



**UNIVERSITÀ
DEGLI STUDI
DI UDINE**
HIC SUNT FUTURA

Linee guida speciali per l'uso di agenti di coding nella didattica dei docenti, nella ricerca scientifica e nelle attività tecnico-amministrative

UDAIN Working Group

Università degli Studi di Udine

Linee guida AI - Sistemi di IA per generazione, revisione, testing, documentazione ed esecuzione di codice

Versione: 1.0

Ambito: didattica dei docenti, ricerca scientifica, attività tecnico-amministrative e sviluppo software istituzionale

Principio cardine

L'IA può assistere la progettazione, la scrittura, la modifica, il testing e la documentazione del software. La decisione di accettare, integrare, utilizzare o rilasciare il codice resta umana e deve essere fondata su verifica, test, controlli di sicurezza e responsabilità proporzionati al rischio.



SOMMARIO

Linee guida speciali per l'uso di agenti di coding nella didattica dei docenti, nella ricerca scientifica e nelle attività tecnico-amministrative	1
Principio cardine	1
1. Premessa	5
Introduzione	5
Legenda degli acronimi	6
2. Introduzione	6
3. Ambito di applicazione	7
4. Metodologia	7
5. Linee guida comuni	8
5.1 Principio generale di responsabilità	8
5.2 Principio generale di trasparenza	9
5.3 Principio generale di verifica	9
5.4 Regola di ingresso del codice	10
6. Tabella di sintesi relativa alla didattica dei docenti	10
6.1 CDM01 - Dichiarare l'uso di un coding agent nella produzione didattica	11
6.2 CDM02 - Revisione e miglioramento di esempi di codice nei materiali didattici	11
6.3 CDL01 - Spiegare codice, algoritmi e procedure computazionali	12
6.4 CDA01 - Preparare esercizi, scenari, dataset sintetici e codice con errori	12
6.5 CDA02 - Generare test automatici per esercitazioni e laboratori	12
6.6 CDA03 - Supportare la valutazione automatizzata solo per aspetti oggettivi	13
6.7 CDA04 - Individuare uso non autorizzato di un coding agent da parte degli studenti	13
6.8 CDU01 - Definire le policy d'uso di un coding agent nel syllabus	14
6.9 CDV01 - Didattica per sviluppatori: uso di un coding agent per comprendere, confrontare e progettare codice	15
7. Tabella di sintesi relativa alla ricerca scientifica	15
7.1 CRV01 - Affidare a un coding agent la produzione autonoma di codice scientifico decisivo	16
7.2 CRV02 - Falsificazione, manipolazione o selezione opportunistica dei dati tramite un coding agent	16
7.3 CRV03 - Conferire dati personali, credenziali o repository riservati	17
7.4 CRA01 - Generare dati sintetici o script per dati sintetici	17
7.5 CRA02 - Generare codice per analisi scientifiche centrali	18
7.6 CRA03 - Verifica di sicurezza, licenze, plagio e originalità del codice	18
7.7 CRL01 - Generare codice sorgente, script, notebook e codice LaTeX	18
7.8 CRL02 - Debugging, refactoring e ottimizzazione mirata	19
7.9 CRL03 - Generare test, controlli di regressione e procedure di validazione	19
7.10 CRL04 - Supportare analisi dati e analisi statistica	20
7.11 CRL05 - Documentare codice, repository e workflow computazionali	20
7.12 CRM01 - Dichiarare l'uso di un coding agent in pubblicazioni e repository	20
8. Tabella di sintesi relativa alle attività tecnico-amministrative e allo sviluppo software istituzionale	21
8.1 CAM01 - Spiegare codice esistente a fini di studio, manutenzione o onboarding	22
8.2 CAM02 - Generare codice e markup documentale	22
8.3 CAL01 - Generare codice sorgente e frammenti di programma	22
8.4 CAL02 - Revisione, refactoring e ottimizzazione di codice esistente	23
8.5 CAL03 - Individuazione di bug (debugging assistito)	23
8.6 CAL04 - Generare test unitari e di integrazione	23
8.7 CAL05 - Generare query e script per l'analisi dati	24



8.8 CAL06 - Tradurre codice tra linguaggi o framework (porting).....	24
8.9 CAL07 - Documentare codice e API tecniche.....	24
8.10 CAA01 - Generare codice per sistemi in produzione o a supporto di processi decisionali amministrativi	25
8.11 CAA02 - Individuare e correggere vulnerabilità di sicurezza	25
8.12 CAA03 - Generare script con accesso a sistemi informativi dell'Ateneo	26
9. Template	26
9.1 Modello breve per materiale didattico.....	26
9.2 Modello per syllabus	26
9.3 Modello per repository di ricerca	26
9.4 Modello per repository o applicativi tecnico-amministrativi	27
9.5 Modello per pubblicazione scientifica.....	27
10. Misure di sicurezza	27
10.1 Misure minime	27
10.2 Misure per didattica	27
10.3 Misure per attività tecnico-amministrative	28
10.4 Misure per ricerca.....	28
11. Glossario	28
12. Regola finale	29
13. Riferimenti essenziali	29
13.1 Normativa	29
13.2 Linee guida e standard tecnici.....	30
13.3 Documenti istituzionali di riferimento.....	30
13.4 Editoria scientifica e integrità della ricerca.....	30
13.5 Criterio di priorità	31
14. Nota di rilascio – versione 1.0	31
15. Dichiarazione sull'uso di IA	32

1. Premessa

Le presenti linee guida sono da considerarsi speciali rispetto alla Policy generale di Ateneo sull'uso dei sistemi di intelligenza artificiale e integrano le linee guida dedicate alla didattica dei docenti, alla ricerca scientifica ed allo sviluppo di soluzioni software per l'amministrazione.

La disciplina è tecnologicamente neutra: si applica a qualsiasi sistema di IA utilizzato per generare, modificare, spiegare, documentare, verificare, testare o eseguire codice, indipendentemente dal produttore, dal modello, dall'interfaccia o dal servizio impiegato.

Rientrano, a titolo esemplificativo, strumenti quali Codex, Claude Code, Pi-Agent, OpenCode e altri AI coding assistant, agenti autonomi o semi-autonomi, estensioni IDE, strumenti CLI e sistemi capaci di leggere file, modificare repository, eseguire comandi o proporre pull request.

Introduzione

Gli strumenti di assistenza e di generazione di codice (possono supportare la scrittura, revisione, spiegazione, documentazione, testing e ottimizzazione del codice. Il loro utilizzo è particolarmente rilevante in ambito universitario perché il codice può essere, a seconda dei casi:

- oggetto di insegnamento;
- strumento didattico;
- mezzo per apprendere concetti disciplinari;
- infrastruttura della ricerca;
- componente essenziale di analisi scientifiche;
- elemento necessario per la riproducibilità dei risultati.

L'impiego di agenti di coding deve essere informato dai principi generali di trasparenza, sicurezza e responsabilità, con intervento umano a monte e a valle dell'utilizzo. Tali criteri sono espressamente richiamati nelle linee guida ricerca, che segnalano i rischi di bias, diffusione di dati personali e produzione di dati inesatti o falsi.



Legenda degli acronimi

Per facilitare la lettura, si riporta di seguito il significato delle sigle utilizzate nel documento.

Sigle degli indicatori (letti come Coding + Ambito + Livello di rischio + numero progressivo):

Sigla	Ambito	Livello di rischio
CDM	Didattica	Minimo
CDL	Didattica	Limitato
CDA	Didattica	Alto rischio
CDU	Didattica	Utilizzatori (regole per il syllabus)
CDV	Didattica	Sviluppatori (insegnamenti di programmazione/AI)
CRV	Ricerca	Vietato
CRA	Ricerca	Alto rischio
CRL	Ricerca	Limitato
CRM	Ricerca	Minimo
CAM	Amministrazione/sviluppo tecnico	Minimo
CAL	Amministrazione/sviluppo tecnico	Limitato
CAA	Amministrazione/sviluppo tecnico	Alto rischio

Esempio: **CRA02** si legge *Coding - Ricerca - Alto rischio - indicazione n. 2.*

Altre sigle e abbreviazioni:

Sigla/termine	Significato
UDAIN	Gruppo di lavoro ("Working Group") e ambiente di sperimentazione ("Sandbox") istituiti dalla Policy generale di Ateneo per il governo dell'uso dell'IA
GV01-GV04	Limiti generali all'impiego dei Sistemi di IA, stabiliti dalla Policy generale di Ateneo (§5), sempre vincolanti e non derogabili dalle presenti linee guida speciali
CISO	Chief Information Security Officer
NDA	Non-Disclosure Agreement (accordo di riservatezza)
SAST	Static Application Security Testing (analisi statica del codice)
DAST	Dynamic Application Security Testing (analisi dinamica del codice)
API	Application Programming Interface (interfaccia di programmazione)
DEVE / NON DEVE	Requisito obbligatorio / divieto assoluto (convenzione linguistica ISO/AGID)
DOVREBBE / NON DOVREBBE	Implicazioni da comprendere e ponderare prima di scegliere approcci alternativi
PUÒ / NON PUÒ	Facoltà, la cui applicazione è a discrezione di chi legge

2. Introduzione

Obiettivo delle presenti linee guida è consentire un uso vantaggioso di agenti di coding, salvaguardando:

- l'efficacia della formazione universitaria;
- il valore sostanziale della valutazione degli studenti;
- la qualità metodologica della ricerca;
- la sicurezza del codice;
- la protezione dei dati;
- la proprietà intellettuale;
- la riproducibilità dei risultati scientifici;
- la responsabilità umana sul prodotto finale.

Un agente di coding non deve essere trattato come autore, coautore, valutatore autonomo, responsabile tecnico o garante della correttezza scientifica. Va gestito come un ottimo strumento di supporto. La responsabilità resta sempre in capo al docente, al ricercatore, al gruppo di ricerca, al personale, al responsabile scientifico e, in generale, rimane legata all'”umano” che si occupa di quello sviluppo.

3. Ambito di applicazione

Le presenti linee guida si applicano a:

1. **personale docente e collaboratori didattici**, quando utilizzano assistenti di coding per preparare materiali didattici, esempi, esercizi, laboratori, test, feedback o attività formative;
2. **ricercatori, dottorandi, assegnisti, gruppi di ricerca e responsabili scientifici**, quando utilizzano assistenti di coding per generare, modificare, documentare, testare o analizzare codice connesso ad attività di ricerca;
3. **personale tecnico-amministrativo e sviluppatori dell'Ateneo**, quando utilizzano assistenti di coding per generare, revisionare, testare, tradurre o documentare codice, script o automazioni destinati ad applicativi gestionali, portali istituzionali, sistemi informativi o infrastrutture dell'Ateneo.

Sono inclusi:

- Agenti di coding con propria interfaccia grafica;
 - Agenti di coding con interfaccia testuale;
 - estensioni IDE o ambienti di sviluppo assistiti da AI;
 - strumenti analoghi per generazione, revisione o spiegazione di codice;
 - agenti AI che possano leggere file, modificare repository, eseguire comandi o proporre pull request.
-

4. Metodologia

La metodologia riprende l'impostazione delle linee guida già predisposte.

Per la **didattica dei docenti**, le indicazioni sono articolate in:

Sigla	Significato	Ambito
CDM	Coding agent - Didattica - rischio minimo	Usi ordinari, dichiarativi o redazionali.
CDL	Coding agent - Didattica - rischio limitato	Usi didattici che richiedono verifica, ma non incidono direttamente sulla valutazione certificativa.
CDA	Coding agent - Didattica - alto rischio	Usi che incidono su esercizi, valutazioni, dati degli studenti, materiali complessi o attività valutative.
CDU	Coding agent - Didattica - utilizzatori	Regole per insegnamenti in cui l'agente di coding è usato come strumento applicativo.



Sigla	Significato	Ambito
CDV	Coding agent - Didattica - sviluppatori	Regole per insegnamenti che riguardano programmazione, AI, data science, sistemi software o sviluppo.

Per la **ricerca scientifica**, le indicazioni sono articolate in:

Sigla	Significato	Ambito
CRV	Coding agent - Ricerca - vietato	Usi incompatibili con integrità scientifica, privacy, sicurezza o responsabilità.
CRA	Coding agent - Ricerca - alto rischio	Usi ammessi solo con cautele rafforzate e validazione.
CRL	Coding agent - Ricerca - rischio limitato	Usi ordinari ammessi con verifica e documentazione proporzionata.
CRM	Co Coding agent dex - Ricerca - rischio minimo	Usi dichiarativi, documentali o di supporto non sostanziale.

Per le **attività tecnico-amministrative e lo sviluppo software istituzionale**, le indicazioni sono articolate in:

Sigla	Significato	Ambito
CAM	Coding agent - Amministrazione - rischio minimo	Usi ordinari di comprensione, studio o formattazione documentale, senza impatto su sistemi istituzionali.
CAL	Coding agent - Amministrazione - rischio limitato	Usi ordinari di sviluppo che richiedono verifica tecnica, ma non toccano direttamente sistemi in produzione o infrastrutture dell'Ateneo.
CAA	Coding agent - Amministrazione - alto rischio	Usi che incidono su sistemi in produzione, processi decisionali amministrativi, sicurezza informatica o infrastrutture istituzionali.

Questo terzo ambito richiama, per coerenza sistematica, i limiti generali già stabiliti dalla Policy generale di Ateneo (§5), in particolare:

- GV01 (divieto di funzionalità agentic senza supervisione umana su risorse dell'Ateneo),
- GV02 (divieto di conferire informazioni confidenziali e credenziali),
- GV03 (divieto di elaborare codice altrui protetto da licenza incompatibile)
- GV04 (divieto di conferire dati personali).

La classificazione del rischio tiene conto di quattro criteri:

Criterio	Domanda guida
Impatto formativo o scientifico	L'output incide su apprendimento, valutazione, risultati scientifici o pubblicazioni?
Accesso a dati o repository	L'agente di coding riceve dati personali, dati di ricerca, codice non pubblico o materiali riservati?
Autonomia operativa	L'agente di coding si limita a proporre testo/codice oppure può modificare file, eseguire comandi o operare su repository
Verificabilità	Lo sviluppatore può comprendere, testare, spiegare e difendere il risultato?

5. Linee guida comuni

5.1 Principio generale di responsabilità

Lo sviluppatore può utilizzare un agente di coding, ma resta responsabile di:

- accuratezza del codice;
- correttezza metodologica;
- sicurezza;



- qualità didattica;
- idoneità alla valutazione;
- rispetto di licenze e copyright;
- protezione dei dati;
- tracciabilità dell'uso.

5.2 Principio generale di trasparenza

L'uso di un agente di coding deve essere dichiarato quando contribuisce in modo significativo alla produzione di:

- materiali didattici;
- esempi o esercizi;
- prove valutative;
- codice scientifico;
- script di analisi;
- notebook;
- repository;
- software rilasciato;
- pubblicazioni scientifiche.

L'impostazione è coerente con le linee guida didattiche, che prevedono la dichiarazione dell'uso di IA nella produzione didattica e distinguono gli usi meramente redazionali dagli usi generativi sostanziali.

5.3 Principio generale di verifica

Nessun codice generato o modificato da un agente di coding deve essere utilizzato senza verifica umana.

La verifica deve essere proporzionata al rischio e può includere:

- lettura del codice;
- esecuzione in ambiente controllato;
- test unitari, di regressione, end-to-end;
- verifica delle dipendenze;
- controllo di sicurezza;
- controllo licenze;
- validazione dei risultati su dati noti;
- confronto con implementazioni indipendenti.



5.4 Regola di ingresso del codice

Il codice prodotto con un coding agent può entrare in materiale didattico, repository o pubblicazione solo se il responsabile umano è in grado di:

1. comprenderlo;
2. spiegarlo;
3. modificarlo;
4. testarlo;
5. giustificarne l'uso;
6. verificarne sicurezza e licenze;
7. assumersene la responsabilità.

6. Tabella di sintesi relativa alla didattica dei docenti

Indicatore	Indicazione dell'attività in sintesi	Descrizione dell'attività	Classificazione
CDM01	Dichiarare l'uso di un coding agent nella produzione didattica	Se il docente usa un coding agent per generare o modificare esempi di codice, esercizi, test, script o materiali tecnici, deve darne comunicazione chiara quando l'uso è significativo.	MINIMO
CDM02	Revisione e miglioramento di esempi di codice nei materiali didattici	Un coding agent è usato per migliorare leggibilità, commenti, formattazione, chiarezza o struttura di esempi già predisposti dal docente, senza aggiungere contenuto sostanziale nuovo.	MINIMO
CDL01	Spiegare codice, algoritmi e procedure computazionali	Un coding agent supporta la produzione di spiegazioni, pseudocodice, walkthrough, commenti didattici e confronti tra implementazioni.	LIMITATO
CDA01	Preparare esercizi, scenari, dataset sintetici e codice con errori	Un coding agent genera esercizi, casi di studio, bug intenzionali, scenari applicativi, dataset sintetici o tracce di laboratorio.	ALTO RISCHIO
CDA02	Generare test automatici per esercitazioni e laboratori	Un coding agent produce test unitari, casi limite, test di autovalutazione o verifiche automatiche per attività didattiche.	ALTO RISCHIO
CDA03	Supportare la valutazione automatizzata solo per aspetti oggettivi	Un coding agent può assistere nella verifica di aspetti tecnici oggettivi, come compilazione, sintassi, presenza di funzioni richieste o superamento di test, ma non decide l'esito della valutazione.	ALTO RISCHIO
CDA04	Individuare uso non autorizzato di un coding agent da parte degli studenti	Il docente valuta incongruenze, discontinuità stilistiche, complessità inattesa del codice, errori non coerenti con il percorso svolto. Gli AI detector non sono prova sufficiente.	ALTO RISCHIO
CDU01	Definire le policy d'uso di un coding agent nel syllabus	Il docente stabilisce se un coding agent è ammesso, per quali attività, con quali limiti, e come deve essere dichiarato dagli studenti.	LIMITATO
CDV01	Didattica per sviluppatori: uso di un coding agent per comprendere, confrontare e progettare codice	Negli insegnamenti di programmazione, informatica, AI, data science o discipline affini, un coding agent può essere oggetto e strumento di analisi critica, progettazione, confronto e testing.	LIMITATO / ALTO RISCHIO



6.1 CDM01 - Dichiarare l'uso di un coding agent nella produzione didattica

L'indicazione rappresenta l'applicazione del principio di trasparenza alla produzione di materiali didattici che includano codice, script, notebook, formule LaTeX, test automatici o esempi computazionali.

Il docente **PUÒ** utilizzare un coding agent per predisporre materiali didattici, a condizione che:

- verifichi integralmente il codice;
- corregga eventuali errori;
- provi gli esempi prima di proporli agli studenti;
- segnali l'uso quando il contributo di un coding agent è sostanziale.

Esempio di dichiarazione breve

Alcuni esempi di codice utilizzati nei materiali didattici sono stati predisposti con il supporto di un coding agent. Il docente ha verificato e modificato gli output prima della pubblicazione.

ATTENZIONE. Non è necessario dichiarare l'uso di strumenti di formattazione automatica, completamento banale o correzione ortografica, purché non vi sia generazione sostanziale di codice o contenuti didattici.

6.2 CDM02 - Revisione e miglioramento di esempi di codice nei materiali didattici

Un coding agent può essere usato come assistente redazionale e tecnico per migliorare:

- leggibilità del codice;
- commenti;
- coerenza stilistica;
- denominazione di variabili;
- struttura di funzioni;
- esempi minimi;
- documentazione inline.

Il docente **DEVE** verificare che la revisione non abbia alterato il significato tecnico, l'obiettivo didattico o il livello di difficoltà previsto.

ATTENZIONE. Se un coding agent aggiunge nuove soluzioni, esempi, algoritmi o spiegazioni sostanziali, si applicano le regole sulla produzione didattica e sulla dichiarazione d'uso.



6.3 CDL01 - Spiegare codice, algoritmi e procedure computazionali

Un coding agent può essere utile per spiegare frammenti di codice, trasformare codice in pseudocodice, generare commenti esplicativi o proporre confronti tra soluzioni alternative.

Il docente **PUÒ** usare un coding agent per:

- preparare spiegazioni progressive;
- generare esempi per studenti con diversi livelli di competenza;
- mostrare errori comuni;
- confrontare implementazioni;
- creare walkthrough da discutere in aula.

Il docente **DEVE** controllare che la spiegazione sia corretta, non semplifichi impropriamente il concetto e non introduca terminologia fuorviante.

6.4 CDA01 - Preparare esercizi, scenari, dataset sintetici e codice con errori

Un coding agent può generare esercizi, scenari applicativi, casi di debugging, dataset sintetici o codice volutamente errato.

L'attività è classificata ad **alto rischio** perché può incidere direttamente sulla qualità della didattica, sulla difficoltà delle prove e sull'equità della valutazione.

Il docente **PUÒ** usare un coding agent per:

- generare tracce di esercizi;
- creare codice con bug da individuare;
- produrre dataset sintetici per esercitazioni;
- differenziare esercizi per livelli di difficoltà;
- creare casi di test per laboratorio.

Il docente **DEVE**:

- verificare la correttezza della traccia;
 - controllare che la difficoltà sia coerente con il corso;
 - evitare dati personali o realistici riconducibili a persone;
 - segnalare eventuale uso significativo;
 - provare la soluzione attesa.
-

6.5 CDA02 - Generare test automatici per esercitazioni e laboratori

Un coding agent può generare test unitari, test di regressione, casi limite e script di autovalutazione.

L'uso è ammesso, ma deve essere trattato con cautela, perché test incompleti o mal costruiti possono fornire agli studenti un'indicazione errata della correttezza del codice.

Il docente **DEVE**:

- verificare i test;
 - controllare che non siano eccessivamente restrittivi o permissivi;
 - distinguere tra test di autovalutazione e test valutativi;
 - evitare che il test automatico diventi unico criterio di valutazione quando sono richieste competenze progettuali, argomentative o metodologiche.
-

6.6 CDA03 - Supportare la valutazione automatizzata solo per aspetti oggettivi

Un coding agent può assistere il docente nella valutazione solo per profili oggettivamente verificabili, quali:

- compilazione;
- esecuzione;
- superamento di test;
- rispetto di una firma di funzione;
- presenza di file richiesti;
- correttezza sintattica;
- rispetto di formati.

Un coding agent **NON DEVE** decidere autonomamente:

- voto finale;
- originalità del contributo;
- correttezza concettuale complessiva;
- qualità progettuale;
- sanzioni per uso non autorizzato;
- attribuzione di plagio o misconduct.

Il docente resta l'unico responsabile dell'esito.

6.7 CDA04 - Individuare uso non autorizzato di un coding agent da parte degli studenti

L'uso di un coding agent per individuare condotte non autorizzate è ad **alto rischio**.

Il docente può osservare:

- salti improvvisi di qualità;

- uso di strutture non trattate a lezione;
- commenti generici o incongruenti;
- codice non spiegabile dallo studente;
- errori incompatibili con il percorso svolto;
- riferimenti a librerie o API non introdotte.

Gli strumenti automatici di detection **NON DEVONO** essere usati come prova unica o decisiva.

In caso di sospetto, la via ordinaria è il dialogo didattico:

- richiesta di spiegare il codice;
- modifica controllata di una parte;
- discussione orale;
- ricostruzione dei passaggi;
- confronto con le regole del syllabus.

Questa impostazione riprende il criterio già presente nelle linee guida didattiche: il sospetto di uso non autorizzato deve essere affrontato con verifica critica e dialogo, non con affidamento meccanico agli AI detector.

6.8 CDU01 - Definire le policy d'uso di un coding agent nel syllabus

Il docente deve chiarire nel syllabus o nelle istruzioni dell'attività:

Elemento	Indicazione
Ammissibilità	Un coding agent è consentito, vietato o consentito solo per alcune attività?
Finalità consentite	Debugging, spiegazione, autovalutazione, generazione di test, consultazione?
Finalità vietate	Soluzione integrale, uso durante prove chiuse, mancata dichiarazione?
Dichiarazione d'uso	Quando e come lo studente deve dichiarare l'uso?
Verifica	Lo studente deve saper spiegare, modificare e difendere il codice?
Conseguenze	Quali effetti ha l'uso non dichiarato o non autorizzato?

Formula esemplificativa

Nel presente insegnamento l'uso di un coding agent e strumenti analoghi è consentito per attività di studio, spiegazione del codice, debugging e autovalutazione, salvo diversa indicazione nelle prove valutative. Non è consentito presentare come interamente proprio codice generato in modo sostanziale da strumenti di IA, salvo esplicita autorizzazione. Ogni uso rilevante deve essere dichiarato. Lo studente deve essere in grado di spiegare e modificare il codice consegnato.

6.9 CDV01 - Didattica per sviluppatori: uso di un coding agent per comprendere, confrontare e progettare codice

Negli insegnamenti rivolti a studenti che devono comprendere o progettare sistemi software, algoritmi, modelli di AI o pipeline computazionali, un coding agent può essere usato come oggetto di studio e come strumento didattico.

Il docente **PUÒ** utilizzare un coding agent per:

- confrontare soluzioni generate e soluzioni umane;
- analizzare errori del modello;
- discutere vulnerabilità;
- valutare qualità del codice;
- mostrare limiti dei sistemi generativi;
- progettare attività di prompt engineering tecnico;
- sviluppare capacità di review critica.

Il rischio diventa **alto** quando un coding agent:

- accede a repository reali;
- usa dataset non pubblici;
- esegue codice;
- modifica file;
- viene usato in prove valutative;
- produce output che incidono sul voto.

7. Tabella di sintesi relativa alla ricerca scientifica

Indicatore	Indicazione dell'attività in sintesi	Descrizione dell'attività	Classificazione
CRV01	Affidare a un coding agent la produzione autonoma di codice scientifico decisivo	Un coding agent non può essere responsabile autonomo del codice su cui si fondano risultati scientifici, analisi centrali o conclusioni pubblicate.	VIETATO
CRV02	Falsificazione, manipolazione o selezione opportunistica dei dati tramite un coding agent	Un coding agent non può essere usato per alterare dati, selezionare risultati in modo opportunistico o rendere più efficace artificialmente un'argomentazione scientifica.	VIETATO
CRV03	Conferire dati personali, credenziali o repository riservati	Non devono essere inseriti dati personali, credenziali, chiavi API, repository coperti da NDA o codice non conferibile.	VIETATO
CRA01	Generare dati sintetici o script per dati sintetici	Un coding agent può supportare la generazione di dati sintetici o del codice che li produce, ma con validazione metodologica e dichiarazione.	ALTO RISCHIO
CRA02	Generare codice per analisi scientifiche centrali	Un coding agent produce codice che incide direttamente su risultati, figure, statistiche, classificazioni o inferenze scientifiche.	ALTO RISCHIO



Indicatore	Indicazione dell'attività in sintesi	Descrizione dell'attività	Classificazione
CRA03	Verifica di sicurezza, licenze, plagio e originalità del codice	Il team usa un coding agent o altri strumenti per supportare controlli su sicurezza, licenze, similitudini o dipendenze.	ALTO RISCHIO
CRL01	Generare codice sorgente, script, notebook e codice LaTeX	Un coding agent genera parti di software, codice eseguibile, formule LaTeX, tabelle o script di supporto.	LIMITATO
CRL02	Debugging, refactoring e ottimizzazione mirata	Un coding agent aiuta a individuare bug, migliorare struttura, rendere più leggibile o ottimizzare il codice.	LIMITATO
CRL03	Generare test, controlli di regressione e procedure di validazione	Un coding agent produce test unitari, controlli di coerenza, script di validazione e casi limite.	LIMITATO
CRL04	Supportare analisi dati e analisi statistica	Un coding agent aiuta a scrivere codice per analisi descrittive, inferenziali, esplorative o predittive sotto supervisione esperta.	LIMITATO
CRL05	Documentare codice, repository e workflow computazionali	Un coding agent produce README, docstring, istruzioni di installazione, commenti o workflow riproducibili.	LIMITATO
CRM01	Dichiarare l'uso di un coding agent in pubblicazioni e repository	L'uso di un coding agent è descritto nella sezione metodologica, nel repository, nei metadati o nei materiali supplementari.	MINIMO

7.1 CRV01 - Affidare a un coding agent la produzione autonoma di codice scientifico decisivo

Un coding agent **NON PUÒ** essere usato come sostituto del ricercatore nella produzione autonoma di codice decisivo per risultati scientifici.

È vietato:

- generare codice centrale senza comprenderlo;
- pubblicare risultati ottenuti da codice non validato;
- usare un coding agent come garante della correttezza metodologica;
- attribuire a un coding agent responsabilità scientifica.

Il codice può essere assistito da un coding agent solo se il gruppo di ricerca lo verifica, testa e assume piena responsabilità sul risultato.

7.2 CRV02 - Falsificazione, manipolazione o selezione opportunistica dei dati tramite un coding agent

Un coding agent **NON DEVE** essere utilizzato per:

- alterare dati;
- eliminare outlier senza giustificazione metodologica;
- generare risultati fittizi;
- modificare output per renderli più coerenti con l'ipotesi;
- produrre script finalizzati a nascondere risultati negativi;

- manipolare figure, tabelle o statistiche.

Questa indicazione riprende il divieto, presente nelle linee guida ricerca, di usare sistemi di IA per manipolare o alterare dati al fine di rafforzare l'argomentazione scientifica.

7.3 CRV03 - Conferire dati personali, credenziali o repository riservati

È vietato inserire in un coding agent:

- dati personali;
- dati particolari;
- dati di soggetti coinvolti in ricerca;
- file con credenziali;
- token;
- chiavi API;
- certificati;
- repository riservati;
- codice coperto da NDA;
- materiale non pubblicato non conferibile a terzi;
- dati o codice soggetti a vincoli contrattuali.

L'uso può essere valutato solo in ambiente autorizzato, con base giuridica, misure tecniche adeguate e autorizzazione del responsabile scientifico o della struttura competente.

7.4 CRA01 - Generare dati sintetici o script per dati sintetici

Un coding agent può essere usato per generare script che producono dati sintetici o per supportare la costruzione di dataset artificiali.

L'attività è ad **alto rischio** perché può incidere su:

- validità dei modelli;
- rappresentatività;
- bias;
- interpretabilità dei risultati;
- replicabilità;
- confronto con dati reali.

Il team **DEVE**:

- documentare la procedura;
- descrivere il metodo di generazione;
- verificare le distribuzioni;

- confrontare dati sintetici e dati reali, ove possibile;
 - dichiarare l'uso nei materiali di ricerca.
-

7.5 CRA02 - Generare codice per analisi scientifiche centrali

Quando un coding agent genera codice che produce direttamente risultati scientifici, figure, classificazioni, stime statistiche o conclusioni, l'uso è ad **alto rischio**.

Il team **DEVE** applicare una validazione rafforzata:

Controllo	Descrizione
Review del codice	Lettura da parte di almeno un ricercatore competente.
Test su dati noti	Esecuzione su dataset controllati o risultati attesi.
Confronto indipendente	Se possibile, confronto con altra implementazione o software consolidato.
Tracciabilità	Conservazione di versione, prompt rilevanti, commit e modifiche.
Riproducibilità	Ambiente, dipendenze e parametri devono essere documentati.
Dichiarazione	L'uso deve essere indicato se incide sul metodo o sui risultati.

7.6 CRA03 - Verifica di sicurezza, licenze, plagio e originalità del codice

Un coding agent o strumenti analoghi possono aiutare a individuare:

- vulnerabilità;
- dipendenze non sicure;
- codice duplicato;
- possibili problemi di licenza;
- funzioni obsolete;
- errori ricorrenti.

Tale uso resta ad **alto rischio**, perché un coding agent può non riconoscere correttamente licenze, vulnerabilità o similitudini sostanziali.

Il controllo umano e, quando necessario, l'uso di strumenti specializzati restano indispensabili.

7.7 CRL01 - Generare codice sorgente, script, notebook e codice LaTeX

Un coding agent può essere usato per generare:

- script di supporto;
- funzioni ausiliarie;
- notebook esplorativi;
- codice LaTeX;
- tabelle;
- grafici;



- procedure di import/export;
- trasformazioni di formato.

L'uso è **limitato** quando non incide direttamente sulle conclusioni scientifiche o quando il codice viene verificato integralmente.

Questa indicazione è coerente con il caso d'uso ricerca relativo alla generazione di codice sorgente, codice eseguibile, formule LaTeX e tabelle complesse.

7.8 CRL02 - Debugging, refactoring e ottimizzazione mirata

Un coding agent può essere utilizzato per:

- individuare bug;
- spiegare errori;
- proporre refactoring;
- migliorare leggibilità;
- ridurre duplicazioni;
- suggerire ottimizzazioni.

Il ricercatore **DEVE** verificare che la modifica:

- non alteri la logica scientifica;
 - non cambi i risultati attesi;
 - non introduca vulnerabilità;
 - non comprometta la riproducibilità;
 - sia coperta da test adeguati.
-

7.9 CRL03 - Generare test, controlli di regressione e procedure di validazione

L'uso di un coding agent per generare test è raccomandabile quando aumenta la controllabilità del codice.

Può includere:

- unit test;
- test di regressione;
- test su casi limite;
- controlli di coerenza;
- validazione di input/output;
- script per riprodurre risultati.

Il team deve evitare che test generati da un coding agent siano considerati sufficienti senza revisione: un test errato può confermare codice errato.

7.10 CRL04 - Supportare analisi dati e analisi statistica

Un coding agent può supportare la scrittura di codice per:

- pulizia dati;
- visualizzazione;
- analisi descrittive;
- analisi inferenziali;
- modelli predittivi;
- pipeline computazionali.

L'uso è ammesso solo sotto supervisione esperta, con modello correttamente specificato e verifica dei risultati.

Questa indicazione riprende i casi d'uso ricerca relativi all'analisi dei dati e all'analisi statistica con IA, che richiedono supervisione esperta e corretta specificazione dei modelli.

7.11 CRL05 - Documentare codice, repository e workflow computazionali

Un coding agent può essere usato per predisporre:

- README;
- docstring;
- commenti;
- guide di installazione;
- esempi di esecuzione;
- istruzioni per riprodurre analisi;
- workflow computazionali;
- descrizione delle dipendenze.

Il ricercatore deve verificare che la documentazione sia accurata, aggiornata e coerente con il repository.

7.12 CRM01 - Dichiarare l'uso di un coding agent in pubblicazioni e repository

L'uso di un coding agent deve essere dichiarato quando contribuisce in modo significativo a:

- codice scientifico;



- analisi dati;
- produzione di figure;
- workflow computazionali;
- software rilasciato;
- repository allegati a pubblicazioni;
- metodi riproducibili.

La dichiarazione può essere inserita in:

- sezione metodi;
 - acknowledgements;
 - data/code availability statement;
 - README;
 - CHANGELOG;
 - release notes;
 - materiali supplementari.
-

8. Tabella di sintesi relativa alle attività tecnico-amministrative e allo sviluppo software istituzionale

Indicatore	Indicazione dell'attività in sintesi	Descrizione dell'attività	Classificazione
CAM01	Spiegare codice esistente a fini di studio, manutenzione o onboarding	Un coding agent spiega codice scritto da altri, a supporto della comprensione o della presa in carico di un progetto esistente.	MINIMO
CAM02	Generare codice e markup documentale	Un coding agent produce formattazione, tabelle complesse o tipografia per documenti tecnici o amministrativi.	MINIMO
CAL01	Generare codice sorgente e frammenti di programma	Un coding agent scrive funzioni, boilerplate o script a supporto di attività tecnico-amministrative ordinarie.	LIMITATO
CAL02	Revisione, refactoring e ottimizzazione di codice esistente	Un coding agent migliora, ristruttura o rende più efficiente codice già esistente.	LIMITATO



Indicatore	Indicazione dell'attività in sintesi	Descrizione dell'attività	Classificazione
CAL03	Individuazione di bug (debugging assistito)	Un coding agent aiuta a individuare errori funzionali in codice esistente.	LIMITATO
CAL04	Generare test unitari e di integrazione	Un coding agent produce suite di test a partire dal codice esistente.	LIMITATO
CAL05	Generare query e script per l'analisi dati	Un coding agent produce query SQL, notebook o script di elaborazione dati amministrativi o gestionali.	LIMITATO
CAL06	Tradurre codice tra linguaggi o framework (porting)	Un coding agent converte codice esistente da un linguaggio o framework a un altro.	LIMITATO
CAL07	Documentare codice e API tecniche	Un coding agent produce commenti, README o documentazione di interfacce tecniche.	LIMITATO
CAA01	Generare codice per sistemi in produzione o a supporto di processi decisionali amministrativi	Un coding agent scrive codice destinato ad applicativi gestionali, portali istituzionali o strumenti a supporto di decisioni amministrative.	ALTO RISCHIO
CAA02	Individuare e correggere vulnerabilità di sicurezza	Un coding agent supporta la revisione di sicurezza e il secure coding su codice esposto a rischi (SQL injection, gestione credenziali, ecc.).	ALTO RISCHIO
CAA03	Generare script con accesso a sistemi informativi dell'Ateneo	Un coding agent produce script, automazioni o connettori che interagiscono con banche dati, API interne o infrastrutture istituzionali.	ALTO RISCHIO

8.1 CAM01 - Spiegare codice esistente a fini di studio, manutenzione o onboarding

Attività a rischio minimo, analoga per finalità alla spiegazione di codice già prevista in ambito didattico (CDL01): un coding agent aiuta a comprendere, non a validare autonomamente la correttezza.

Chi sviluppa **PUÒ** utilizzare un coding agent per farsi spiegare codice esistente, tenendo presente che la spiegazione fornita non sostituisce la verifica diretta del comportamento del codice nei casi critici o quando la comprensione è propedeutica a una modifica su sistemi in produzione (v. CAA01).

8.2 CAM02 - Generare codice e markup documentale

Coerente con il caso d'uso ricerca CRL01, attività a basso impatto data la natura non eseguibile su sistemi terzi del markup documentale.

Chi sviluppa **PUÒ** utilizzare un coding agent per generare codice o markup, verificando la corretta resa tipografica del documento finale prima della diffusione o pubblicazione.

8.3 CAL01 - Generare codice sorgente e frammenti di programma

Coerentemente con l'indicazione già prevista per la ricerca (CRL01), l'uso di un coding agent per scrivere parti di software è ammesso come attività di supporto ordinaria anche in ambito tecnico-amministrativo.



Chi sviluppa **PUÒ** utilizzare un coding agent per generare codice, rispettando le seguenti indicazioni:

- il codice generato **DEVE** essere compreso e verificato prima dell'integrazione, non limitandosi a un controllo di superficie sull'esecuzione;
 - **NON DEVE** essere integrato codice altrui protetto da licenza incompatibile (CRV03, per estensione);
 - **DOVREBBE** essere indicata, ove opportuno, l'origine un coding agent nel codice o nel repository (es. commit message), coerentemente con il principio generale di trasparenza (§5.2);
 - il codice **DOVREBBE** essere sempre conservato in repository e sistemi di controllo versione che consentano di analizzare le differenze tra le varie versioni per consentire l'analisi di quanto modificato (dai coding agent e/o dall'umano);
-

8.4 CAL02 - Revisione, refactoring e ottimizzazione di codice esistente

Chi sviluppa **PUÒ** utilizzare un coding agent per revisionare e ottimizzare codice esistente, rispettando le seguenti indicazioni:

- le modifiche proposte **DEVONO** essere confrontate con il comportamento atteso del codice originale, per escludere regressioni;
 - **DOVREBBE** essere eseguita la suite di test esistente dopo ogni modifica assistita da un coding agent.
-

8.5 CAL03 - Individuazione di bug (debugging assistito)

Chi sviluppa **PUÒ** utilizzare un coding agent per individuare errori funzionali, rispettando le seguenti indicazioni:

- la correzione proposta **DEVE** essere verificata rispetto alla causa reale dell'errore, non semplicemente accettata perché elimina il sintomo;
 - **DOVREBBE** essere mantenuta traccia del ragionamento che ha portato alla correzione, utile in caso di errori ricorrenti su sistemi condivisi.
-

8.6 CAL04 - Generare test unitari e di integrazione

Chi sviluppa **PUÒ** generare test con un coding agent, rispettando le seguenti indicazioni:

- i test generati **DEVONO** essere eseguiti e verificati per assicurare che coprano effettivamente i casi rilevanti, non solo quelli banali;



- **DOVREBBE** essere evitata una copertura test puramente nominale (test che non falliscono mai per costruzione), coerentemente con quanto già indicato per la ricerca (CRL03);
 - il codice dei test **DOVREBBE** essere sempre conservato in repository e sistemi di controllo versione che consentano di analizzare le differenze tra le varie versioni per consentire l'analisi di quanto modificato (dai coding agent e/o dall'umano);
-

8.7 CAL05 - Generare query e script per l'analisi dati

Chi sviluppa **PUÒ** generare query e script di analisi con un coding agent, rispettando le seguenti indicazioni:

- un coding agent **NON PUÒ** ricevere in input dati personali o riservati non anonimizzati (CRV03);
 - le query/script di analisi ed i loro risultati **DEVONO** essere verificati da chi li sta progettando;
 - i risultati **DEVONO** essere verificati rispetto alla logica sottostante, specie quando alimentano report o cruscotti gestionali;
-

8.8 CAL06 - Tradurre codice tra linguaggi o framework (porting)

Chi sviluppa **PUÒ** utilizzare un coding agent per il porting di codice, rispettando le seguenti indicazioni:

- il codice tradotto **DEVE** essere testato per verificarne l'equivalenza funzionale con l'originale, poiché la traduzione automatica può introdurre differenze semantiche non evidenti tra linguaggi;
 - restano fermi i vincoli di licenza sul codice di origine (CRV03);
 - il sistema finale dovrà essere testato in un ambiente di staging prima essere distribuito in sistemi di produzione;
-

8.9 CAL07 - Documentare codice e API tecniche

Chi sviluppa **PUÒ** generare documentazione con un coding agent, rispettando la seguente indicazione:

- la documentazione generata **DEVE** essere verificata per accuratezza rispetto al comportamento reale del codice, evitando disallineamenti tra codice e commenti che possano indurre in errore chi lo manutiene in seguito.
-



8.10 CAA01 - Generare codice per sistemi in produzione o a supporto di processi decisionali amministrativi

Il profilo di rischio è alto perché il codice destinato ad applicativi gestionali o a strumenti a supporto di decisioni amministrative ricade nei requisiti minimi già individuati dalla Policy generale per l'uso di strumenti digitali a supporto dei processi decisionali (§1.2 Policy): tracciabilità, supervisione umana effettiva, ricostruibilità dei criteri, replicabilità.

Chi sviluppa **PUÒ** utilizzare un coding agent per generare codice destinato alla produzione, rispettando le seguenti indicazioni:

- il codice generato **DEVE** essere sottoposto a revisione umana integrale prima del rilascio, non limitata al solo funzionamento apparente;
- **DEVE** essere testato secondo le procedure ordinarie di quality assurance previste per il contesto (ad es. test unitari, di integrazione, end to end, umani di accettazione);
- **DEVE** essere documentata la componente generata con un coding agent, coerentemente con il principio generale di trasparenza (§5.2);
- **NON DEVE** essere collegato a processi che formano graduatorie, ammissioni/esclusioni o attribuzione di benefici senza le garanzie di cui alla Policy generale §1.2.
- il codice del sistema e dei test **DEVE** essere sempre conservato in repository e sistemi di controllo versione che consentano di analizzare le differenze tra le varie versioni per consentire l'analisi di quanto modificato (dai coding agent e/o dall'umano);

ATTENZIONE. Resta fermo il divieto GV01 della Policy generale: un coding agent non può operare in autonomia su tali sistemi (es. deploy automatico, modifica diretta di configurazioni di produzione) senza supervisione umana.

8.11 CAA02 - Individuare e correggere vulnerabilità di sicurezza

L'uso di un coding agent per attività di secure coding si inserisce nel quadro delle Linee guida AGID per lo sviluppo sicuro, che raccomandano pratiche idonee a ridurre le classi di vulnerabilità note e l'impiego di test come l'analisi statica (SAST) e dinamica (DAST) prima del rilascio.

Chi sviluppa **PUÒ** avvalersi di un coding agent per l'individuazione di vulnerabilità, rispettando le seguenti indicazioni:

- le correzioni suggerite **DEVONO** essere verificate da personale competente prima dell'applicazione;
- **NON DEVE** essere condiviso con un coding agent codice contenente credenziali, chiavi API, token di accesso o configurazioni di sicurezza (CRV03);
- **DOVREBBE** essere privilegiato, ove disponibile, l'uso di strumenti di analisi statica/dinamica dedicati rispetto alla sola revisione conversazionale, quando la criticità del sistema lo richiede.



8.12 CAA03 - Generare script con accesso a sistemi informativi dell'Ateneo

Script e connettori che interagiscono con banche dati o infrastrutture istituzionali comportano un rischio elevato per la sicurezza informatica (NIS2/D.Lgs. 138/2024), poiché un errore nel codice generato può produrre accessi non autorizzati o perdita di dati.

Chi sviluppa **PUÒ** generare tali script con il supporto di un coding agent, rispettando le seguenti indicazioni:

- il codice **DEVE** essere eseguito in ambiente di test prima di qualunque esecuzione su sistemi di produzione;
 - un coding agent **NON PUÒ** ricevere in input credenziali, stringhe di connessione o dati reali estratti dai sistemi (CRV03);
-

9. Template

9.1 Modello breve per materiale didattico

Alcuni esempi di codice contenuti in questo materiale sono stati predisposti con il supporto di un coding agent / [altro strumento]. Il docente ha verificato, modificato e validato i contenuti prima della loro diffusione.

9.2 Modello per syllabus

Nel presente insegnamento l'uso di un coding agent e strumenti analoghi è [consentito / consentito con limiti / non consentito] per le attività di [specificare]. L'uso è ammesso per supporto alla comprensione, debugging, spiegazione del codice e autovalutazione, salvo diversa indicazione nelle prove valutative. Ogni uso rilevante deve essere dichiarato. Lo studente deve essere in grado di spiegare, modificare e difendere il codice consegnato.

9.3 Modello per repository di ricerca

In questo repository è stato utilizzato un coding agent / [altro strumento] per [debugging / refactoring / generazione di test / documentazione / prototipazione]. Ogni output è stato verificato dal gruppo di ricerca prima dell'integrazione. Gli autori mantengono piena responsabilità sul codice e sui risultati prodotti.



9.4 Modello per repository o applicativi tecnico-amministrativi

In questo repository/applicativo è stato utilizzato un coding agent / [altro strumento] per [generazione codice / debugging / refactoring / generazione test / documentazione]. Il codice è stato revisionato, testato e validato dal personale tecnico competente prima del rilascio in produzione. La responsabilità sul codice rilasciato resta in capo allo sviluppatore e alla struttura di riferimento.

9.5 Modello per pubblicazione scientifica

During the preparation of the code and computational materials associated with this work, the authors used un coding agent / [tool name] for [debugging / code generation / test generation / documentation / refactoring]. All AI-assisted outputs were reviewed, tested and, where necessary, revised by the authors, who assume full responsibility for the code, methods and results.

Durante la preparazione del codice e dei materiali computazionali associati a questo lavoro, gli autori hanno utilizzato un coding agent / [nome strumento] per [debugging / generazione di codice / generazione di test / documentazione / refactoring]. Tutti gli output assistiti da IA sono stati verificati, testati e, ove necessario, modificati dagli autori, che assumono piena responsabilità per il codice, i metodi e i risultati.

10. Misure di sicurezza

10.1 Misure minime

Prima di usare un coding agent, verificare che il materiale non contenga:

Elemento	Regola
Password	Non inserire mai.
Token	Non inserire mai.
Chiavi API	Non inserire mai.
Certificati	Non inserire mai.
Dati personali	Non inserire salvo base giuridica e ambiente autorizzato.
Dati di ricerca non pubblici	Non inserire salvo autorizzazione del responsabile scientifico.
Repository riservati	Non conferire senza verifica di licenze, contratti e autorizzazioni.
Codice con licenze incompatibili	Verificare prima dell'uso e del rilascio.

10.2 Misure per didattica

- usare dati fittizi o sintetici;
- evitare dati reali di studenti;
- testare sempre codice ed esercizi;
- chiarire le regole nel syllabus;
- evitare valutazioni interamente automatizzate;
- prevedere spiegazioni orali o modifiche in presenza quando serve verificare l'apporto individuale.



10.3 Misure per attività tecnico-amministrative

- eseguire in ambiente di test prima di qualunque rilascio in produzione;
- non conferire a un coding agent credenziali, stringhe di connessione, token o chiavi API (v. CRV03, per estensione);
- sottoporre a revisione umana integrale il codice destinato a sistemi in produzione o a processi decisionali amministrativi;
- privilegiare, per il codice critico dal punto di vista della sicurezza, strumenti di analisi statica/dinamica dedicati (SAST/DAST) accanto alla revisione assistita da un coding agent;
- documentare nel repository o nel codice la componente generata con un coding agent, coerentemente con il principio generale di trasparenza (§5.2).

10.4 Misure per ricerca

- usare repository puliti da segreti;
- separare codice pubblico e codice riservato;
- documentare prompt e modifiche rilevanti;
- usare versionamento;
- eseguire test;
- verificare dipendenze;
- controllare licenze;
- validare risultati scientifici con metodi indipendenti quando l'uso è centrale.

11. Glossario

Termine	Definizione
Coding agent	Strumento di AI coding assistant capace di supportare attività di generazione, revisione, spiegazione, testing e documentazione del codice.
AI coding assistant	Sistema di IA che assiste nello sviluppo software, proponendo codice, correzioni, spiegazioni o test (alcuni nomi commerciali sono: Codex, Claude Code, Antigravity, Pi-Agent, OpenCode, ecc.)
Repository	Archivio di codice, documentazione, dati e cronologia/versionamento delle modifiche.
Prompt	Istruzione fornita al sistema di IA.
Output	Risultato generato dal sistema di IA.
Refactoring	Modifica della struttura del codice senza alterarne la funzionalità attesa.
Test di regressione	Test volto a verificare che una modifica non abbia introdotto errori in funzionalità già esistenti.
Dato sintetico	Dato generato artificialmente per simulare caratteristiche di dati reali.
Validazione	Processo di verifica della correttezza del codice o del risultato rispetto a obiettivi, requisiti e dati noti.

12. Regola finale

Un coding agent può essere utilizzato nella didattica dei docenti, nella ricerca scientifica e nelle attività tecnico-amministrative quando aumenta chiarezza, qualità, efficienza, verificabilità, sicurezza e riproducibilità.

Non deve essere utilizzato quando:

- sostituisce il giudizio del docente, del ricercatore o dello sviluppatore;
- compromette la valutazione delle competenze;
- produce codice non compreso;
- espone dati personali, credenziali o materiali riservati;
- introduce rischi di sicurezza in sistemi in produzione o infrastrutture istituzionali;
- altera dati o risultati;
- rende opaco il contributo umano;
- impedisce di difendere il metodo o il codice;
- opera in autonomia, senza supervisione umana, su risorse dell'Ateneo (GV01 della Policy generale).

Un coding agent può aiutare a scrivere, spiegare, testare e documentare codice. Non può assumere responsabilità didattica, scientifica, tecnica, amministrativa o giuridica.

13. Riferimenti essenziali

Le presenti linee guida sono state elaborate a partire da tre ordini di fonti: la normativa europea e nazionale applicabile all'IA, gli standard tecnici e le linee guida istituzionali per lo sviluppo sicuro del software, e i documenti di riferimento già adottati in ambito accademico (Policy generale di Ateneo, linee guida di altri Atenei, standard editoriali sull'integrità della ricerca).

13.1 Normativa

- Regolamento (UE) 2024/1689 (AI Act), in particolare artt. 3, 10, 14, 16, 26, 50, 73.
- Legge 23 settembre 2025, n. 132, *Disposizioni e deleghe al Governo in materia di intelligenza artificiale*.
- Regolamento (UE) 2016/679 (GDPR), in particolare artt. 5, 6, 9, 22, 32, 33.
- Direttiva (UE) 2022/2555 (NIS2), recepita con D.Lgs. 4 settembre 2024, n. 138.
- Decreto Legislativo 7 marzo 2005, n. 82, *Codice dell'Amministrazione Digitale* (CAD).
- Legge 22 aprile 1941, n. 633, *Protezione del diritto d'autore*.
- Decreto Legislativo 10 febbraio 2005, n. 30, *Codice della Proprietà Industriale*, art. 98 (segreti commerciali).

- D.P.R. 10 gennaio 1957, n. 3, art. 15 (segreto d'ufficio).

13.2 Linee guida e standard tecnici

- AGID, *Linee guida per l'adozione dell'intelligenza artificiale nella Pubblica Amministrazione*, Determinazione 17/2025.
- AGID, *Linee guida per lo sviluppo dell'intelligenza artificiale nella Pubblica Amministrazione*, Determinazione 43/2026 – in particolare le indicazioni su sviluppo sicuro del codice, analisi statica (SAST) e dinamica (DAST), gestione della supply chain.
- AGID, *Linee guida per il procurement di IA nella Pubblica Amministrazione*, Determinazione 43/2026.
- AGID, *Linee guida per lo sviluppo del software sicuro*.
- ISO/IEC 23894:2023 – *Information technology – Artificial intelligence – Guidance on risk management*.
- ISO/IEC 42001:2023 – *Information technology – Artificial intelligence – Management system*.
- ISO/IEC 5338:2023 – *Information technology – Artificial intelligence – AI system life cycle processes*.
- ISO/IEC TR 24028:2020 – *Information technology – Artificial intelligence (AI) – Overview on trustworthiness*.
- NIST AI Risk Management Framework (nist.gov/itl/ai-risk-management-framework).

13.3 Documenti istituzionali di riferimento

- Università degli Studi di Udine, **Policy generale sull'uso di strumenti di intelligenza artificiale** (v. 2.5), in particolare §1.2 (contesto accademico), §2.1 (principi P1-P10), §4.1 (approccio risk-based), §5 (limiti generali GV01-GV04).
- Università degli Studi di Udine, *Linee guida IA – Ricerca* (v. 1.1) – fonte diretta dei casi d'uso ripresi e adattati nella sezione 7 (in particolare RL08, RL13 sulla generazione di codice e la protezione dei dati nei prompt).
- Università degli Studi di Udine, *Linee guida IA – Didattica dei Docenti* – fonte diretta dei casi d'uso ripresi e adattati nella sezione 6 (in particolare D06/DA03 su esercizi e codice con errori, e il criterio di verifica critica per l'uso non autorizzato).

13.4 Editoria scientifica e integrità della ricerca

- COPE (Committee on Publication Ethics), *Authorship and AI tools*.
- ICMJE (International Committee of Medical Journal Editors), *Use of Artificial Intelligence in Publishing*.
- Elsevier, *The use of generative AI and AI-assisted technologies in writing for Elsevier*.

- European Commission, *Living guidelines on the responsible use of generative AI in research*.

13.5 Criterio di priorità

In caso di conflitto tra le fonti sopra elencate: (1) prevalgono sempre i limiti generali della Policy di Ateneo (GV01-GV04) e le eventuali condizioni contrattuali della fornitura Codex/ChatGPT negoziata dall'Ateneo; (2) in subordine, valgono le indicazioni AGID e la normativa nazionale/europea; (3) la letteratura di settore ha funzione interpretativa e di raffinamento, non sostitutiva.

14. Nota di rilascio – versione 1.0

Data: 6 ottobre 2026 **Stato:** prima versione consolidata, a valle delle precisazioni operative sulla fornitura Codex (funzionalità di ChatGPT) fornita da UniUD.

In risposta ai chiarimenti richiesti in fase di redazione, si dà atto di quanto segue:

1. **Sandbox non ancora attivata.** L'ambiente controllato "UNIUD Sandbox", richiamato dalla Policy generale (§6) e citato nelle presenti linee guida (es. CAA01, CAA03) come sede per la sperimentazione di modalità agentic o di connessione a sistemi istituzionali, **non è a oggi operativo**. Fino alla sua attivazione, ogni riferimento alla Sandbox nel presente documento ha carattere prospettico e non costituisce un canale di autorizzazione effettivamente disponibile.
2. **Divieto di base per la connessione a banche dati interne.** Per impostazione predefinita, è **vietato collegare un coding agent a banche dati interne** dell'Ateneo. Questo dato **rafforza** quanto già previsto in CAA03 (che ammetteva la generazione di script con accesso a sistemi informativi solo con misure cautelative) e **richiede una revisione formale dell'indicazione**: in assenza di un'autorizzazione specifica rilasciata secondo le procedure che verranno definite con l'attivazione della Sandbox, l'uso descritto in CAA03 è da considerarsi, allo stato attuale, **non consentito** piuttosto che "ad alto rischio con cautele". Si segnala questa discrepanza testuale come punto da correggere nella prossima revisione, per evitare che la formulazione attuale di CAA03 sia letta come un'autorizzazione condizionata già operativa.
3. **Clausole contrattuali sulla fornitura.** Il contratto di fornitura Codex/ChatGPT prevede clausole di **non addestramento** sui dati conferiti dagli utenti e una **policy di data retention definita**. Questo elemento riduce il rischio che i prompt e il codice condivisi vengano utilizzati per addestrare modelli di terzi o restino disponibili a tempo indeterminato, e rafforza la base di affidamento per gli usi ordinari (CAM, CAL, CDM, CDL, CRL). **Non elimina tuttavia** gli obblighi di riservatezza su segreti d'ufficio, segreti commerciali e dati personali (GV02, GV04, CRV03): la non addestrabilità del modello non equivale a un'autorizzazione al trattamento, che richiede comunque base giuridica specifica per i dati personali e autorizzazione del titolare per le informazioni riservate. In altri sistemi, magari acquisiti individualmente, questa garanzia

potrebbe non essere scontata. Va quindi verificato caso per caso la compatibilità tra garanzia contrattuale e criticità del software che si sta sviluppando.

Prossimi passi per la versione 1.1: riformulare CAA01-CAA03 e i richiami a GV01 nella Premessa e nel §3, distinguendo esplicitamente tra l'ambiente sandbox di esecuzione di Codex (isolato) e l'eventuale connessione a repository o sistemi reali dell'Ateneo, tenendo conto dei tre punti sopra indicati.

15. Dichiarazione sull'uso di IA

Il presente documento è stato redatto con il supporto di uno strumento di intelligenza artificiale generativa (Claude Sonnet 5, Anthropic; ChatGPT 5.6, OpenAI), nell'ambito di un processo di elaborazione a guida e responsabilità umana. Il quadro concettuale e normativo di riferimento, la struttura del documento, i vincoli applicativi e ogni scelta di merito sono stati definiti dall'autore; la stesura testuale delle singole indicazioni è stata prodotta dallo strumento di IA sulla base di tali indicazioni, ed è stata successivamente verificata, corretta e approvata dall'autore in ciascuna delle sue fasi. La responsabilità sui contenuti resta pertanto interamente in capo all'autore umano.