

Insegnare informatica nel primo ciclo

Dalle Indicazioni alle attività in classe

Michael Lodi · Agnese Del Zozzo · Alberto Montresor · Giorgia Bissoli

Laboratorio "invitato" · ITADINFO 2026 · Padova, 10 ottobre 2026

[ITADINFO]

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

UNIVERSITÀ DI TRENTO

Verona FabLab

Da dove partiamo: Sulle orme di Milù

Mondadori Education, 2026

- **Informatica**: un fascicolo di 48 pagine per il primo triennio della primaria
- **Guida di Informatica** per l'insegnante: 120 pagine
- Materiali coerenti con le Indicazioni 2025, pensati anche per chi insegna senza una formazione informatica



Fascicolo per la classe



Guida per l'insegnante

La guida aperta: Insegnare Informatica



Per docenti del primo ciclo

- Versione derivata della guida Mondadori, con licenza **CC BY-NC-SA 4.0**
- **Autocontenuta**: si usa anche senza il sussidiario
- Pensata per la primaria, adatta a chiunque si avvicini per la prima volta all'informatica

informaticainclasse.it

scarica la guida

Per approfondire: gli atti di ITADINFO 2026



Sugli atti:

«Aprite il sussidiario»: dalla produzione editoriale a una guida "open" per insegnare Informatica nel primo ciclo

M. Lodi, A. Del Zozzo, A. Montresor, G. Bissoli

Oggi non lo presentiamo: racconta come sono nati fascicolo e guida, i principi che ci hanno guidato e la scelta di aprire la guida.

ITADINFO 2026 · Quarto Convegno Italiano sulla Didattica dell'Informatica · a cura di V. Lonati e G. Pozzan

Il laboratorio

1

Introduzione

2

Attività 1

Luci di Natale

Informazione e dati

3

Attività 2

Gruccia-bilancia

Algoritmi

4

Attività 3

Programmatori, esecutori e tangram

Programmazione

5

Debriefing

Le tre attività coprono le tre aree dell’informatica che le Indicazioni Nazionali 2025 introducono in Matematica.



Informatica nel primo ciclo

Le Indicazioni Nazionali 2025 e i concetti di base

Le Indicazioni Nazionali 2025

- Dall’a.s. 2026/2027 l’informatica entra nel primo ciclo, dentro Matematica e Tecnologia
- Distinguono esplicitamente le competenze **digitali** da quelle **informatiche**

In Matematica, tre aree

Programmazione

Attività 3

Programmatori, esecutori e tangram

Informazione e dati

Attività 1

Luci di Natale

Algoritmi

Attività 2

Gruccia-bilancia

Competenze digitali o informatiche?

Competenza digitale

«padronanza di uso efficace, sicuro e consapevole di dispositivi, strumenti e tecnologie digitali»

Indicazioni Nazionali 2025

Competenza informatica

Capire come i dispositivi rappresentano ed elaborano dati in modo automatico, seguendo procedimenti ideati dall'uomo

Insegnare Informatica, cap. 2

*L'astronomia non studia i telescopi, la biologia non studia i microscopi: l'informatica non studia i computer, ma **l'elaborazione automatica dell'informazione**.*

Gli obiettivi del laboratorio

1. Separare l’informatica dalle competenze digitali
2. Rendere espliciti programmatore, linguaggio di istruzioni ed esecutore
3. Distinguere informazione, dati e loro rappresentazione
4. Accompagnare il passaggio dall’esecuzione di istruzioni alla progettazione di programmi e algoritmi



Luci di Natale

Attività 1 · Informazione e dati: rappresentare informazioni

La storia di Alice e Bob



I messaggi di Alice e Bob

1. Ci vediamo oggi pomeriggio?
2. Non posso venire fuori perché non mi sento bene, resto a casa.
3. Porta la palla.
4. Andiamo a mangiare fuori?
5. Non posso venire fuori, però vieni a casa mia.
6. Ci vediamo domani.

- Due case una di fronte all'altra, in una valle di montagna
- Quando nevica molto, telefoni e Internet non funzionano
- Le luci di Natale sulle case formano **una lettera alla volta**

Cinque codici

Codice 1 tutte le lettere del messaggio	**Codice 2** una parola a caso	**Codice 3** l'iniziale	**Codice 4** un inizio prestabilito	**Codice 5** una sigla concordata
CI VEDIAMO OGGI POMERIGGIO	*dipende dalla parola scelta*	C	CI VEDIAMO O	CP
NON POSSO VENIRE FUORI PERCHÉ NON MI SENTO BENE RESTO A CASA		N	NON POSSO VENIRE FUORI PERCHÉ	NB
PORTA LA PALLA		P	P	PP
ANDIAMO A MANGIARE FUORI		A	A	AM
NON POSSO VENIRE FUORI PERÒ VIENI A CASA MIA		N	NON POSSO VENIRE FUORI PERÒ	NP
CI VEDIAMO DOMANI		C	CI VEDIAMO D	CD

Luci di Natale: analizzare i codici

? Domande stimolo

- Quali codici permettono ad Alice e Bob di capirsi sempre, senza mai sbagliarsi? Da cosa lo avete capito?
- Tra questi, quale permette di accendere meno lettere possibile? Come fate a dirlo?
- Riuscite a inventare un nuovo codice ancora più breve, ma in modo che Alice e Bob possano sempre capirsi?

Il nostro codice 6

Codice 6 una lettera concordata qualsiasi	**I messaggi di Alice e Bob**
A	Ci vediamo oggi pomeriggio?
B	Non posso venire fuori perché non mi sento bene, resto a casa.
C	Porta la palla.
D	Andiamo a mangiare fuori?
E	Non posso venire fuori, però vieni a casa mia.
F	Ci vediamo domani.

Univocità e costo

	Si capiscono sempre?	Lettere da accendere per i 6 messaggi (spazi esclusi)
Codice 1	sì	156
Codice 2	dipende	*dipende dalla parola scelta*
Codice 3	no	6
Codice 4	sì	70
Codice 5	sì	12
Nostro codice 6	sì	6

Luci di Natale: il codice più breve

! Aspetti a cui fare attenzione

- Basta un simbolo diverso per ogni messaggio, ad esempio le lettere dalla A alla F
- La scelta dei simboli è arbitraria, **purché l'associazione messaggio–simbolo sia condivisa**: senza legenda, i simboli da soli non «dicono» niente

Informazione, dati, rappresentazione

Informazione

Una conoscenza utile rispetto a un obiettivo

es. *Può partecipare all'attività «dai 10 anni in su»?*

Dati

Scelti per esprimerla: da soli non hanno significato

es. *2016 · 07/03/2016 · 10 anni*

Rappresentazione

Come i dati vengono codificati fisicamente

es. *Cifre sulla carta, bit nel computer, luci…*

La stessa informazione si può esprimere con dati diversi, e gli stessi dati si possono rappresentare in modi diversi: l'importante è mettersi d'accordo.

Luci di Natale: che cosa c’è di informatico

Informazione il messaggio che Alice e Bob vogliono comunicare

Dati le sequenze di lettere scelte per esprimerlo

Rappresentazione le luci accese sulle case

Progettare un codice è un compromesso tra **chiarezza** (univocità) e **brevità** (costo).

Nelle Indicazioni Nazionali 2025

COMPETENZE (FINE PRIMARIA)

- Esprimere informazioni mediante dati di varia natura e codificare tali dati anche digitalmente

OBIETTIVI (FINE TERZA PRIMARIA)

- Scegliere ed utilizzare oggetti o simboli per rappresentare informazioni

CONOSCENZE (PRIMARIA)

- Rappresentazione di dati e informazioni e loro differenza

COMPETENZE (FINE SECONDARIA DI PRIMO GRADO)

- Rappresentare informazioni in relazione al compito da svolgere

CONOSCENZE (SECONDARIA DI PRIMO GRADO)

- Sistema binario



Gruccia-bilancia

Attività 2 · Algoritmi: trovare il più pesante

Gruccia-bilancia: da dove viene

CS Unplugged

Bell, Witten, Fellows.
Attività 7: trovare il contenitore più leggero con una bilancia a due piatti

Traduzione italiana: R. Davoli, G.M. Bianco, P. Grossi

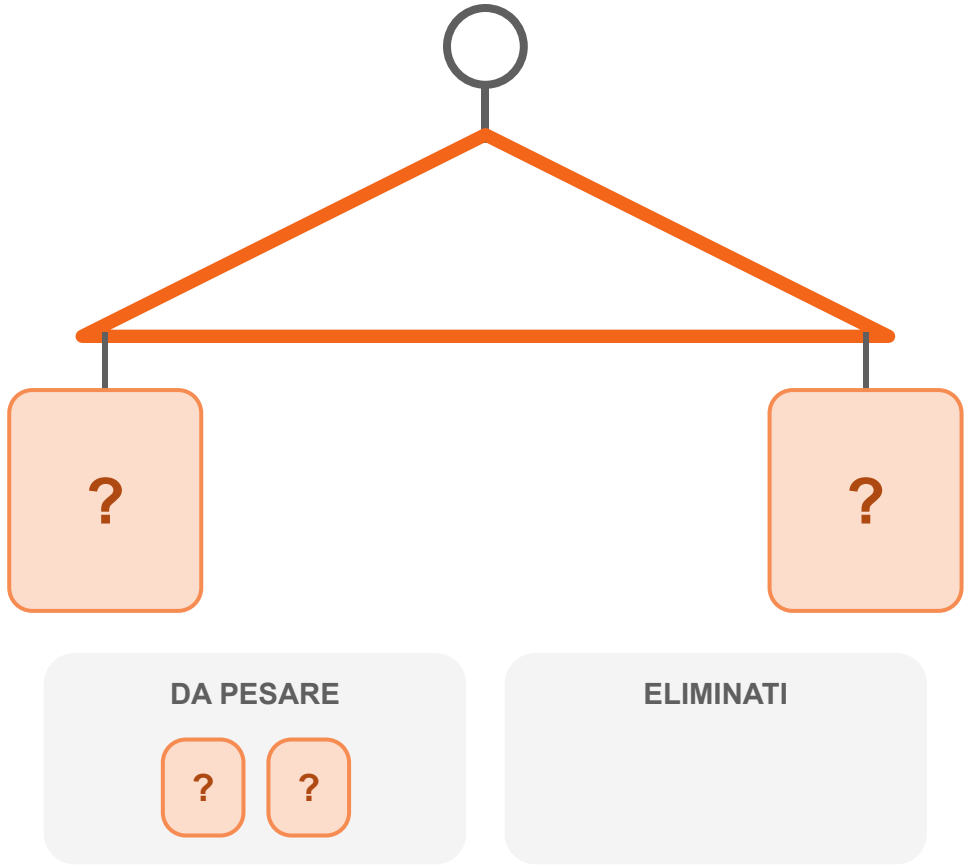
Informatica senza e con computer

Lodi, Davoli, Montanari, Martini (2020): il percorso per la primaria a cui si ispira l'attività

Nella guida (cap. 6, Attività 1) la bilancia diventa **una gruccia**: materiali a costo (quasi) zero.

Gruccia-bilancia: il problema

- 4 sacchetti opachi, apparentemente identici, con pesi tutti diversi
- Una gruccia usata come bilancia
- Due scatole: SACCHETTI DA PESARE e SACCHETTI ELIMINATI
- Si può avere in mano **un solo sacchetto** alla volta
- Alla fine, sulla gruccia deve restare **solo il più pesante**



Il Manuale *delle* istruzioni e i ruoli

MANUALE *DELLE* ISTRUZIONI (GRUCCIA-BILANCIA)
METTI IL SACCHETTO CHE HAI IN MANO NEI SACCHETTI ELIMINATI
APPENDI IL SACCHETTO CHE HAI IN MANO ALLA GRUCCIA
PRENDI IN MANO **UN SACCHETTO** DAI SACCHETTI DA PESARE
PRENDI IN MANO IL SACCHETTO **PIÙ LEGGERO** TRA I DUE SULLA GRUCCIA

REGOLA

In ogni momento puoi avere in mano **un solo sacchetto**.

OBIETTIVO

Alla fine, sulla gruccia deve restare **solo il sacchetto più pesante**.

Programmatore
pensa la strategia e dà le istruzioni

Esecutore
le esegue alla lettera

Controllore
controlla il rispetto delle regole

Attaccapanni
regge la gruccia

Gruccia-bilancia: perché funziona?

? Domande stimolo

- Dopo la prima pesata, quanti sacchetti potrebbero ancora essere il più pesante?
- Il sacchetto eliminato può essere il più pesante? Perché?
- Se ripetete sempre lo stesso ragionamento, che cosa succede dopo 3 eliminazioni?
- La strategia funziona anche con sacchetti di pesi diversi? E con più sacchetti?

Una strategia: il campione

- Sulla gruccia resta sempre un **campione**: il più pesante tra quelli visti finora
- Ogni nuovo sacchetto sfida il campione; il più leggero viene eliminato
- Con 4 sacchetti bastano 14 istruzioni in sequenza

```
 1. PRENDI IN MANO UN SACCHETTO DAI SACCHETTI DA PESARE
 2. APPENDI IL SACCHETTO CHE HAI IN MANO ALLA GRUCCIA
 3. PRENDI IN MANO UN SACCHETTO DAI SACCHETTI DA PESARE
 4. APPENDI IL SACCHETTO CHE HAI IN MANO ALLA GRUCCIA
 5. PRENDI IN MANO IL SACCHETTO PIÙ LEGGERO TRA I DUE SULLA GRUCCIA
 6. METTI IL SACCHETTO CHE HAI IN MANO NEI SACCHETTI ELIMINATI
 7. PRENDI IN MANO UN SACCHETTO DAI SACCHETTI DA PESARE
 8. APPENDI IL SACCHETTO CHE HAI IN MANO ALLA GRUCCIA
 9. PRENDI IN MANO IL SACCHETTO PIÙ LEGGERO TRA I DUE SULLA GRUCCIA
10. METTI IL SACCHETTO CHE HAI IN MANO NEI SACCHETTI ELIMINATI
11. PRENDI IN MANO UN SACCHETTO DAI SACCHETTI DA PESARE
12. APPENDI IL SACCHETTO CHE HAI IN MANO ALLA GRUCCIA
13. PRENDI IN MANO IL SACCHETTO PIÙ LEGGERO TRA I DUE SULLA GRUCCIA
14. METTI IL SACCHETTO CHE HAI IN MANO NEI SACCHETTI ELIMINATI
```

il primo campione

1ª sfida

2ª sfida

3ª sfida

E se…?

…i sacchetti fossero 100?

Scrivere ogni istruzione diventa lunghissimo: la ripetizione è necessaria

…due sacchetti pesassero uguale?

L'algoritmo deve prevedere che cosa fare, altrimenti l'esecutore non sa come comportarsi

…cambiassimo i pesi?

Una strategia corretta funziona sempre, non solo con i sacchetti di oggi

È un problema generale: la soluzione non è un sacchetto, ma un **procedimento** che funziona con qualsiasi dato.

Dalla sequenza alla ripetizione

- Nelle 14 istruzioni lo stesso blocco di 4 si ripete tre volte (3–6, 7–10, 11–14)
- Con 100 sacchetti servirebbero **398** istruzioni in sequenza
- Con una **ripetizione** il blocco si scrive una volta sola, per qualsiasi numero di sacchetti

Con la ripetizione

```
PRENDI IN MANO UN SACCHETTO DAI SACCHETTI DA PESARE
APPENDI IL SACCHETTO CHE HAI IN MANO ALLA GRUCCIA
FINCHÉ CI SONO ANCORA SACCHETTI DA PESARE:
    PRENDI IN MANO UN SACCHETTO DAI SACCHETTI DA PESARE
    APPENDI IL SACCHETTO CHE HAI IN MANO ALLA GRUCCIA
    PRENDI IN MANO IL SACCHETTO PIÙ LEGGERO TRA I DUE SULLA GRUCCIA
    METTI IL SACCHETTO CHE HAI IN MANO NEI SACCHETTI ELIMINATI
```

Dall’unplugged al plugged: un simulatore

- Conservano il vincolo concettuale dell’attività fisica
- Permettono prove ripetute e il confronto tra strategie
- Aiutano l’accessibilità e una didattica inclusiva
- Open source, funzionano nel browser
- Modalità "programmatore" con un semplice linguaggio a blocchi

Trova il più pesante

lodi.ml/gruccia

Che cos’è un algoritmo

Algoritmi: «descrizioni precise e non ambigue **(in riferimento a uno specifico esecutore)** di procedure (per raggiungere […] obiettivi) che si prestano ad essere automatizzate»

Indicazioni Nazionali 2025

Problema specifico

60 yogurt alla banana, 45 alla fragola, 38 all’ananas: quanti in tutto?

La soluzione è un numero: 143

Problema generale

Ogni giorno arriva un certo numero di yogurt di ogni gusto: quanti in tutto?

La soluzione è un procedimento: un algoritmo

Gruccia-bilancia: che cosa c’è di informatico

- **Problema ≠ algoritmo**: l’algoritmo è il procedimento che funziona per qualsiasi sacchetto
- **Ricerca del massimo**: scorrere una sequenza e agire se vale una condizione
- **Correttezza**: funziona sempre? Si prova con dati diversi
- **Generalizzazione**: dalla sequenza alla ripetizione

Nelle Indicazioni Nazionali 2025

COMPETENZE (FINE PRIMARIA)

- Descrivere procedure per lo svolgimento di compiti pratici mediante algoritmi
- Scrivere e comprendere semplici programmi, espressi in elementari linguaggi di programmazione a scopo didattico, e valutarne l’adeguatezza rispetto al compito che si vuole automatizzare

CONOSCENZE (PRIMARIA)

- Concetto di algoritmo e sua esecuzione rigorosa
- Controllo di correttezza dei programmi

COMPETENZE (FINE SECONDARIA DI PRIMO GRADO)

- Comprendere il concetto di variabile e scrivere […] programmi che le usano […], valutandone la correttezza

OBIETTIVI (FINE SECONDARIA DI PRIMO GRADO)

- Comprendere e utilizzare il concetto di algoritmo come strumento per la risoluzione di problemi
- Valutare l’esito di un algoritmo o di un programma seguendone i passi e tenendo traccia del valore delle variabili



Programmatori, esecutori e tangram

Attività 3 · Programmazione: un linguaggio condiviso

Tangram: come si gioca

- A coppie, senza vedersi: di spalle o con un divisorio
- Chi programma compone una figura con tutti e 7 i pezzi del tangram
- …e la descrive **solo a parole**: niente gesti, niente sbirciate
- Chi esegue la ricostruisce; poi si confrontano le figure: sono sovrapponibili pezzo per pezzo?



programmatore esecutore

Tre turni di gioco

Turno 1

∞

Domande di chiarimento libere

Turno 2

3

Al massimo 3 domande di chiarimento

Turno 3

0

Nessuna domanda

Ogni turno si gioca due volte, scambiando i ruoli. Al terzo turno chi esegue è come un computer: **non può chiedere chiarimenti**.

Tangram: dopo ogni turno

? Domande stimolo

- Siete riusciti sempre a capirvi? Quando no, da cosa dipendeva?
- (A chi programma) Quali configurazioni sono state più difficili da descrivere?
- (A chi esegue) Quali istruzioni sono state più difficili da capire? Che domande facevate?
- Al prossimo turno, a che cosa farete più attenzione per capirvi meglio?

In classe: dal parlato allo scritto

Fase 2, dalla classe terza · il testo è il programma, il gruppo che lo legge è l'esecutore

1. A consegna il proprio testo e i pezzi a B
2. B esegue alla lettera, «a caccia» di ambiguità
3. A osserva in silenzio, senza intervenire
4. A corregge il testo
5. B ricostruisce di nuovo la figura
6. Confronto con il disegno-soluzione; poi si scambiano i ruoli

Tangram: perché funzioni

! Aspetti a cui fare attenzione

- L’obiettivo non è ricostruire la figura «giusta»: chi scrive cerca di evitare le ambiguità, chi interpreta cerca di trovarne il più possibile
- Si fa **debugging del testo, non delle persone**: si parla di istruzioni da scrivere meglio, mai di «compagni che non sanno spiegare»

Tangram: che cosa c’è di informatico

- **Il linguaggio naturale è ambiguo**: per capirsi serve un linguaggio condiviso
- Meno domande = più vicini a un computer, che non può chiedere spiegazioni
- Prepara ai **linguaggi formali**, con regole e significato precisi
- **Debugging**: un ciclo di prova, osservazione e revisione

Nelle Indicazioni Nazionali 2025

COMPETENZE (FINE PRIMARIA)

- Descrivere procedure per lo svolgimento di compiti pratici mediante algoritmi

OBIETTIVI (FINE TERZA PRIMARIA)

- Descrivere a parole attività della vita quotidiana tramite sequenze di passi precisi e non ambigui e saperle eseguire
- Scrivere semplici programmi e verificare, mediante la loro esecuzione, se svolgono il compito previsto ed eventualmente correggerli

COMPETENZE (FINE SECONDARIA DI PRIMO GRADO)

- Riconoscere le connessioni con altri linguaggi formali in particolare con quello informatico

OBIETTIVI (FINE SECONDARIA DI PRIMO GRADO)

- Riprodurre […] figure del piano e dello spazio […] in base a una descrizione fatta da altri […] e comunicare la costruzione ad altri, in modo che possano riprodurle
- Esaminare un algoritmo o un programma per capirne il comportamento, identificarne eventuali difetti e correggerli (debug)

Esecutore, manuale delle istruzioni, programma

- **Esecutore**: esegue in modo meccanico e prevedibile solo le istruzioni che «comprende»
- **Manuale *delle* istruzioni**: tutte e sole le istruzioni che l'esecutore sa eseguire, con il loro effetto
- **Programma***: una sequenza finita di istruzioni del manuale

MANUALE *DELLE* ISTRUZIONI	
ISTRUZIONE	**AZIONE**
`VAI AVANTI DI UNA CASELLA`	Avanza di una casella nella direzione in cui guarda
`GIRA A SINISTRA`	Ruota su se stesso di 90° verso sinistra
`GIRA A DESTRA`	Ruota su se stesso di 90° verso destra

Programmatore o esecutore?



Esecutore

Esegue alla lettera solo le istruzioni del manuale, senza interpretare né indovinare



Programmatore

Progetta le istruzioni, ne prevede gli effetti, osserva l'esecuzione e corregge gli errori

Molte attività *si fermano all'esecuzione*: l'obiettivo è passare **dall'eseguire al programmare**.

Programmare davvero: Code.org e Scratch

- **Un esecutore automatico** è «stupido e intransigente»: non interpreta e non indovina
- Fa emergere le ambiguità e rende necessarie istruzioni complete, precise e verificabili
- Dà un **feedback immediato**: si prova, si osserva, si corregge
- **Sequenze e ripetizioni** diventano comportamenti che si vedono sullo schermo

Programma il Futuro

SCRATCH

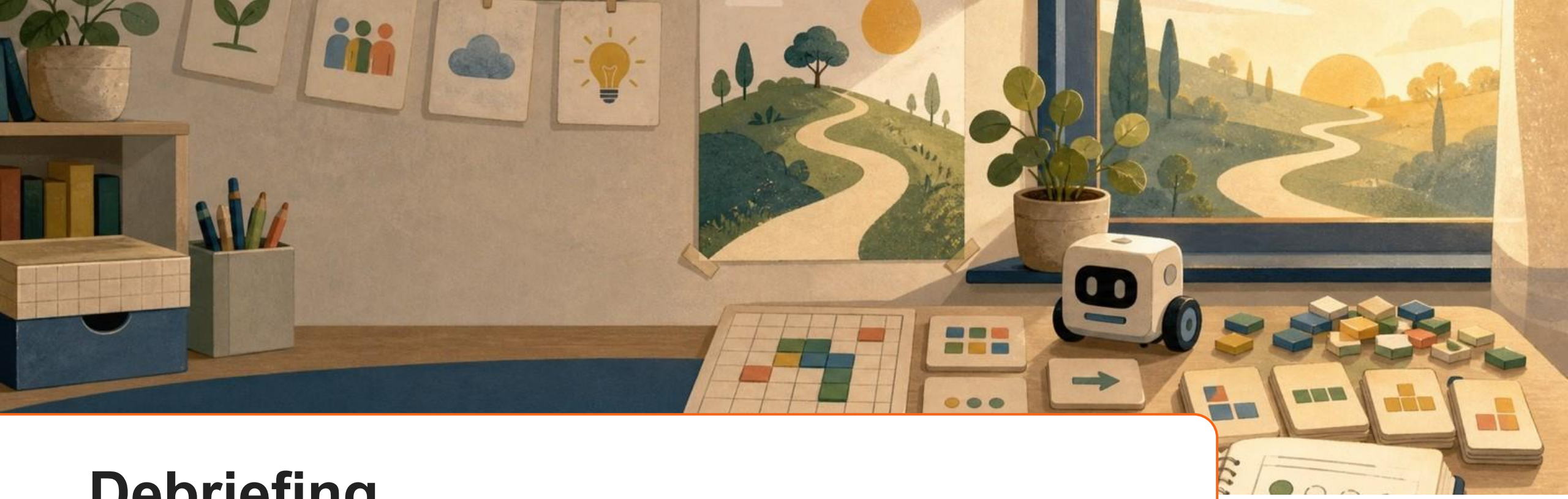
Nella guida

CAPITOLO 7 · CODE.ORG

- Percorsi guidati a blocchi, con sfide progressive e un feedback a ogni livello

CAPITOLO 8 · SCRATCH

- Ambiente aperto e costruzionista: storie, giochi, animazioni
- Si parte facile e si arriva lontano: «pavimento basso, soffitto alto, pareti ampie»



Debriefing

Che cosa abbiamo imparato, e come portarlo in classe

Tre attività, un filo rosso TODO RIVEDERE

ATTIVITÀ	AREA	IDEE INFORMATICHE
Programmatori, esecutori e tangram	Programmazione	Linguaggio condiviso e non ambiguo; programmatore ed esecutore; debugging
Luci di Natale	Informazione e dati	Informazione, dati e rappresentazione; codici univoci e il loro costo
Gruccia-bilancia	Algoritmi	Problema e algoritmo; manuale delle istruzioni; correttezza e generalizzazione

In tutte: un esecutore esplicito, ruoli chiari, l'errore come risorsa per il debugging.

Mettiamo a confronto TODO RIVEDERE

? Domande stimolo

- Che cosa significa descrivere una procedura per un esecutore che dispone solo di certe istruzioni?
- Come si riconoscono e si riducono le ambiguità?
- Come verificare che una strategia funzioni non solo in un caso particolare, ma in generale?
- Che ruolo hanno la prova, l'errore e la revisione?

Una lente sulle pratiche che già facciamo

Rappresentare informazioni, usare linguaggi simbolici, descrivere procedure, risolvere problemi: molte attività, soprattutto in matematica, contengono già principi dell'informatica.



Eseguire una procedura

Applicare passi già noti, come la divisione in colonna



Progettare una procedura

Inventare, descrivere e verificare un algoritmo per un esecutore

La differenza va resa esplicita: è lì che entra l'informatica.

Rischi da evitare

! Aspetti a cui fare attenzione

- «Coding» usato per tutto, senza collegamenti espliciti ai principi dell'informatica
- L'algoritmo ridotto a «ricetta»
- Un esecutore che resta implicito
- Confusione tra problema specifico, problema generale e algoritmo
- Confusione tra informazione, dati e codifica

I principi che ci hanno guidato

1. **Informatica ≠ competenza digitale** ≠ uso delle tecnologie
2. **Esecutori espliciti**, ciascuno con il suo manuale delle istruzioni
3. **Dall'eseguire al programmare**: prevedere, osservare, correggere
4. **Più aree della disciplina**, non solo la programmazione
5. **Un ponte tra unplugged e plugged**: Code.org, Scratch, simulatori

Ora tocca a voi

Scegliete un’attività: quali vincoli e quale livello di formalizzazione per la vostra classe?

Per semplificare

- Meno elementi da gestire: pezzi, messaggi, sacchetti
- Testi già scritti tra cui scegliere, invece che da scrivere
- Domande guida più esplicite

Per rendere più sfidante

- Configurazioni tridimensionali al posto del tangram
- Inventare un codice, o correggere esempi con errori
- Più sacchetti, pesi uguali, contare le pesate

La guida aperta: informaticainclasse.it



Per docenti del primo ciclo

- Pensata per la primaria, adatta a chiunque si avvicini per la prima volta all'informatica

Una guida aperta: costruiamola insieme

1. **Provatela in classe**
   e raccontateci com’è andata

2. **Segnalate errori**
   e proponete varianti

3. **Documentate le sperimentazioni**
   per un ciclo di ricerca-azione

4. **La formazione è il nodo**
   servono altre guide, esempi commentati, comunità di pratica

Sulle orme di Milù

Mondadori Education, 2026

- **Informatica**: un fascicolo di 48 pagine per il primo triennio della primaria
- **Guida di Informatica** per l'insegnante: 120 pagine
- Materiali coerenti con le Indicazioni 2025, pensati anche per chi insegna senza una formazione informatica



Fascicolo per la classe



Guida per l'insegnante

Ne volete ancora? Il Teachers' Day di CMSC 2026

Giovedì 15 ottobre, a Padova

Dipartimento di Ingegneria dell'Informazione, Aula Magna «A. Lepschy»

Partecipare

- Gratuito per i docenti di primaria e secondaria di primo grado
- Ancora pochi posti: max 50, in ordine di iscrizione

Il programma

- Un nostro laboratorio, con altre attività della guida
- Metà degli interventi in italiano, metà in inglese

Programma e iscrizioni: cmsc2026.github.io/teachers-day

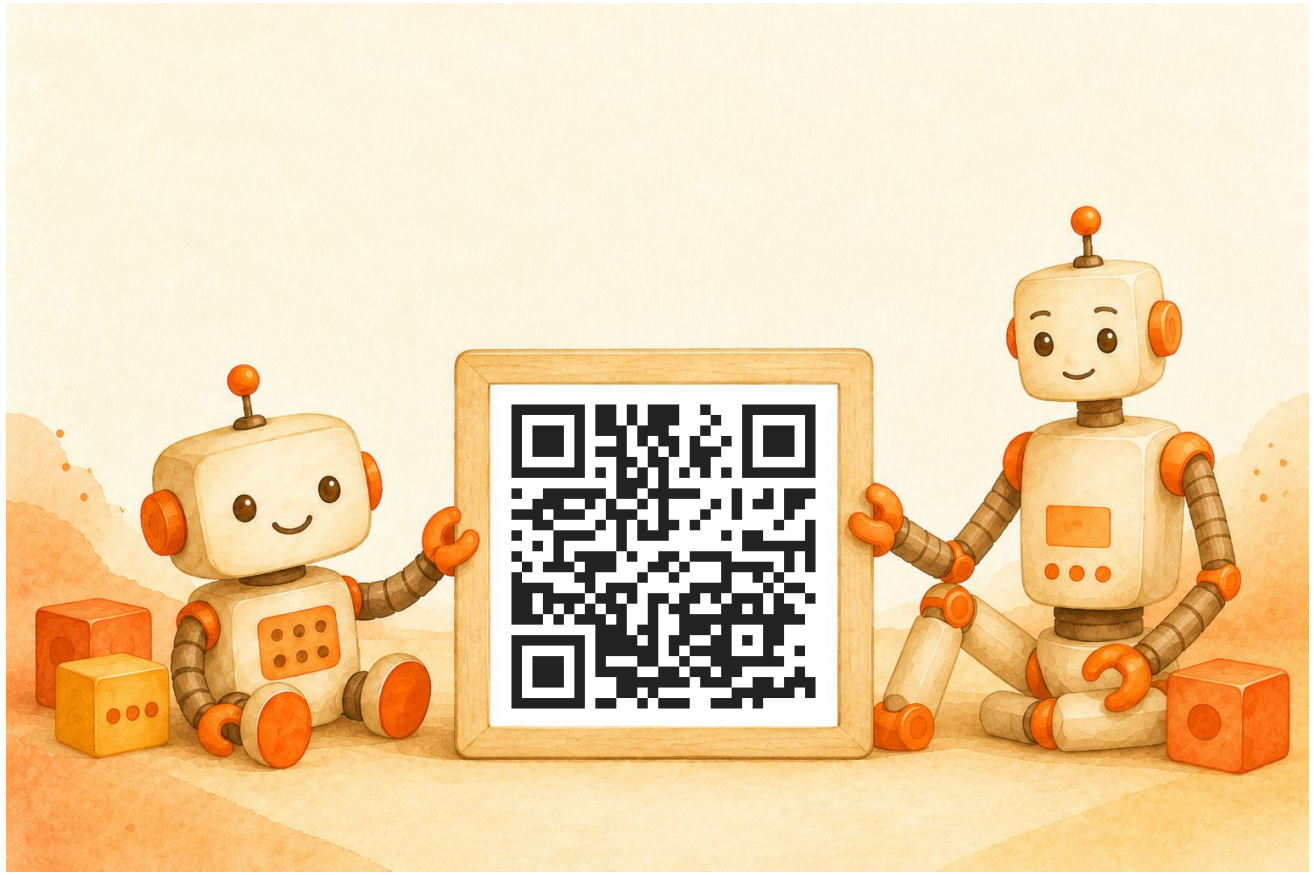
Grazie!

Insegnare Informatica

Una guida per docenti del primo ciclo

informaticainclasse.it

ITADINFO 2026 · Padova, 10 ottobre



[ITADINFO] ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA UNIVERSITÀ DI TRENTO Verona FabLab

Illustrazioni della guida

Slide nascosta · copia un’illustrazione e usala con il layout «Sezione illustrata»



Indice



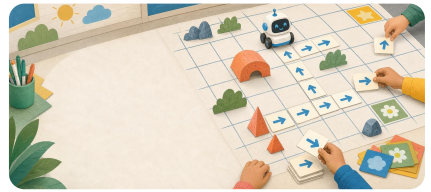
1 · Introduzione



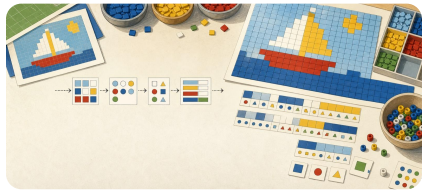
2 · Concetti



3 · Algoritmi quotidiani



4 · Movimenti su griglia



5 · Rappresentazione



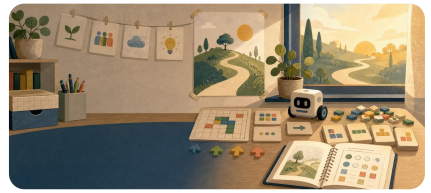
6 · Strategie



7 · Code.org



8 · Scratch



9 · Conclusioni

Materiali da stampare

Luci di Natale: sei messaggi, cinque codici… e il vostro

Alice e Bob accendono sulle loro case una lettera alla volta. Ecco i sei messaggi scritti con cinque codici diversi: nell'ultima colonna inventate il vostro.

	Codice 1 tutte le lettere del messaggio	**Codice 2** una parola a caso	**Codice 3** l'iniziale	**Codice 4** un inizio prestabilito	**Codice 5** una sigla concordata	**Codice 6** inventa tu un codice più breve ma univoco
1	CI VEDIAMO OGGI POMERIGGIO	*una parola a caso scelta all'interno del messaggio*	C	CI VEDIAMO O	CP	
2	NON POSSO VENIRE FUORI PERCHÉ NON MI SENTO BENE RESTO A CASA		N	NON POSSO VENIRE FUORI PERCHÉ	NB	
3	PORTA LA PALLA		P	P	PP	
4	ANDIAMO A MANGIARE FUORI		A	A	AM	
5	NON POSSO VENIRE FUORI PERÒ VIENI A CASA MIA		N	NON POSSO VENIRE FUORI PERÒ	NP	
6	CI VEDIAMO DOMANI		C	CI VEDIAMO D	CD	

- Quali codici permettono ad Alice e Bob di capirsi sempre, senza mai sbagliarsi? Da cosa lo avete capito?
- Tra questi, quale permette di accendere meno lettere possibile? Come fate a dirlo?
- Riuscite a inventare un nuovo codice ancora più breve, ma in modo che Alice e Bob possano sempre capirsi?

Attività tratta da **Insegnare Informatica. Una guida per docenti del primo ciclo**
di Michael Lodi, Agnese Del Zozzo, Alberto Montresor, Giorgia Bissoli

CC BY NC SA

ELIMINATI

CC BY NC SA

DA PESARE

Attività tratta da **Insegnare Informatica. Una guida per docenti del primo ciclo**
di Michael Lodi, Agnese Del Zozzo, Alberto Montresor, Giorgia Bissoli

CC BY NC SA

Gruccia-bilancia: trovare il più pesante

MANUALE DELLE ISTRUZIONI (GRUCCIA-BILANCIA)
METTI IL SACCHETTO CHE HAI IN MANO NEI SACCHETTI ELIMINATI
APPENDI IL SACCHETTO CHE HAI IN MANO ALLA GRUCCIA
PRENDI IN MANO **UN SACCHETTO** DAI SACCHETTI DA PESARE
PRENDI IN MANO IL SACCHETTO **PIÙ LEGGERO** TRA I DUE SULLA GRUCCIA

REGOLA

In ogni momento puoi avere in mano **un solo sacchetto**.

OBIETTIVO

Alla fine, sulla gruccia deve restare **solo il sacchetto più pesante**.

	Algoritmo per trovare il più pesante tra 4 sacchetti
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

Attività tratta da **Insegnare Informatica. Una guida per docenti del primo ciclo**
di Michael Lodi, Agnese Del Zozzo, Alberto Montresor, Giorgia Bissoli

CC BY NC SA