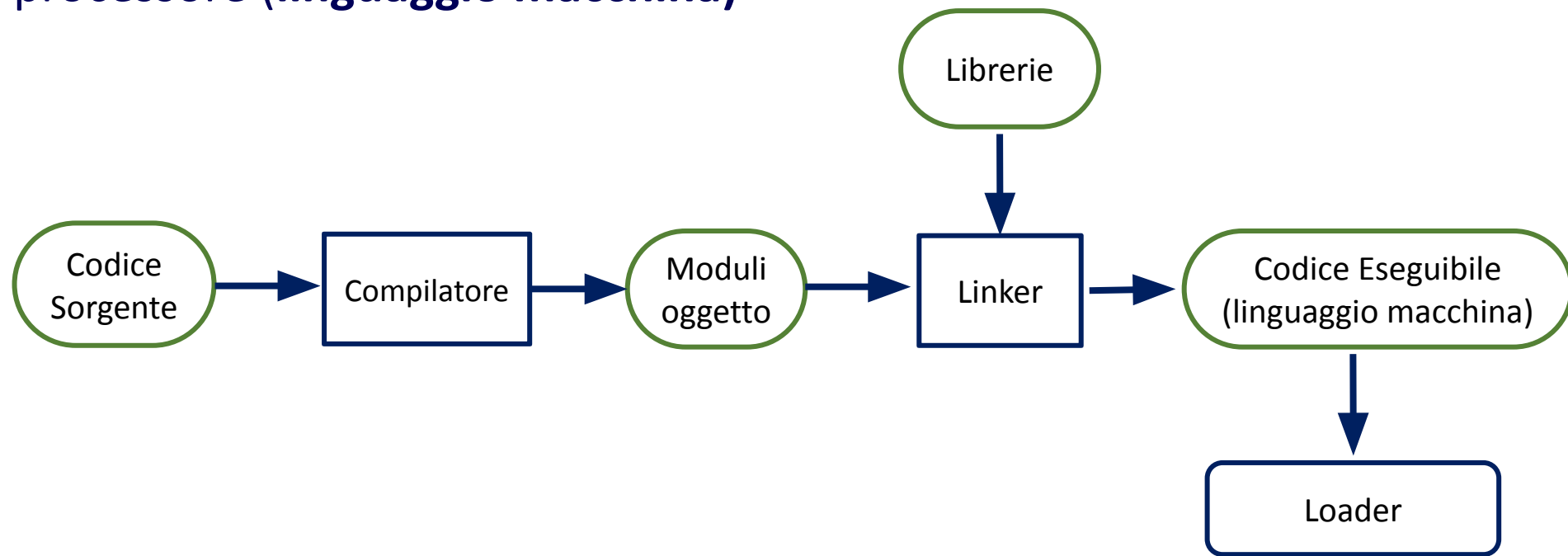
Anche se i linguaggi di programmazione sono formali e *vicini* al calcolatore, in genere non sono **direttamente** interpretabili dal calcolatore (o, meglio ancora, dal **processore**)

*E' necessario tradurre il codice sorgente in un linguaggio che sia interpretabile dal processore*

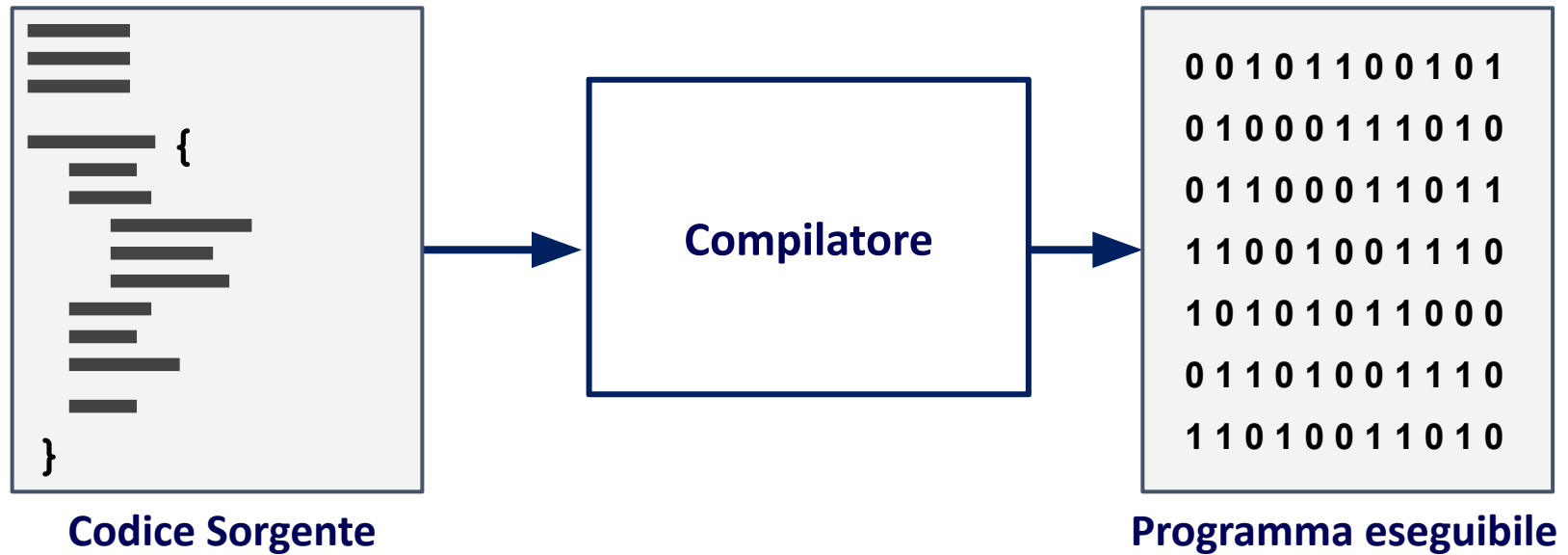


# Compilazione

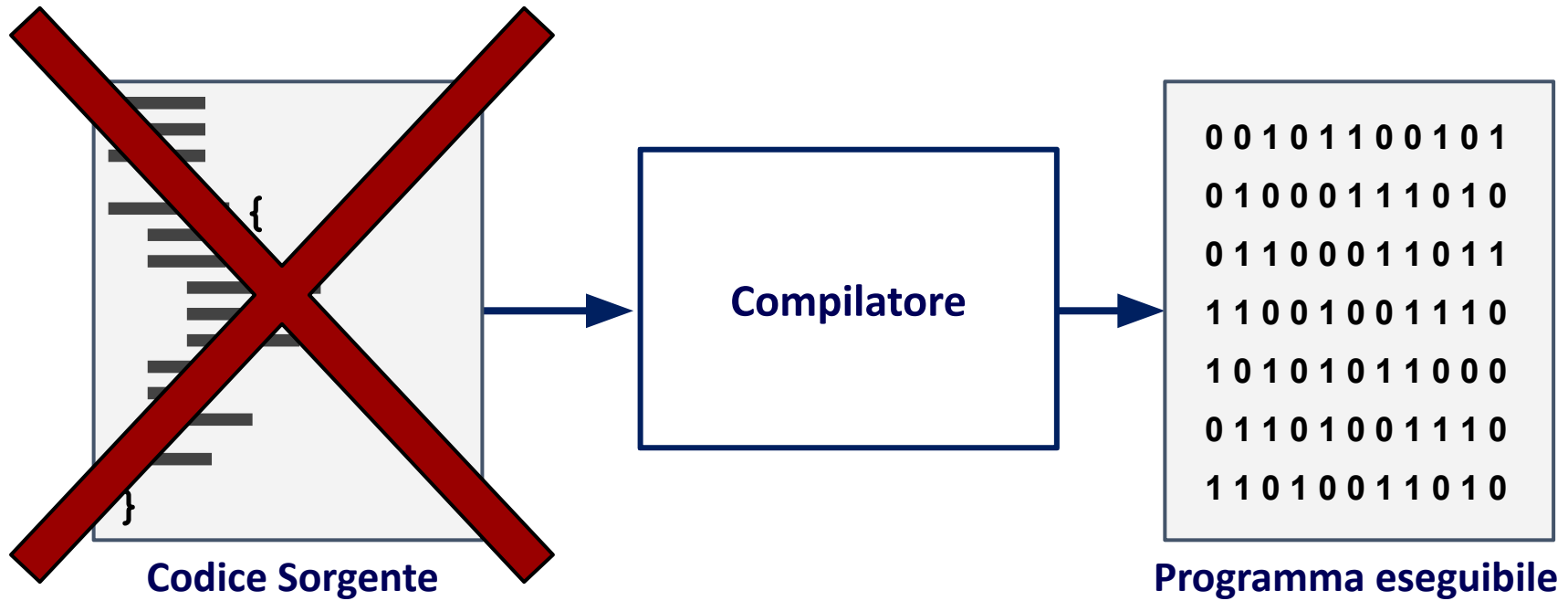
La **compilazione** è l'insieme dei processi che traducono il codice sorgente di un programma (scritto in un determinato linguaggio di programmazione) in un formato che sia interpretabile direttamente dal processore (**linguaggio macchina**)



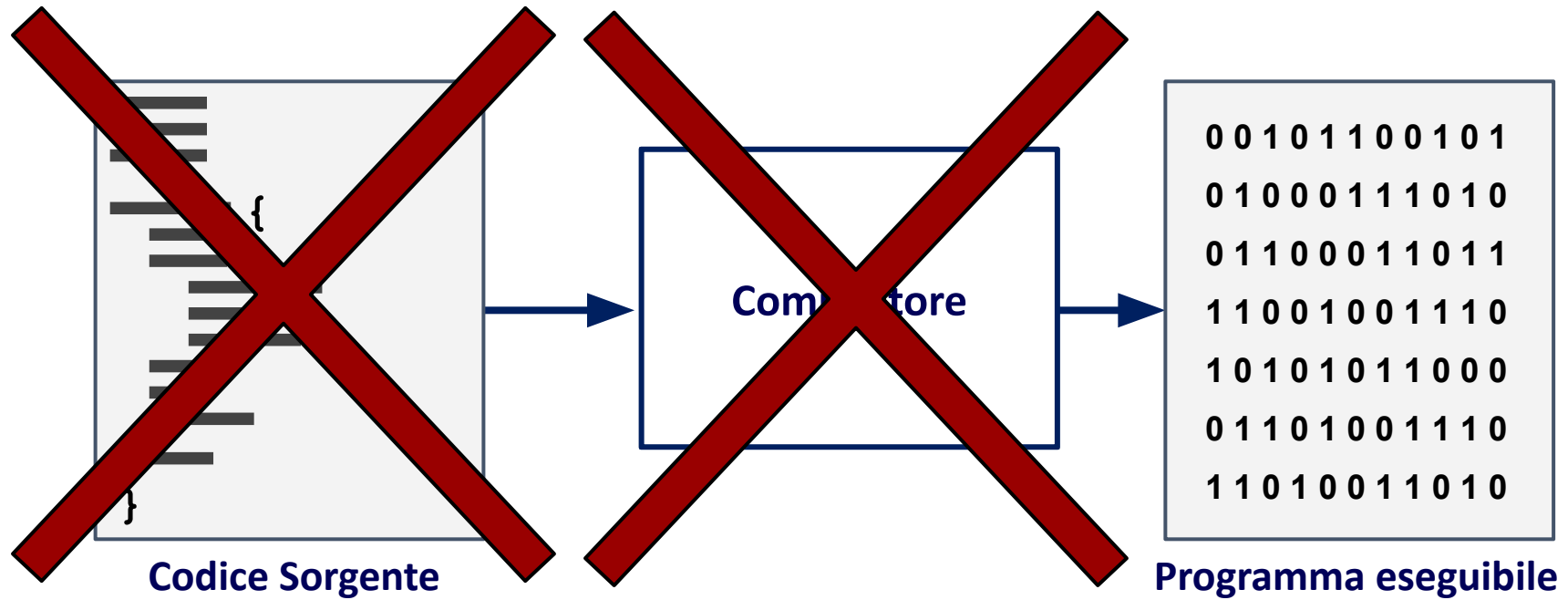
# Esecuzione di un programma



# Esecuzione di un programma



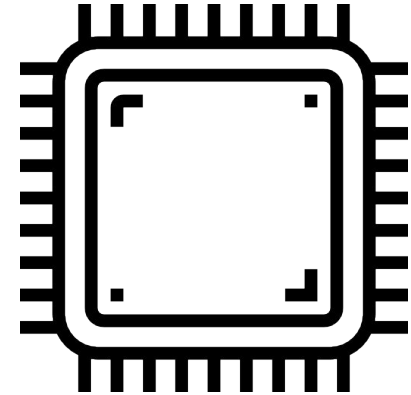
# Esecuzione di un programma



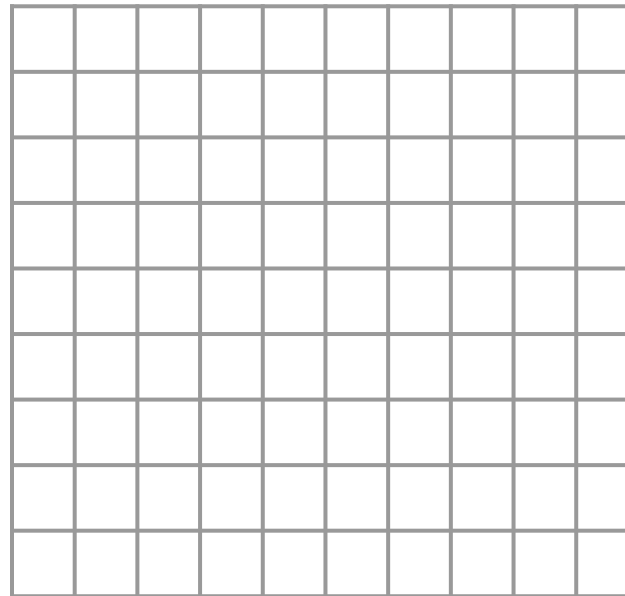
# Esecuzione di un programma

```
0 0 1 0 1 1 0 0
0 1 0 0 1 1 0 1
0 1 1 0 0 1 0 1
1 1 0 0 1 1 1 1
1 0 1 0 1 1 0 0
0 1 1 0 1 1 1 1
1 1 0 1 0 1 0 1
```

**Programma eseguibile**



**Processore**

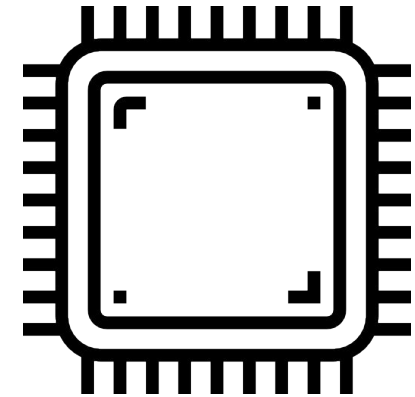


**Memoria Principale (RAM)**

# Esecuzione di un programma

	0	0	1	0	1	1	0	0	
	0	1	0	0	1	1	0	1	
	0	1	1	0	0	1	0	1	
	1	1	0	0	1	1	1	1	
	1	0	1	0	1	1	0	0	
	0	1	1	0	1	1	1	1	
	1	1	0	1	0	1	0	1	

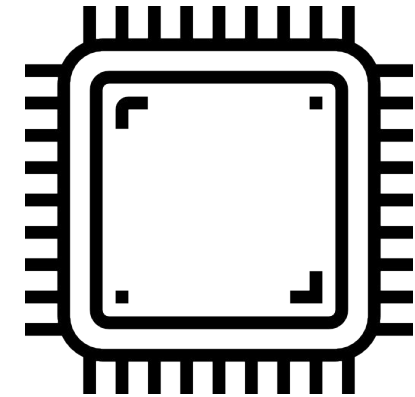
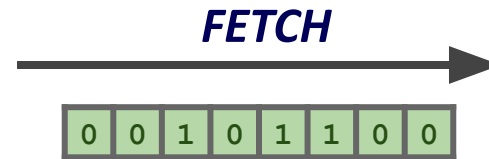
**Memoria Principale (RAM)**



**Processore**

# Esecuzione di un programma

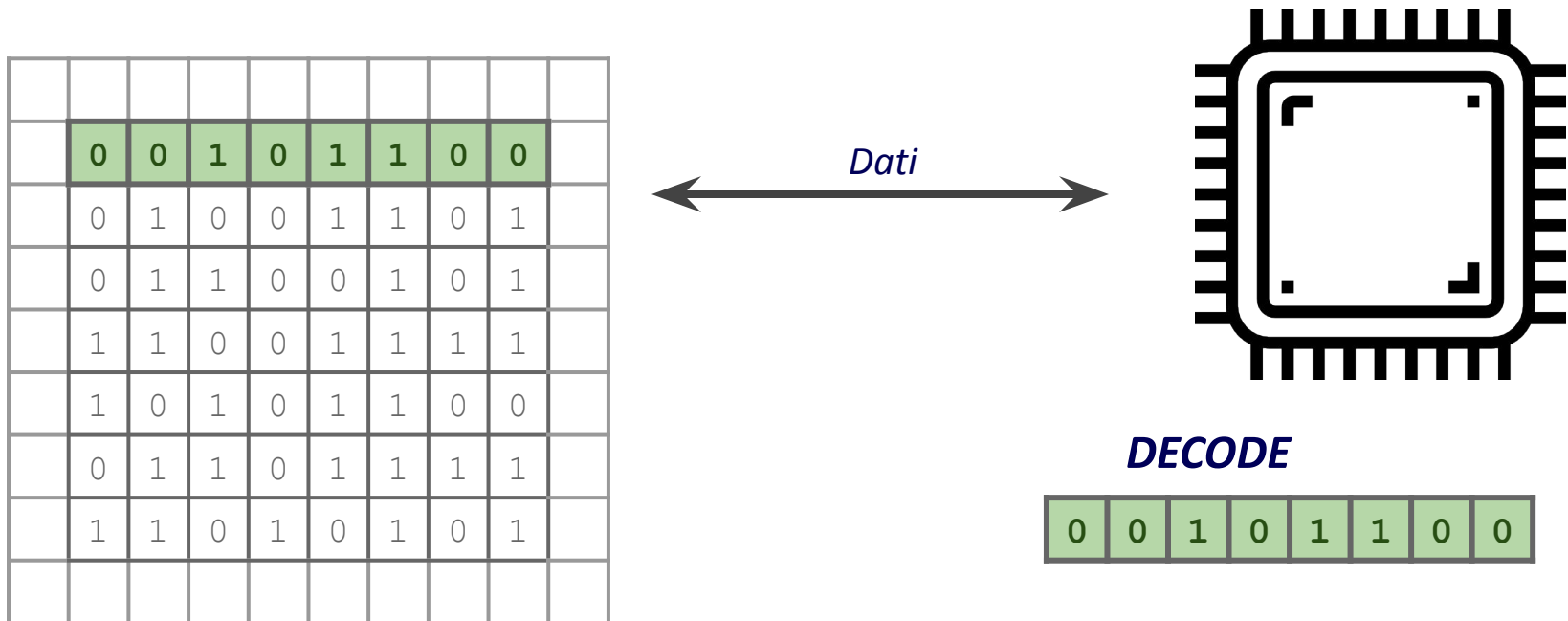
	0	0	1	0	1	1	0	0	
	0	1	0	0	1	1	0	1	
	0	1	1	0	0	1	0	1	
	1	1	0	0	1	1	1	1	
	1	0	1	0	1	1	0	0	
	0	1	1	0	1	1	1	1	
	1	1	0	1	0	1	0	1	



**Processore**

***FETCH:*** Il processore accede alla memoria principale ed estrae la **prossima istruzione**.

# Esecuzione di un programma

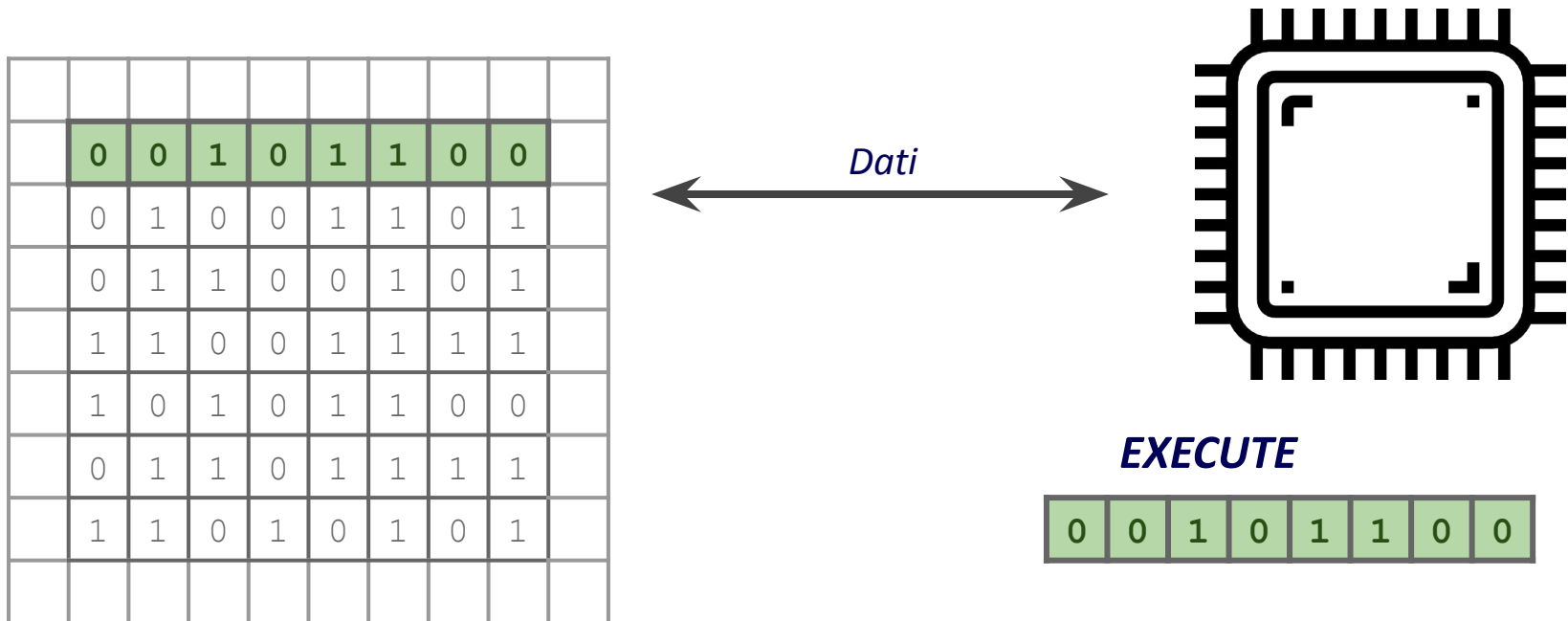


**Decode:** Il processore **decodifica** l'istruzione (*capisce* di che tipo di istruzione si tratta). Il processore carica dalla memoria anche eventuali **dati** richiesti dall'istruzione. Per esempio, se si tratta dell'istruzione di **somma**, il processore carica gli *addendi* dalla memoria e li inserisce nei suoi **registri**).

Alla fine della fase di decodifica, il processore aggiorna l'indirizzo della **successiva istruzione** da eseguire.



# Esecuzione di un programma



**Execute:** Il processore **esegue** l'istruzione. Per esempio, se si tratta dell'istruzione di **somma**, il processore calcola la somma degli **operandi** precedentemente caricati e salva il risultato dell'operazione in **memoria**.

Se tutto va a buon fine, il ciclo **Fetch-Decode-Execute** può ricominciare.

# Compilazione

Ricapitolando, per la scrittura di un programma, dunque, è necessario:

- **Editor di testo:** programma che permette la scrittura del codice
- **Compilatore:** traduce il codice sorgente in linguaggio macchina
- **Linker:** programma che prende più moduli oggetto (che vengono prodotti dal compilatore) e li *combina* per generare un unico codice binario eseguibile. In genere viene invocato automaticamente dal compilatore

Esistono anche software di sviluppo integrati (IDE, Integrated Development Environment) che includono, fra le altre cose, gli strumenti descritti precedentemente. Alcuni di questi sono:

- Eclipse
- Code::Blocks
- NetBeans
- ...

Hanno molti *pro* (processi automatizzati, gestione semplificata dei progetti di grande dimensione, segnalazione di errori di sintassi) ma anche qualche *contro* (richiedono spesso risorse hardware notevoli e una configurazione iniziale...)

# Ambiente di sviluppo

E' possibile sviluppare programmi in C su qualsiasi **sistema operativo**.

Per semplicità, verrà utilizzato un sistema operativo basato su *Linux*, nello specifico **Ubuntu**, sul quale sono presenti tutti gli strumenti necessari per lo sviluppo di programmi in C.

- **Editor di Testo:** Gedit, Mousepad, Emacs, Sublime....
- **Compilatore:** *gcc* (GNU Compiler Collection)
- **Linker:** *ld* (ma viene invocato direttamente dal compilatore)

# Ambiente di sviluppo

Quando si inizia ad avere a che fare con sistemi operativi basati su Linux, spesso viene usato il termine “**distribuzione Linux**” (o anche **distro**)

Con il termine **distribuzione linux** (o semplicemente **distro**) si fa riferimento ad una *distribuzione software* di un sistema operativo realizzato a partire dal Kernel Linux (il *core* di un sistema operativo).

Ogni distribuzione ha il proprio *desktop environment* (KDE, XFCE, GNOME), i propri software (gestore pacchetti, gestore delle finestre, applicazioni specifiche..)..  
..



# Ambiente di sviluppo

Per l'installazione di una macchina virtuale Ubuntu, bisogna:

1. Scaricare ed Installare **Virtual Box**
2. Scaricare la distribuzione **Ubuntu**
3. Installare Ubuntu su VirtualBox.

A seguire vengono descritti nel dettaglio i passi necessari per  
l'intera configurazione

# Ambiente di sviluppo

## 1. Download VirtualBox:

<https://www.virtualbox.org/wiki/Downloads>



About  
Screenshots  
Downloads  
Documentation  
    End-user docs  
    Technical docs  
Contribute  
Community

# VirtualBox

## Download VirtualBox

Here you will find links to VirtualBox binaries and its source code.

### VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

If you're looking for the latest VirtualBox 5.1 packages, see [VirtualBox 5.1 builds](#). Consider upgrading.

### VirtualBox 5.2.18 platform packages

- [Windows hosts](#)
- [OS X hosts](#)
- [Linux distributions](#)
- [Solaris hosts](#)

**Selezionare il proprio sistema operativo**

The binaries are released under the terms of the GPL version 2.

See the [changelog](#) for what has changed.

You might want to compare the checksums to verify the integrity of downloaded packages. *The SHA256 checksums should be favored ; treated as insecure!*

- [SHA256 checksums](#), [MD5 checksums](#)

**Note:** After upgrading VirtualBox it is recommended to upgrade the guest additions as well.

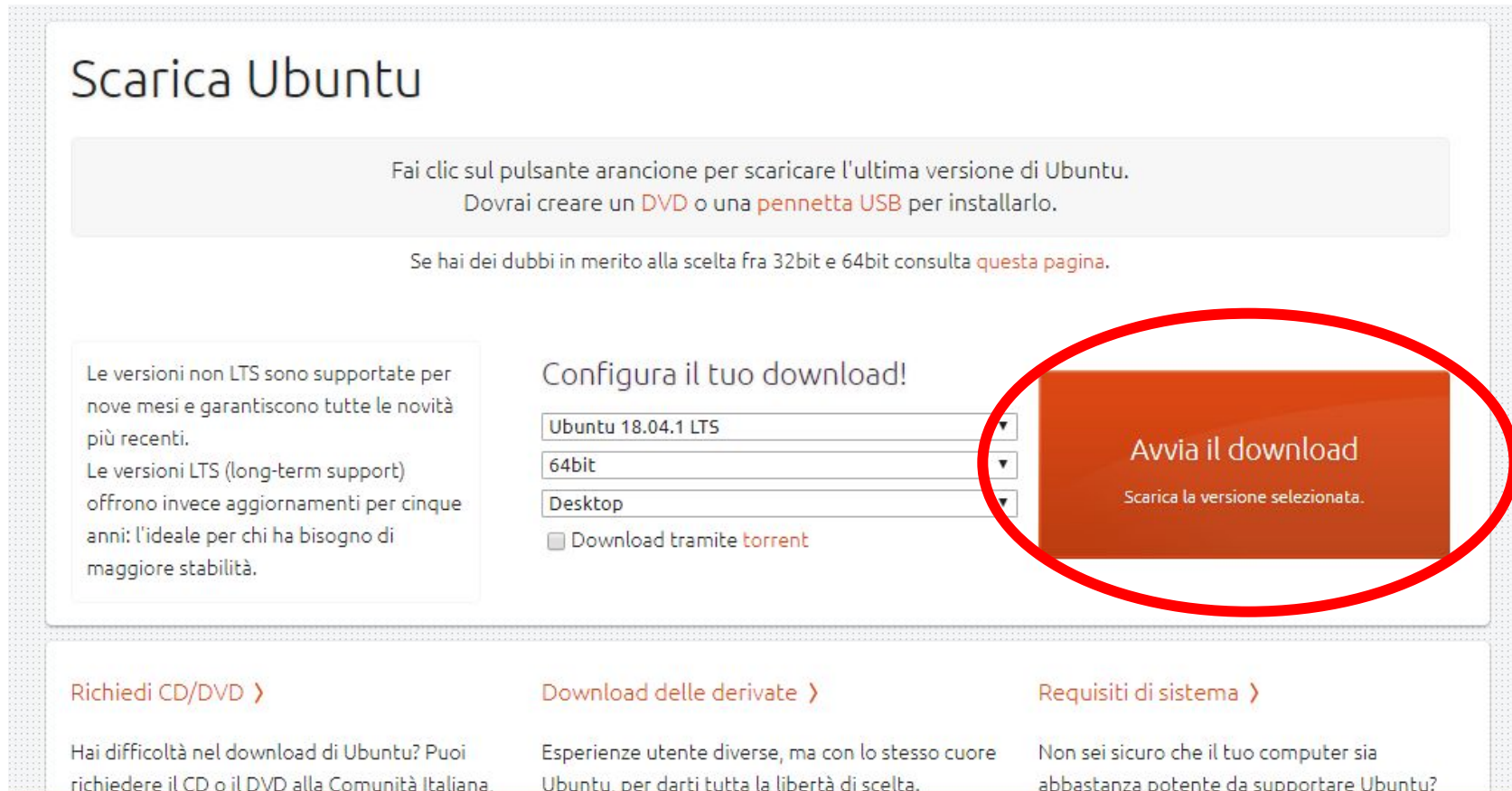
### VirtualBox 5.2.18 Oracle VM VirtualBox Extension Pack

- [All supported platforms](#)

# Ambiente di sviluppo

## 2. Download Ubuntu: <https://www.ubuntu-it.org/download>

(Scaricare la versione mostrata in figura)



Scarica Ubuntu

Fai clic sul pulsante arancione per scaricare l'ultima versione di Ubuntu.  
Dovrai creare un DVD o una pennetta USB per installarlo.

Se hai dei dubbi in merito alla scelta fra 32bit e 64bit consulta [questa pagina](#).

Le versioni non LTS sono supportate per nove mesi e garantiscono tutte le novità più recenti.  
Le versioni LTS (long-term support) offrono invece aggiornamenti per cinque anni: l'ideale per chi ha bisogno di maggiore stabilità.

Configura il tuo download!

Ubuntu 18.04.1 LTS  
64bit  
Desktop

☐ Download tramite [torrent](#)

**Avvia il download**  
Scarica la versione selezionata.

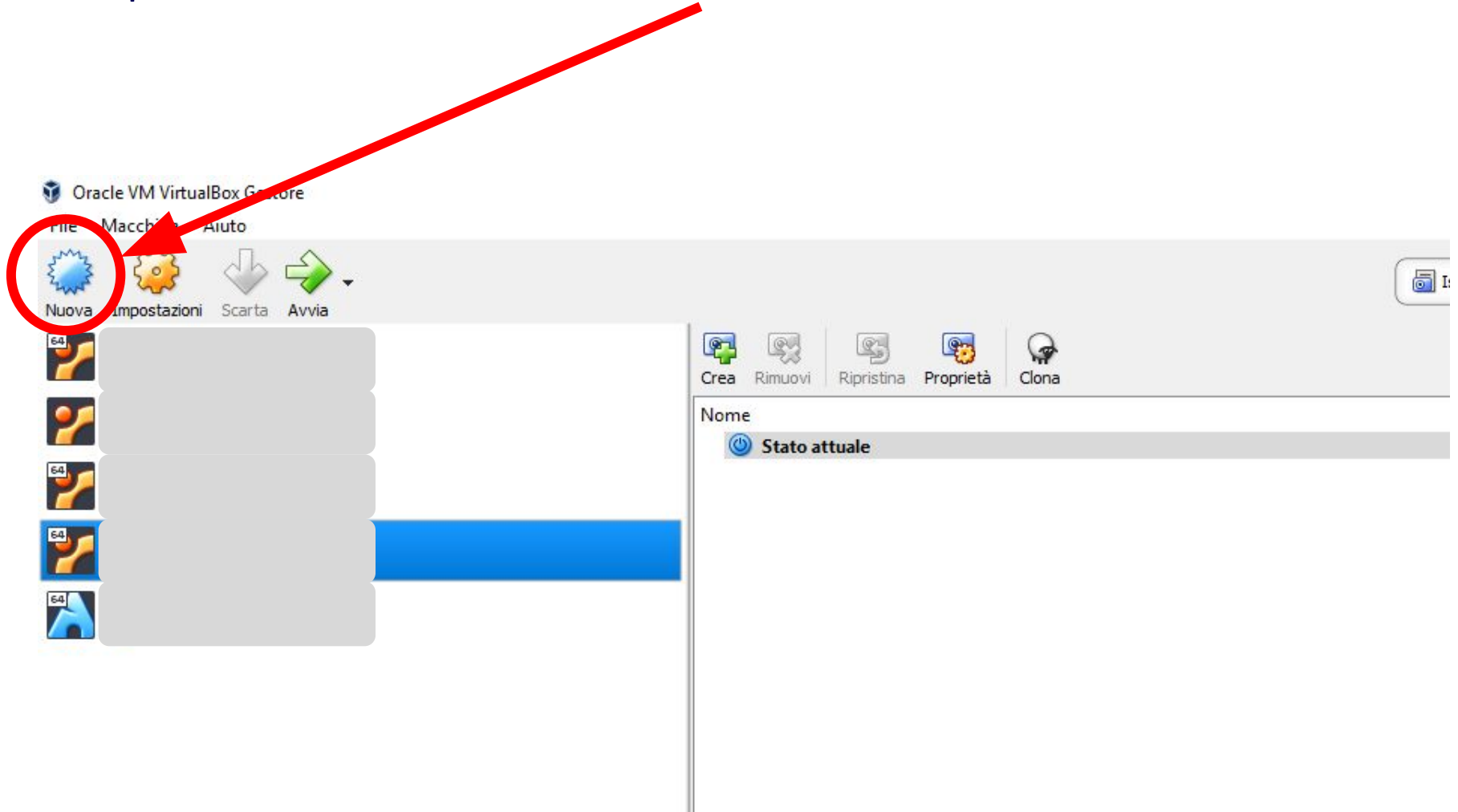
[Richiedi CD/DVD >](#)  
Hai difficoltà nel download di Ubuntu? Puoi richiedere il CD o il DVD alla Comunità Italiana.

[Download delle derivate >](#)  
Esperienze utente diverse, ma con lo stesso cuore Ubuntu, per darti tutta la libertà di scelta.

[Requisiti di sistema >](#)  
Non sei sicuro che il tuo computer sia abbastanza potente da supportare Ubuntu?

# Ambiente di sviluppo

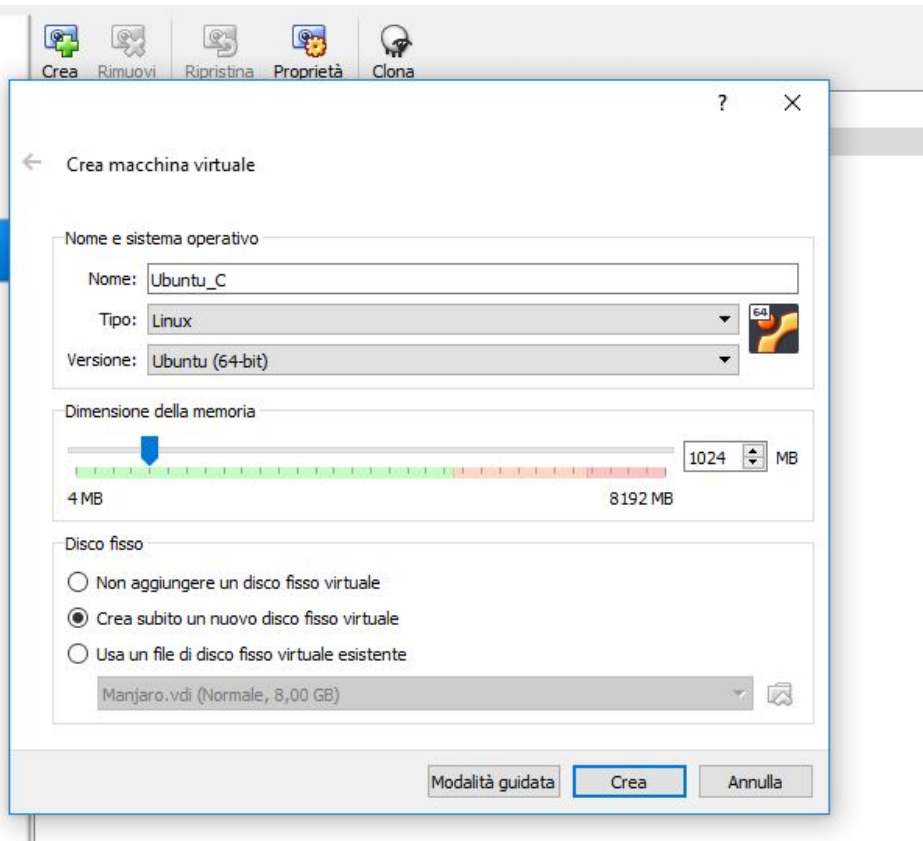
## 3. Aprire virtualBox e selezionare **Nuova**





# Ambiente di sviluppo

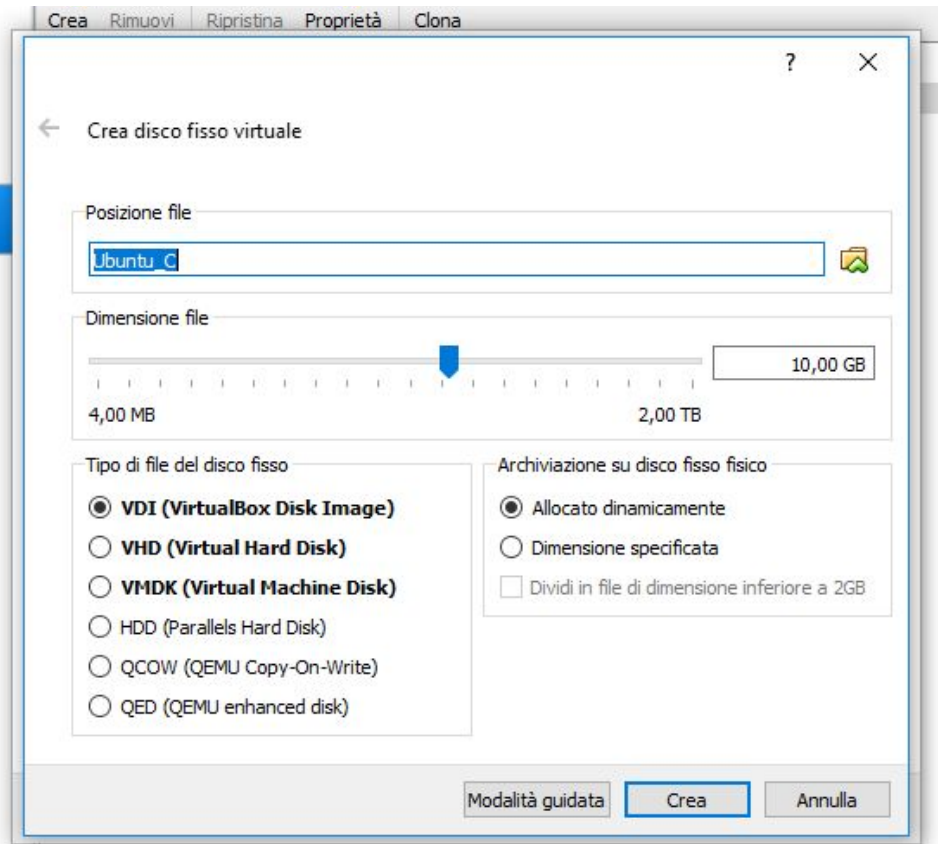
4. Inserire i parametri come in figura (e come descritto accanto) e selezionare **Crea**



- **Nome:** nome arbitrario. Serve solo per identificare la macchina che si sta installando
- **Tipo:** tipologia del sistema operativo che si intende installare. Selezionare Linux
- **Versione:** versione del sistema operativo. Selezionare Ubuntu
- **Dimensione della memoria:** quantità di RAM da assegnare alla macchina virtuale. Dipende dal proprio computer. In genere si consiglia di non assegnare più della metà della quantità totale di RAM che si ha a disposizione (es: in un calcolatore con 8GB di RAM, non assegnare ad una macchina virtuale più di 4GB).

# Ambiente di sviluppo

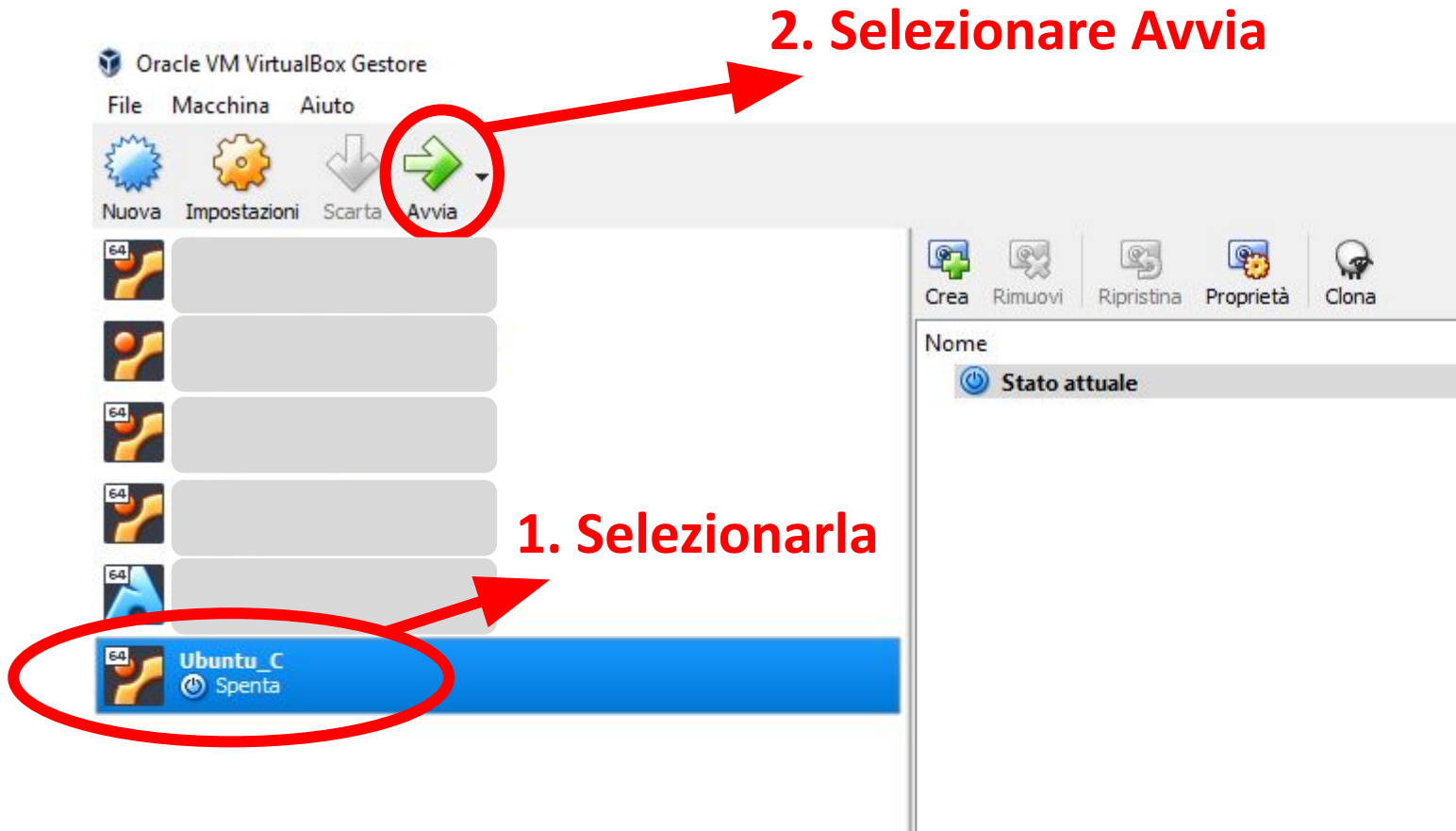
5. Inserire i parametri come in figura (e come descritto accanto) e selezionare **Crea**



- **Posizione File:** posizione in cui verrà salvata l'immagine del disco che si sta per creare. Si può selezionare qualsiasi percorso (banalmente, si può creare una cartella sul Desktop e selezionarla)
- **Dimensione file:** quantità di memoria che si intende assegnare alla macchina che si sta installando. Una dimensione di 10GB indica che il sistema Ubuntu verrà installato in un disco (virtuale) di 10GB, che poi sarà anche lo spazio che si avrà a disposizione. 15GB sono più che sufficienti.
- **Lasciare gli altri parametri invariati**

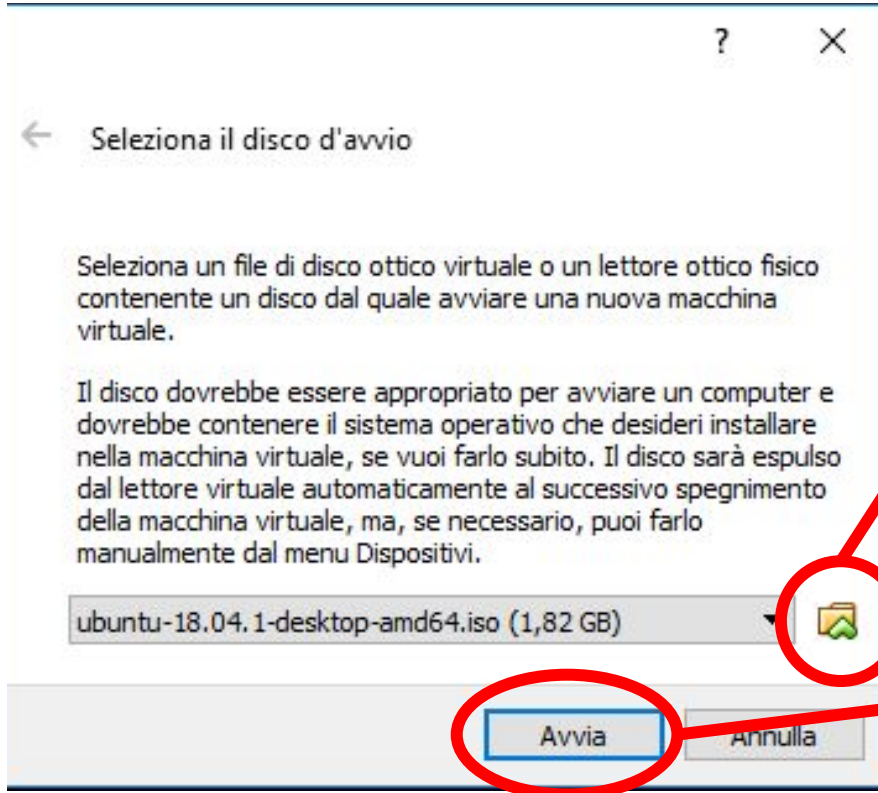
# Ambiente di sviluppo

6. Selezionare la macchina appena creata e cliccare su **Avvia**



# Ambiente di sviluppo

**7. Cliccare sull'icona di selezione (mostrata in figura) e selezionare il file di Ubuntu scaricato al passo 2. Premere su **Avvia****

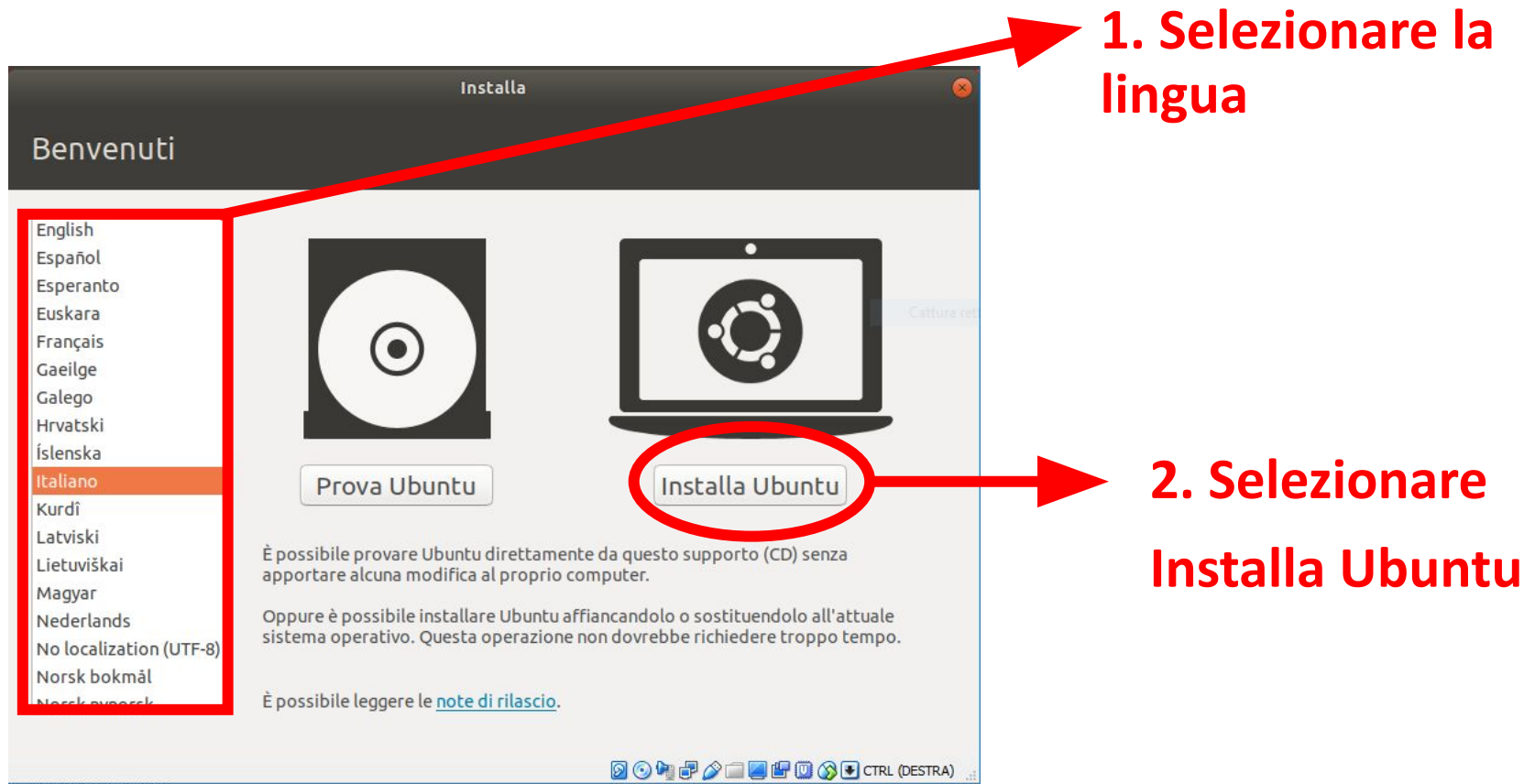


**1. Cliccare sull'icona e selezionare il disco di Ubuntu scaricato precedentemente.**

**2. Cliccare su Avvia**

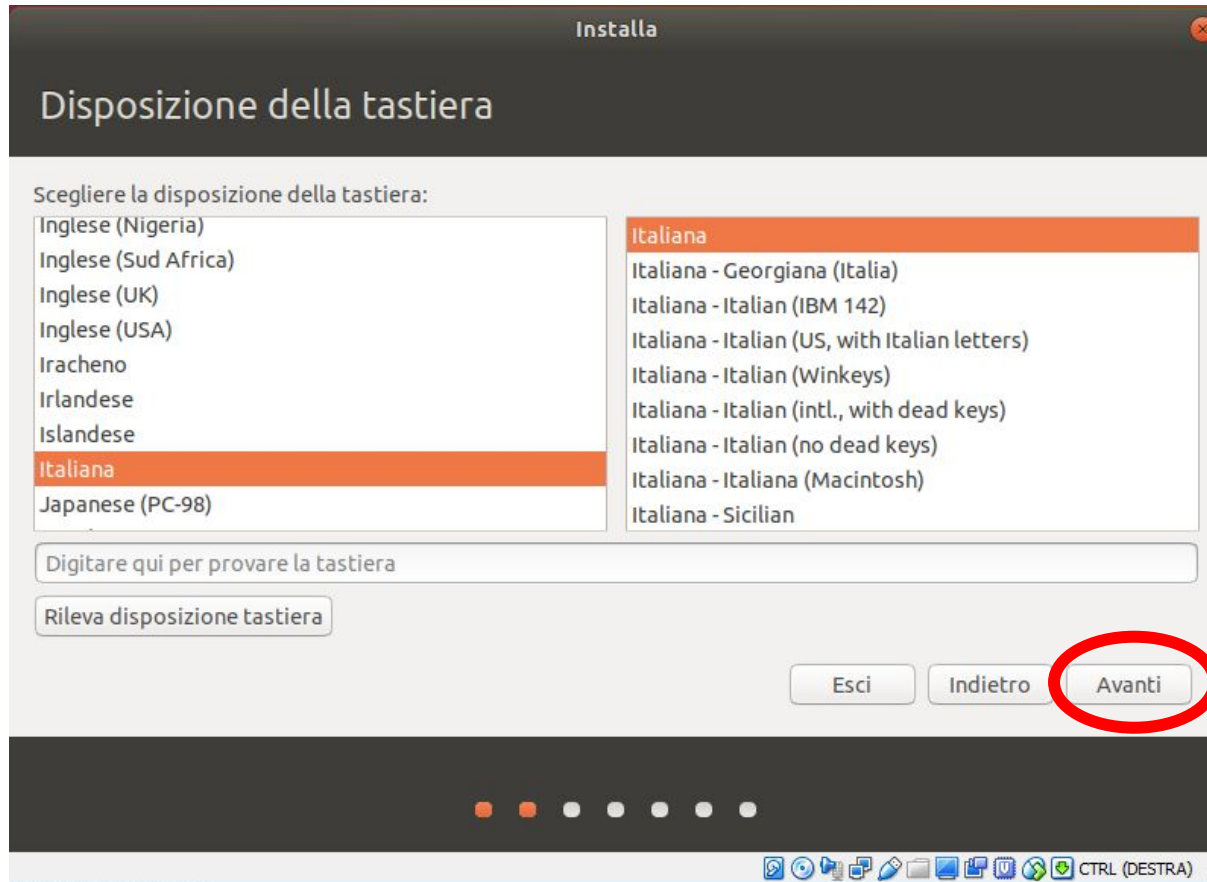
# Ambiente di sviluppo

## 8. Selezionare la lingua desiderata e cliccare su **Installa Ubuntu**



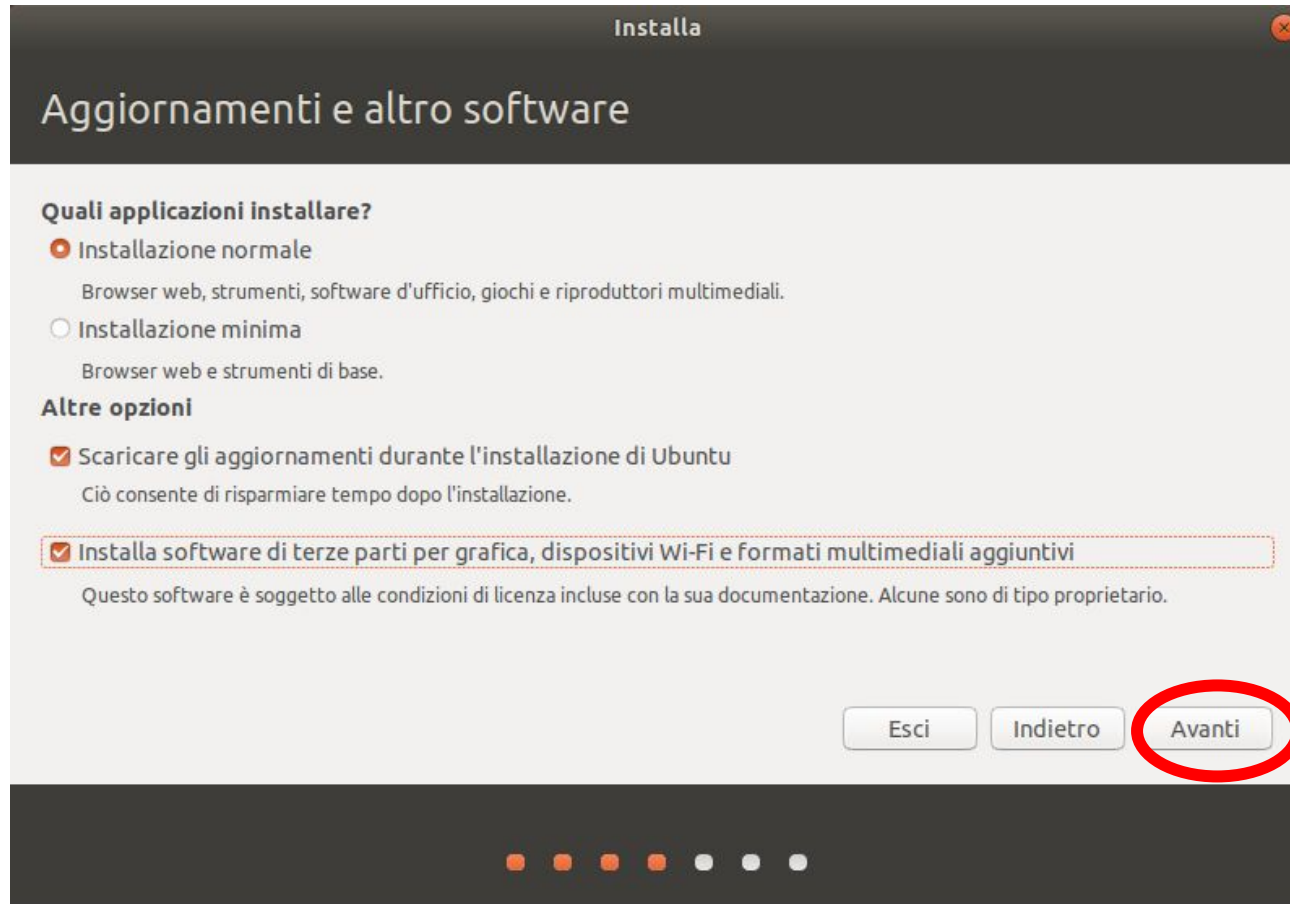
# Ambiente di sviluppo

9. Selezionare il layout della tastiera (come mostrato in figura) e selezionare **Avanti**



# Ambiente di sviluppo

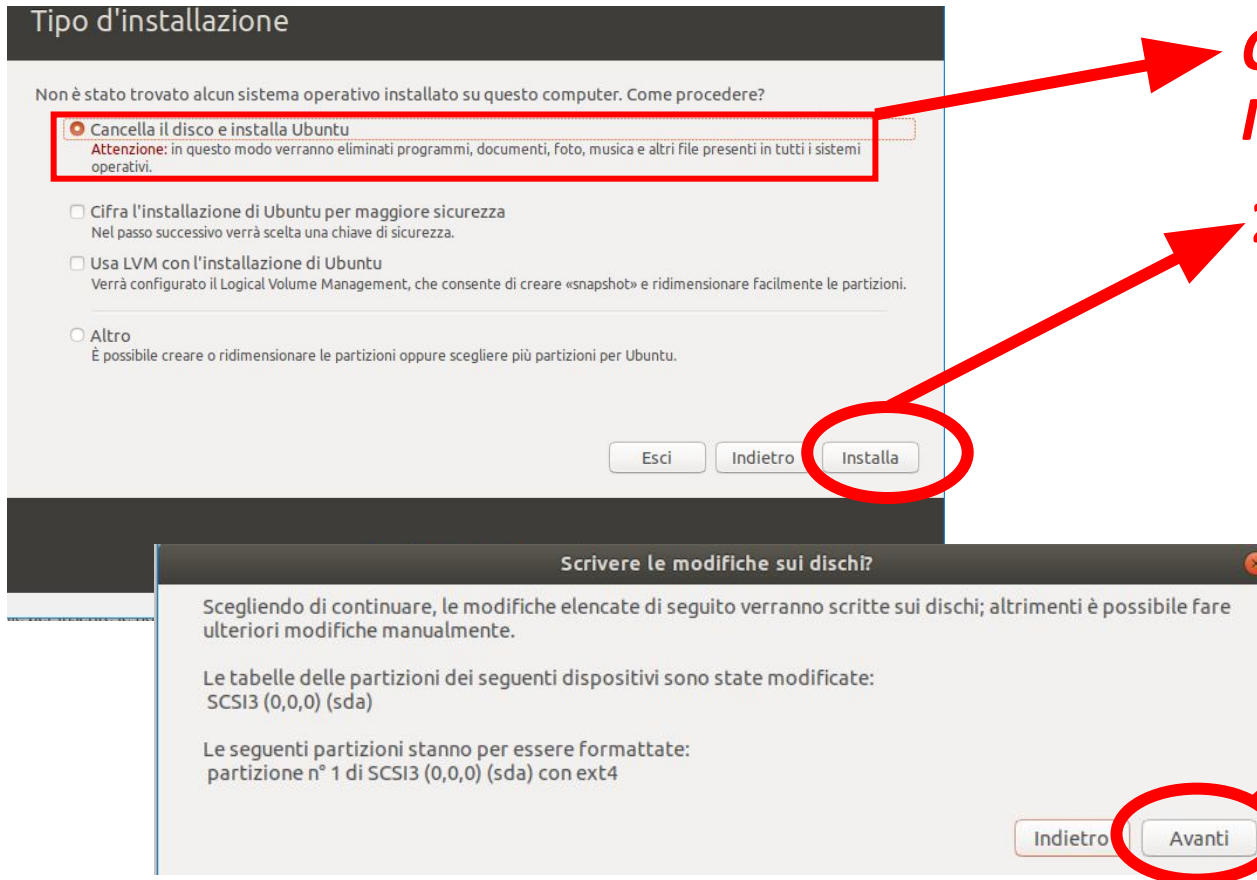
**10.** Selezionare i parametri mostrati in figura e cliccare su **Avanti**





# Ambiente di sviluppo

**11.** Selezionare i parametri mostrati in figura e cliccare su **Avanti**. Alla richiesta di conferma, selezionare **Avanti**.



**1. Selezionare  
Cancella il disco e  
Installa Ubuntu**

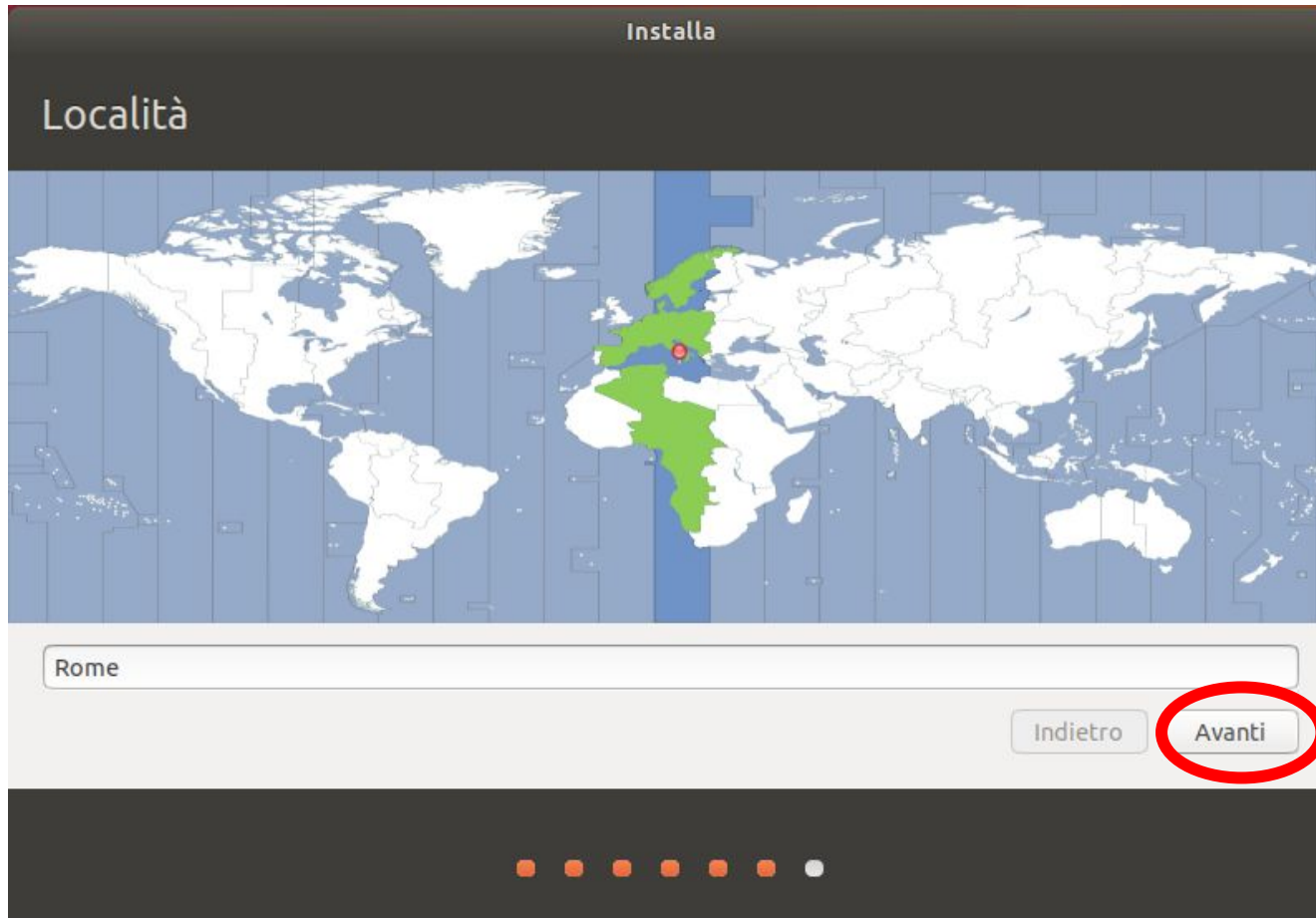
**2. Selezionare Avanti**

**3. Alla richiesta  
di conferma,  
selezionare  
Avanti**



# Ambiente di sviluppo

## 12. Selezionare la località (Roma) e cliccare su **Avanti**



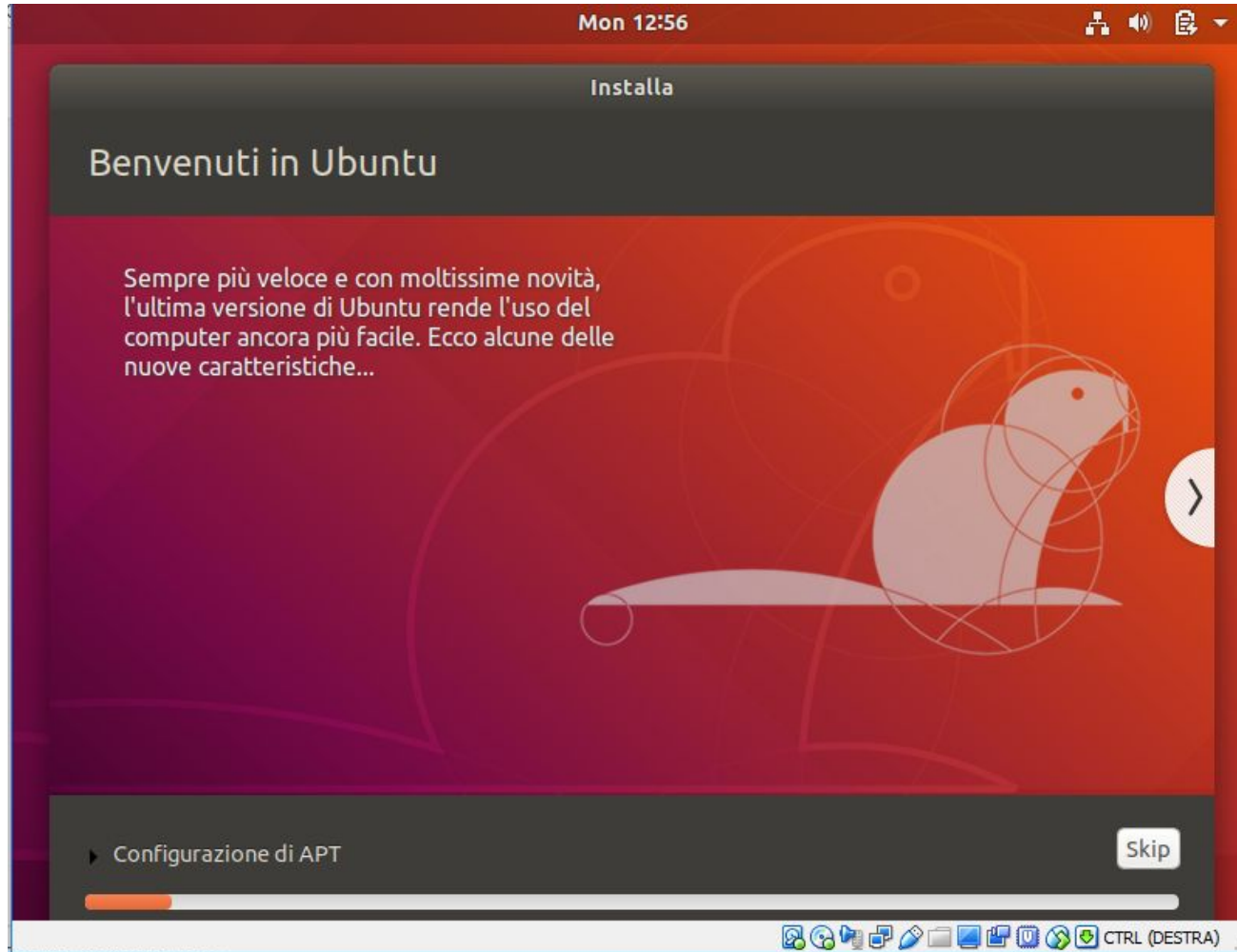
# Ambiente di sviluppo

**13.** Inserire **Nome** (arbitrario), **Nome Computer** (arbitrario) e **Nome Utente** (utente del sistema operativo, login...), **Password** e indicare anche se la password dovrà essere richiesta per eseguire l'accesso o meno. Selezionare **Avanti**

**Password obbligatoria  
anche se si decide di  
Accedere automaticamente**

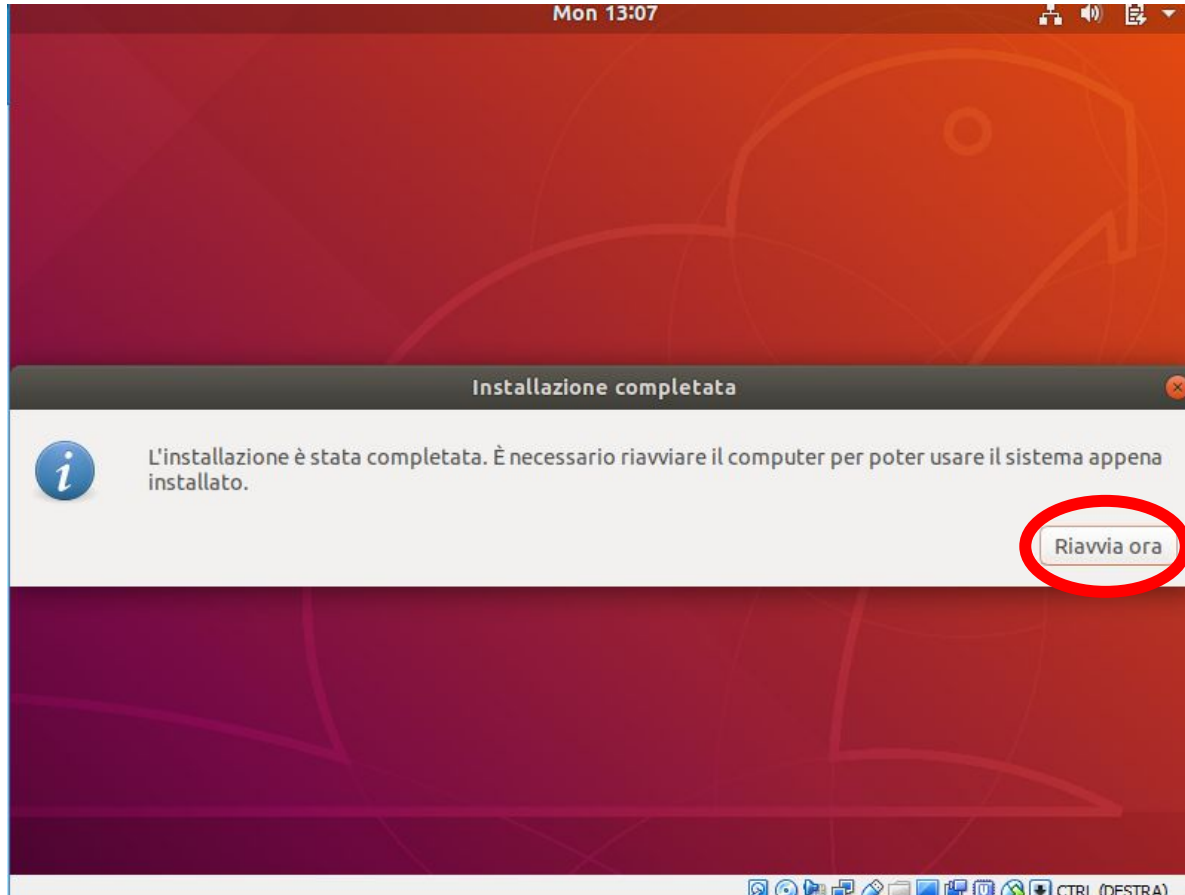
# Ambiente di sviluppo

## 14. Attendere la fine dell'installazione....



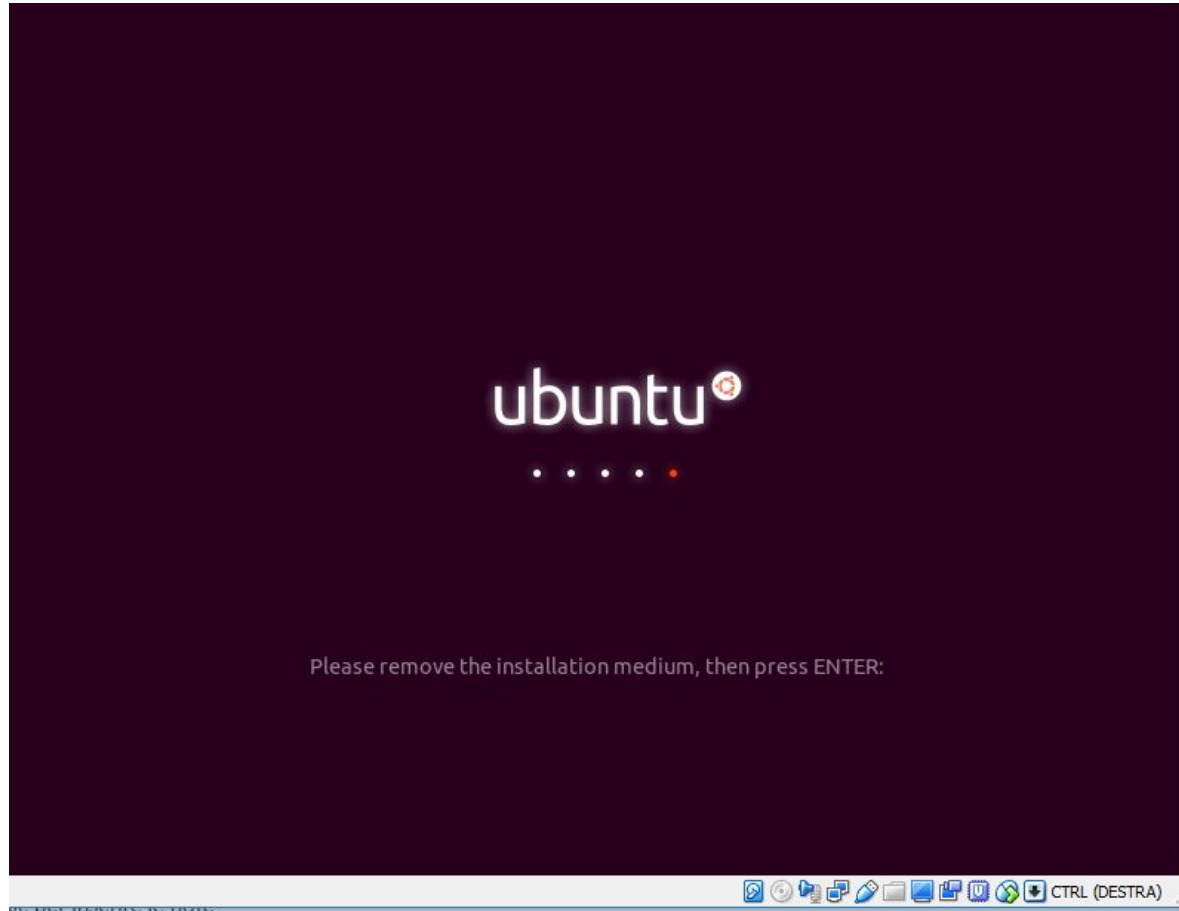
# Ambiente di sviluppo

15. Selezionare **Riavvia ora** e attendere che il sistema si riavii



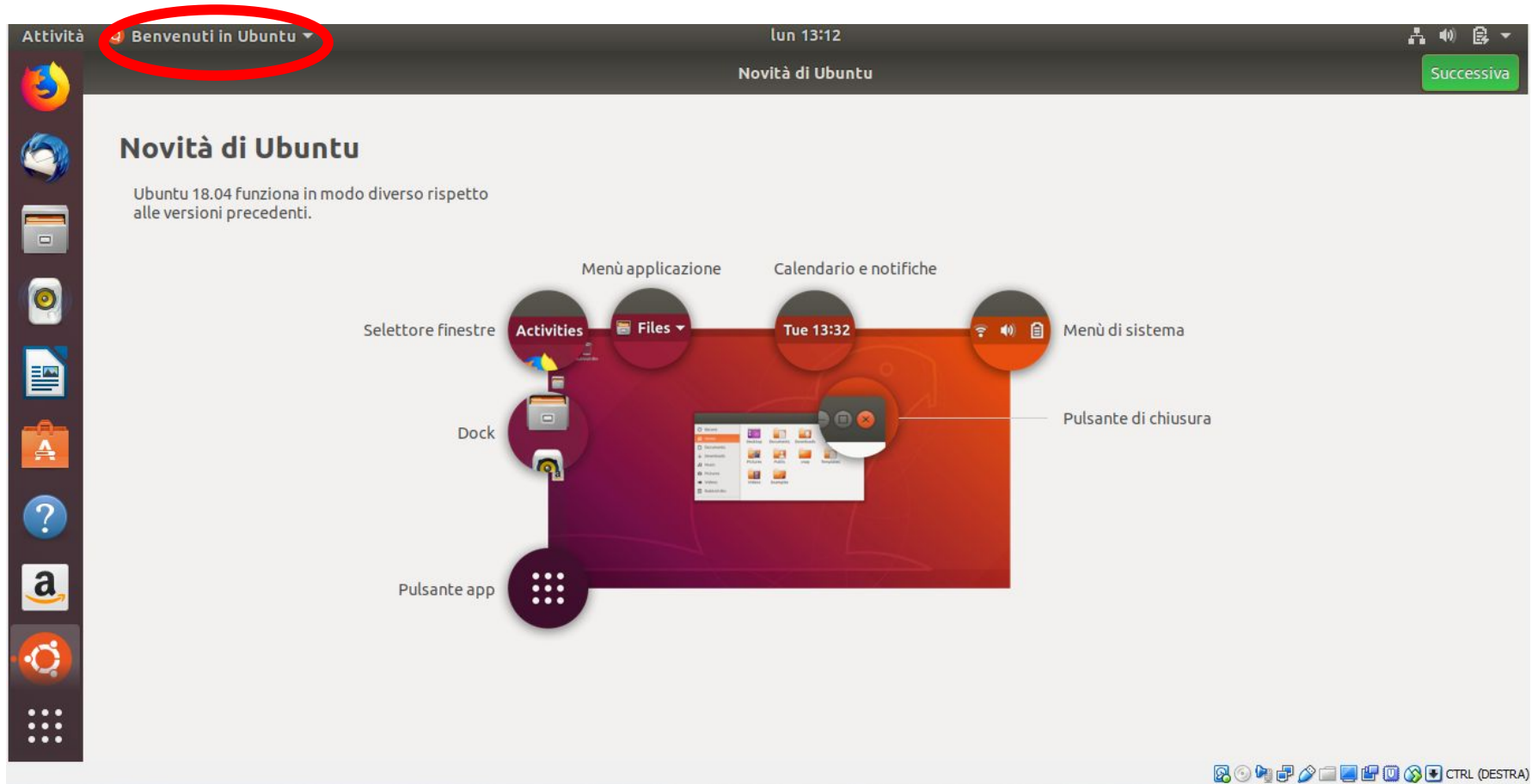
# Ambiente di sviluppo

**16.** Premere semplicemente **Enter** (Invio) e attendere...



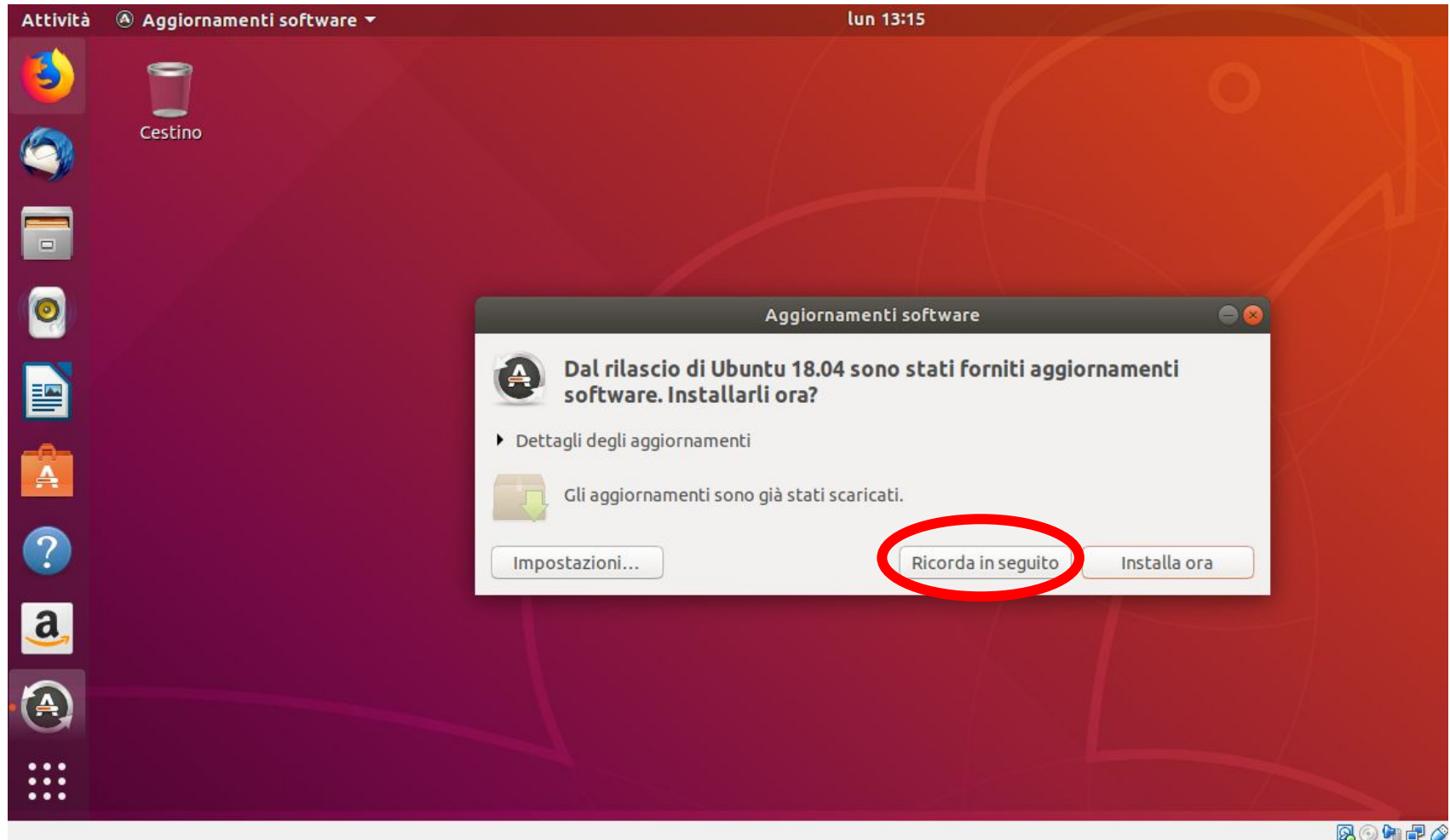
# Ambiente di sviluppo

**17.** Se non si sono verificati errori, Ubuntu partirà mostrando la schermata seguente. Selezionare l'icona *Benvenuti in Ubuntu* e, dal menù a tendina, selezionare *Esci*.



# Ambiente di sviluppo

**18.** Verrà chiesto se si desidera installare gli aggiornamenti. Per il momento, selezionare **Ricorda in seguito**



# Ambiente di sviluppo

## 19. Aprire il **Terminale** seguendo i passi seguenti



The image shows a screenshot of an Ubuntu desktop environment. On the left, the 'Attività' (Activities) overview is visible, showing a vertical dock with various application icons. A red circle highlights the 'Show Applications' icon (a 3x3 grid of dots) at the bottom of the dock. A red arrow points from this circle to a search bar in the application menu. The search bar contains the text 'terminale' and is also circled in red. Below the search bar, the 'Terminale' application icon (a black square with a white prompt character '>\_') is circled in red. Three red arrows point from the numbered instructions to these elements: the first arrow points to the application menu search bar, the second arrow points to the search bar, and the third arrow points to the 'Terminale' icon.

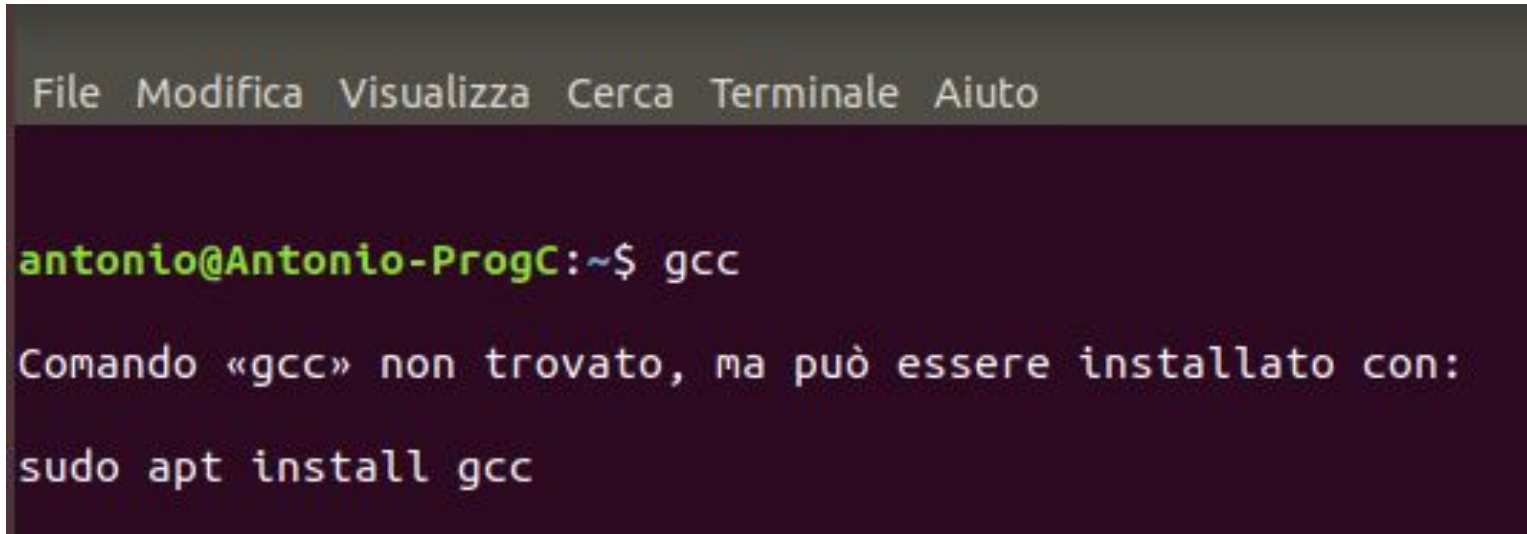
1. Cliccare sull'icona per aprire il menu delle applicazioni
2. Inserire "terminale"
3. Cliccare per aprire il *Terminale*



# Ambiente di sviluppo

**20.** Verificare se il compilatore è installato.

Per far ciò, inserire il comando **gcc**. Se compare il messaggio seguente, dare il comando **sudo apt install gcc** (come suggerito dalla shell) e inserire, se richiesta, la propria password (inserita in fase di installazione di Ubuntu). Attendere la fine dell'installazione (**N.B. Connessione ad Internet richiesta!**)



```
File Modifica Visualizza Cerca Terminale Aiuto

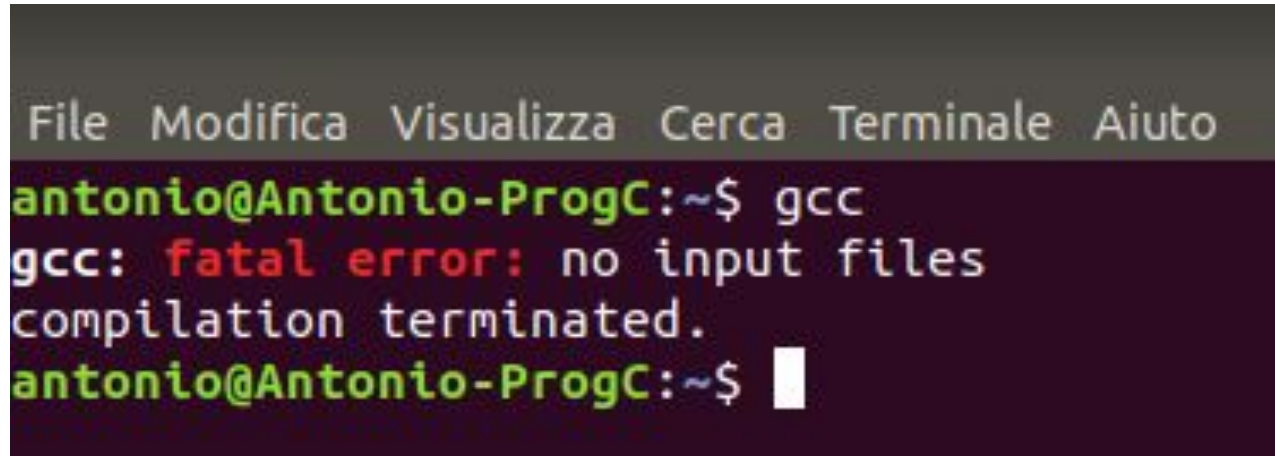
antonio@Antonio-ProgC:~$ gcc

Comando «gcc» non trovato, ma può essere installato con:

sudo apt install gcc
```

# Ambiente di sviluppo

**21.** Digitare nuovamente il comando **gcc** e se tutto è andato a buon fine, dovrebbe comparire il seguente messaggio.

A screenshot of a terminal window with a dark background. At the top, there is a menu bar with the following items: 'File', 'Modifica', 'Visualizza', 'Cerca', 'Terminale', and 'Aiuto'. Below the menu bar, the terminal shows a prompt 'antonio@Antonio-ProgC:~\$' followed by the command 'gcc'. The next line shows the error message 'gcc: fatal error: no input files' in red text, followed by 'compilation terminated.' on the next line. The prompt 'antonio@Antonio-ProgC:~\$' appears again at the end of the line, followed by a white cursor block.

```
File  Modifica  Visualizza  Cerca  Terminale  Aiuto
antonio@Antonio-ProgC:~$ gcc
gcc: fatal error: no input files
compilation terminated.
antonio@Antonio-ProgC:~$
```

Il messaggio di errore indica che non si è specificato nessun file per la compilazione (va bene così!!)

A questo punto, il compilatore è installato correttamente.