

# Competenze Tecniche e AI per l'Innovazione

## Corso di 80 ore

Questo percorso di 80 ore è pensato per sviluppare competenze concrete e immediatamente spendibili nel mondo del lavoro, fornendo basi solide nei sistemi operativi, nella programmazione e nell'utilizzo consapevole dell'intelligenza artificiale. L'obiettivo non è soltanto imparare a usare strumenti digitali, ma comprendere come funzionano e come integrarli in modo efficace in un contesto professionale.

Durante il corso si lavorerà in ambiente Windows e Linux, acquisendo familiarità con l'organizzazione dei sistemi, la gestione dei file e dei processi e la strutturazione di un progetto tecnico. Parallelamente verrà introdotto Python, una versione basica, linguaggio versatile e molto richiesto, attraverso esercitazioni pratiche orientate alla risoluzione di problemi reali.

Un elemento distintivo del percorso è l'integrazione dell'intelligenza artificiale come supporto allo sviluppo: gli studenti impareranno a utilizzare strumenti AI per comprendere meglio il codice, migliorare la documentazione, analizzare soluzioni e rafforzare il proprio metodo di lavoro, mantenendo sempre un approccio critico e responsabile.

Al termine delle 80 ore, i partecipanti avranno acquisito autonomia tecnica, capacità organizzativa e una visione moderna dello sviluppo digitale, competenze fondamentali per affrontare con maggiore sicurezza le sfide dell'innovazione tecnologica.

## Abstract corso (250 caratteri)

Corso di 80 ore dedicato allo sviluppo di competenze tecniche e digitali: Windows, Linux con WSL (Windows Subsystem for Linux, che permette di utilizzare un sistema Linux all'interno di Windows), Python e strumenti professionali integrati con l'intelligenza artificiale. Un percorso pratico e orientato al lavoro per costruire metodo, autonomia e capacità progettuale.

# **Note sull'opera, Edizione R 27/02/2026**

*A cura di: Loris LORO*

## **Abstract**

*In questo testo l'autore presenta il percorso di formazione del corso*

*In bibliografia/sitografia le fonti di informazione su cui si basa il testo.*

*Il curatore concede sotto **Licenza Creative Commons** l'utilizzo del presente documento come Opera di Ingegno, autorizzandone la pubblicazione sulla piattaforma: [www.picolab.eu](http://www.picolab.eu).*

*La pubblicazione del documento su altre piattaforme o l'estrazione di parte del testo da questo documento per l'impiego in pubblicazioni diverse è soggetta al consenso del curatore:  
[loro.loris@picolab.eu](mailto:loro.loris@picolab.eu)*

*Ingegnere elettronico opera a Torino dove vive tuttora; il curatore ha lavorato e lavora come formatore presso istituti pubblici e privati italiani che si occupano di informatica ed elettronica, ha collaborato e collabora con industrie e studi professionali in qualità di consulente.*

*Torino, venerdì 27 febbraio 2026*

***Sommario Competenze Tecniche e AI per l'Innovazione***

---

## **BIBLIOGRAFIA & Sitografia**

- Sito web di riferimento: [www.picolab.eu](http://www.picolab.eu), [www.picolab.it](http://www.picolab.it)
- [Uso di un PC con Windows 11](#): (link NON ATTIVO SUL WEB)

*Sommario Competenze Tecniche e AI per l'Innovazione*

---

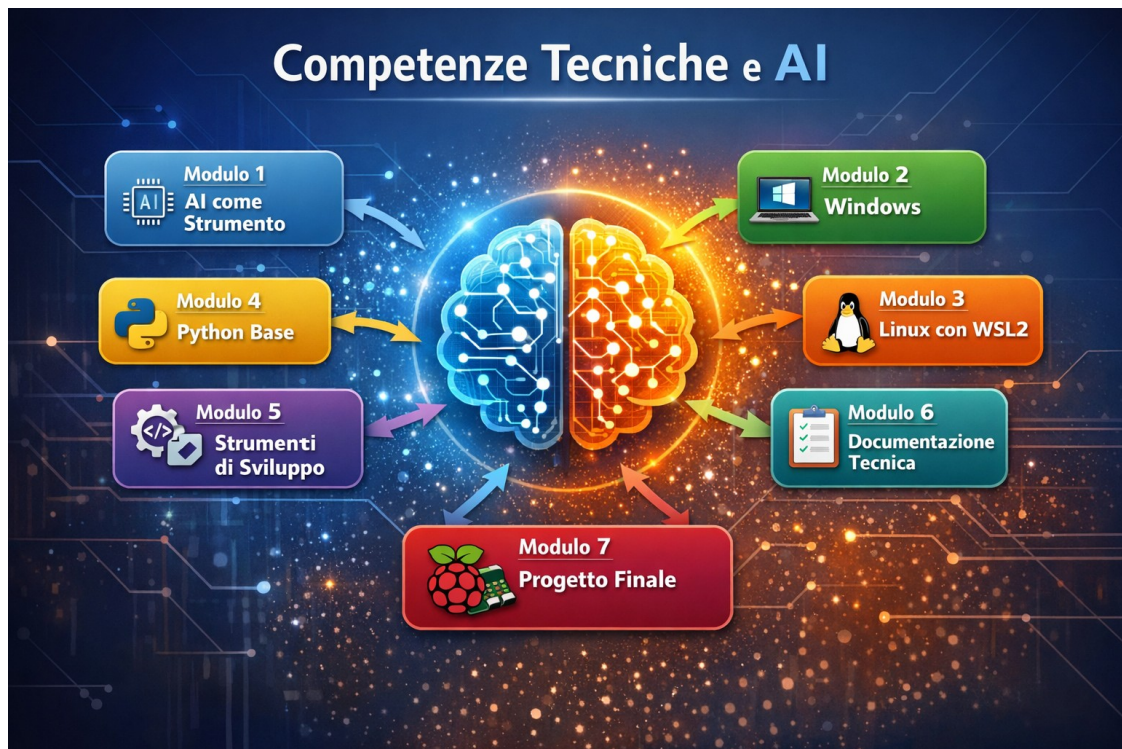
# Sommario Competenze Tecniche e AI per l’Innovazione

Corso di 80 ore – 20 lezioni da 4 ore

Il corso sviluppa competenze tecniche spendibili nel mondo del lavoro integrando sistemi operativi, programmazione Python, strumenti professionali e intelligenza artificiale come supporto costante all’apprendimento e allo sviluppo. La programmazione Python viene svolta utilizzando l’ambiente **Thonny**, scelto perché semplice, adatto ai neofiti e molto diffuso anche su Raspberry Pi.

---

Competenze Tecniche e AI per l’Innovazione.....	1
Abstract corso (250 caratteri).....	1
Note sull’opera, Edizione R 27/02/2026.....	2
BIBLIOGRAFIA & Sitografia.....	3
Sommario Competenze Tecniche e AI per l’Innovazione.....	4
Presentazione del percorso.....	5
Metodologia Didattica Operativa.....	6
Modulo 1 — Intelligenza Artificiale come Strumento di Apprendimento.....	7
Modulo 2 — Windows e Ambiente di Lavoro Digitale.....	7
Modulo 3 — Linux con WSL2.....	7
Modulo 4 — Python Base con Thonny.....	8
Modulo 5 — Strumenti Professionali di Sviluppo.....	8
Modulo 6 — Documentazione Tecnica.....	9
Modulo 7 — Progetto Finale su Raspberry Pi 3 B+.....	9
Requisiti del Progetto Finale (Raspberry Pi 3 B+).....	10
Distribuzione ore per modulo.....	10



### **Presentazione del percorso**

*La mappa concettuale rappresenta in modo sintetico la struttura del percorso formativo, articolato in sette blocchi progressivi e integrati.*

*Si parte dall'**Intelligenza Artificiale come strumento di apprendimento**, che accompagna l'intero corso, per poi consolidare le basi operative con **Windows** e **Linux con WSL2**. Il percorso prosegue con **Python Base**, cuore della parte applicativa, e con gli **Strumenti Professionali di Sviluppo**, che introducono metodo e organizzazione del lavoro.*

*La **Documentazione Tecnica** formalizza quanto appreso, mentre il **Progetto Finale su Raspberry Pi** rappresenta la sintesi concreta di tutte le competenze sviluppate.*

*L'insieme dei blocchi evidenzia un percorso coerente e orientato all'autonomia tecnica, alla capacità progettuale e all'uso consapevole dell'AI.*

## **Metodologia Didattica Operativa**

Ogni lezione da 4 ore segue una struttura ricorrente che integra l'uso consapevole dell'intelligenza artificiale.

### **1. Attivazione e contestualizzazione (30 minuti)**

Introduzione dell'argomento e definizione dell'obiettivo operativo. Gli studenti formulano una richiesta guidata all'AI per ottenere una spiegazione sintetica o un esempio pratico, imparando a costruire prompt chiari e contestualizzati.

### **2. Fondamenti teorici essenziali (60 minuti)**

Spiegazione del docente con dimostrazioni operative. Confronto tra spiegazione docente e output AI per sviluppare capacità critica e abitudine alla verifica.

### **3. Laboratorio guidato (90 minuti)**

Esercitazione pratica. L'AI può essere usata per suggerimenti e analisi errori, ma ogni proposta deve essere compresa e validata. Per Python, il laboratorio avviene in **Thonny** con esecuzione, debug e ispezione delle variabili.

### **4. Revisione e debugging (30 minuti)**

Analisi degli errori e miglioramento delle soluzioni. Uso dell'AI come supporto al debugging (lettura errori, ipotesi, verifiche), con controllo finale umano.

### **5. Documentazione tecnica (30 minuti)**

Aggiornamento del manuale personale in Markdown. L'AI supporta revisione e chiarezza, ma la responsabilità finale del testo resta dello studente.

### ***Sommario Competenze Tecniche e AI per l'Innovazione***

---

## **Modulo 1 — Intelligenza Artificiale come Strumento di Apprendimento**

Durata: 8 ore – 2 lezioni

### **Scopo del modulo**

Introdurre fin dall'inizio l'AI come strumento strutturato di studio e sviluppo, imparando a fare domande efficaci, a verificare le risposte e a usare l'AI per migliorare codice e documentazione senza dipendenza.

### **Lezione 1 (4 ore)**

AI generativa: cosa fa e cosa non fa, limiti, responsabilità. Prompt efficace, contesto e vincoli. Esercizi guidati di richiesta e verifica.

### **Lezione 2 (4 ore)**

AI per apprendimento e sviluppo: generazione di esempi controllati, spiegazione di errori, revisione di testi tecnici. Metodologia di “verifica a tre passi”: capire, testare, documentare.

### ***Sommario Competenze Tecniche e AI per l'Innovazione***

---

## **Modulo 2 — Windows e Ambiente di Lavoro Digitale**

Durata: 8 ore – 2 lezioni

### **Scopo del modulo**

Comprendere Windows come piattaforma tecnica di sviluppo e preparare l'ambiente di lavoro in modo ordinato e ripetibile.

### **Lezione 3 (4 ore)**

Architettura di Windows 11, gestione risorse, file system e percorsi, PowerShell e Windows Terminal, buone pratiche di organizzazione cartelle e progetto.

### **Lezione 4 (4 ore)**

Virtualizzazione e prerequisiti, introduzione a WSL2, installazione e configurazione di Ubuntu, differenza tra filesystem Windows e Linux, regole operative per lavorare in modo corretto.

### ***Sommario Competenze Tecniche e AI per l'Innovazione***

---

## **Modulo 3 — Linux con WSL2**

Durata: 16 ore – 4 lezioni

### **Scopo del modulo**

Sviluppare autonomia operativa in Linux per utilizzare strumenti tecnici, organizzare progetti e lavorare in modo “da laboratorio” con procedure ripetibili.

### **Lezione 5 (4 ore)**

Filesystem Linux, directory principali, navigazione da terminale, comandi base e gestione file.

### **Lezione 6 (4 ore)**

Permessi e utenti, lettura dei permessi, chmod e chown, casi tipici di errore e risoluzione.

### **Lezione 7 (4 ore)**

Processi e gestione del sistema: ps, top, esecuzione in background, segnali, introduzione concettuale ai servizi e ai log.

### **Lezione 8 (4 ore)**

Organizzazione professionale del progetto (struttura cartelle, naming), introduzione al versionamento (concetto), preparazione del repository per il lavoro successivo.

## ***Sommario Competenze Tecniche e AI per l'Innovazione***

---

### **Modulo 4 — Python Base con Thonny**

Durata: 24 ore – 6 lezioni

#### **Scopo del modulo**

Imparare Python in modo pratico e adatto ai neofiti, usando **Thonny** per scrivere, eseguire e fare debug. L'obiettivo è arrivare a programmi ordinati, comprensibili e facilmente testabili.

#### **Lezione 9 (4 ore)**

Installazione e configurazione di Thonny su Windows 11. Concetti base: script, esecuzione, variabili, tipi di dato. Uso della console integrata.

#### **Lezione 10 (4 ore)**

Input/output, stringhe, conversioni e validazione. Debug in Thonny: breakpoints, step-by-step, ispezione variabili.

#### **Lezione 11 (4 ore)**

Strutture di controllo: if/elif/else e while. Esercizi guidati con debug per comprendere flusso del programma.

#### **Lezione 12 (4 ore)**

Cicli for, range, iterazioni. Prime simulazioni numeriche semplici (ad esempio “valore che tende a un setpoint”).

#### **Lezione 13 (4 ore)**

Funzioni e modularità: parametri, return, organizzazione del codice. Introduzione alla scrittura “pulita” (nomi, commenti, piccoli test manuali).

#### **Lezione 14 (4 ore)**

Liste e dizionari, gestione di dati strutturati, JSON. Lettura e scrittura su file con esempi orientati allo “stato di un sistema”.

## ***Sommario Competenze Tecniche e AI per l'Innovazione***

---

### **Modulo 5 — Strumenti Professionali di Sviluppo**

Durata: 12 ore – 3 lezioni

## **Scopo del modulo**

Consolidare un metodo di lavoro professionale: ambienti, dipendenze, versionamento e primi servizi applicativi. Thonny resta l'IDE per Python, mentre gli strumenti di progetto si gestiscono da terminale.

### **Lezione 15 (4 ore)**

Gestione pacchetti e dipendenze: pip, requirements.txt, differenza tra librerie e progetto. Collegamento tra ciò che si fa in Thonny e l'ambiente di esecuzione (interprete, percorsi, moduli).

### **Lezione 16 (4 ore)**

Git operativo: init, add, commit, branch, messaggi di commit utili. Procedura minima di lavoro "da laboratorio" (salvataggio frequente e tracciabilità).

### **Lezione 17 (4 ore)**

Introduzione alle API (concetto e utilità), logging e gestione errori. Mini-servizio dimostrativo semplice e documentabile (focus su chiarezza, non su complessità).

## ***Sommario Competenze Tecniche e AI per l'Innovazione***

---

## **Modulo 6 — Documentazione Tecnica**

Durata: 4 ore – 1 lezione

### **Scopo del modulo**

Imparare a scrivere documentazione chiara e ripetibile: ciò che è stato fatto, come rifarlo, come verificare che funzioni, come risolvere i problemi tipici.

### **Lezione 18 (4 ore)**

Scrittura tecnica essenziale, Markdown, struttura di un manuale, checklist operative. Introduzione a MkDocs per organizzare la documentazione del progetto.

## ***Sommario Competenze Tecniche e AI per l'Innovazione***

---

## **Modulo 7 — Progetto Finale su Raspberry Pi 3 B+**

Durata: 8 ore – 2 lezioni

### **Scopo del modulo**

Applicare in modo integrato le competenze acquisite realizzando un progetto funzionante su hardware reale. Il Raspberry Pi 3 B+ diventa la "macchina di produzione" su cui installare, configurare ed eseguire il progetto, mantenendo tracciabilità e documentazione. Thonny viene utilizzato anche su Raspberry (dove è comunemente disponibile) oppure in remoto tramite file trasferiti e testati.

### **Lezione 19 (4 ore)**

Installazione Raspberry Pi OS, configurazione rete e accesso SSH. Preparazione ambiente Python e installazione dipendenze. Trasferimento progetto (git clone o copia controllata). Verifica esecuzione del programma su Pi.

## Lezione 20 (4 ore)

Completamento progetto e consegna: esecuzione affidabile, test funzionali, gestione log, documentazione completa del processo di installazione e deploy. Uso dichiarato dell'AI per revisione, debugging e qualità della documentazione.

### *Sommario Competenze Tecniche e AI per l'Innovazione*

---

## Requisiti del Progetto Finale (Raspberry Pi 3 B+)

Il progetto finale deve includere:

- codice Python sviluppato e testato in Thonny (con uso del debug documentato almeno in un caso)
- struttura ordinata del progetto e repository Git aggiornato
- dipendenze dichiarate e riproducibili
- una guida di installazione e deploy con comandi e verifiche
- una sezione "Uso dell'AI" che descriva come è stata utilizzata e come sono state verificate le risposte

Esempi di progetto finale compatibili con Pi 3 B+:

- mini monitor di sistema con log e output periodico
- piccola applicazione con lettura/scrittura stato su file JSON
- servizio semplice avviabile manualmente e verificabile (l'autostart è un'estensione opzionale se il tempo lo consente)

### *Sommario Competenze Tecniche e AI per l'Innovazione*

---

## Distribuzione ore per modulo

<b>Modulo</b>	<b>Ore</b>	<b>Percentuale sul totale (80 ore)</b>
AI come Strumento di Apprendimento	8	10%
Windows e Ambiente di Lavoro	8	10%
Linux con WSL2	16	20%
Python Base con Thonny	24	30%
Strumenti Professionali di Sviluppo	12	15%
Documentazione Tecnica	4	5%
Progetto Finale su Raspberry Pi 3 B+	8	10%
Totale	80	100%

### *Sommario Competenze Tecniche e AI per l'Innovazione*

---