



Istituto Statale d'Istruzione Superiore  
"Arturo Malignani" - Udine



Istituto Statale di Istruzione Superiore "A. Malignani" - Udine



# Errore e Apprendimento

IND  
IRE

ISTITUTO  
NAZIONALE  
DOCUMENTAZIONE  
INNOVAZIONE  
RICERCA EDUCATIVA



14-15 novembre 2019  
ISIS "A. Malignani" - Udine  
*Santino Bandiziol*

© Errore e Apprendimento 0v05 - 2019  
Santino Bandiziol

*Le informazioni contenute nelle presenti pagine sono state verificate e documentate con la massima cura possibile. Nessuna responsabilità derivante dal loro utilizzo potrà venire imputata all'Autore o alle società coinvolte nella loro creazione, pubblicazione e distribuzione.*

Alcuni diritti riservati.

Documento prodotto con L<sup>A</sup>T<sub>E</sub>X.  
L'immagine di copertina è di proprietà di  
Katieb50  
Titolo originale dell'opera: "Looking for gators"  
rilasciata con licenza CC BY

Le restanti immagini sono di pubblico dominio o elaborate dall'autore.

Questo documento è rilasciato con licenza



**Creative Commons BY-NC-SA**

Attribuzione – Non Commerciale – Stessa licenza

<http://creativecommons.org/licenses/by-nc-sa/3.0/it/deed.it>

**Attribuzione** — Devi riconoscere una menzione di paternità adeguata, fornire un link alla licenza e indicare se sono state effettuate delle modifiche. Puoi fare ciò in qualsiasi maniera ragionevole possibile, ma non con modalità tali da suggerire che il licenziante avalli te o il tuo utilizzo del materiale.

**Non commerciale** — Non puoi usare il materiale per scopi commerciali.

**Stessa licenza** — Se remixi, trasformi il materiale o ti basi su di esso, devi distribuire i tuoi contributi con la stessa licenza del materiale originario.

# Indice

<b>Indice</b>	<b>i</b>
<b>Introduzione</b>	<b>v</b>
<b>I LA DIDATTICA</b>	<b>1</b>
<b>1 L'errore</b>	<b>3</b>
1.1 Un errore per amico . . . . .	4
1.1.1 Favorire l'accettazione dell'errore . . . . .	5
1.1.2 Due mentalità . . . . .	6
1.2 Un esempio concreto . . . . .	7
1.2.1 La trascrizione . . . . .	8
1.2.2 Le note . . . . .	11
1.2.3 Una riflessione importante . . . . .	11
1.3 Un secondo esempio . . . . .	12
1.3.1 La trascrizione . . . . .	13
1.3.2 Le note . . . . .	14
1.3.3 Un'altra riflessione . . . . .	15
1.4 Conclusioni . . . . .	15
<b>2 L'apprendimento</b>	<b>17</b>
2.1 Un carico particolare . . . . .	18
2.1.1 La soluzione . . . . .	19
2.1.2 Com'è andata? . . . . .	21
2.2 Un problema da terza elementare . . . . .	22
2.2.1 La somma delle cifre . . . . .	22
2.2.2 La soluzione . . . . .	23
2.2.3 Commento alla soluzione . . . . .	25
2.3 Il lavoro di gruppo . . . . .	25
2.4 Il ruolo del docente . . . . .	27
2.5 La Robotica e le materie curriculari . . . . .	28
<b>II GLI STRUMENTI</b>	<b>31</b>
<b>3 Mindstorm</b>	<b>33</b>
3.1 I Mindstorms . . . . .	34
3.1.1 Tracker . . . . .	34

3.1.2	Reptar . . . . .	35
3.1.3	Race Truck . . . . .	35
3.1.4	Dinorex . . . . .	36
3.1.5	Everstorm . . . . .	36
3.2	L'elettronica di Mindstorm . . . . .	36
3.3	La programmazione di Mindstorm . . . . .	38
3.3.1	Le tavolozze di programmazione . . . . .	38
3.3.2	L'area di programmazione . . . . .	39
3.3.3	Enchanting: un linguaggio alternativo . . . . .	39
3.4	Riassumendo . . . . .	40
<b>4</b>	<b>Arduino</b> . . . . .	<b>41</b>
4.1	Le schede . . . . .	41
4.1.1	Arduino Uno . . . . .	42
4.1.2	Arduino Lilypad . . . . .	42
4.1.3	Arduino Zero Pro . . . . .	43
4.1.4	Gli Shield Arduino . . . . .	43
4.2	L'ambiente di sviluppo . . . . .	44
4.2.1	ArduBlock . . . . .	45
4.2.2	C/C++ . . . . .	46
4.3	Taglia e Incolla . . . . .	47
4.4	Riassumendo . . . . .	48
<b>III</b>	<b>LA DOCUMENTAZIONE</b> . . . . .	<b>49</b>
<b>5</b>	<b>Mindstorm e il Web</b> . . . . .	<b>51</b>
5.1	Hardware . . . . .	51
5.2	Tutorial . . . . .	52
5.2.1	Costruisci un robot . . . . .	52
5.2.2	Video . . . . .	53
5.3	Librerie . . . . .	53
5.4	Miscellanea . . . . .	53
<b>6</b>	<b>Arduino e il Web</b> . . . . .	<b>55</b>
6.1	Hardware . . . . .	55
6.2	Tutorial . . . . .	56
6.2.1	Tutorial italiani . . . . .	56
6.2.2	Tutorial in lingua inglese . . . . .	57
6.2.3	Video italiani . . . . .	57
6.2.4	Video in lingua inglese . . . . .	57
6.3	Librerie . . . . .	58
6.4	Miscellanea . . . . .	58
6.4.1	<i>Language reference</i> . . . . .	59
6.4.2	<i>Arduino Store</i> . . . . .	59
6.4.3	Software e IDE . . . . .	59
6.4.4	<i>Arduino Educational</i> . . . . .	59
6.4.5	Il Forum . . . . .	60
6.5	Riassumendo . . . . .	60

*INDICE*

iii

**Bibliografia**

63



# Introduzione

*"I don't know what's the matter with people:  
they don't learn by understanding;  
they learn by some other way  
- by rote, or something.  
Their knowledge is so fragile!"*

Surely you're joking, Mr. Feynman - Richard Feynman

Il seminario di Avanguardie Educative del 14 e 15 novembre a Udine è organizzato su ritmi molti alti per i docenti che seguono i diversi corsi. Ciò si traduce in una maratona che può risultare molto faticosa e rischia di lasciare ben poco alla fine dei due giorni.

Si è pensato, quindi, di fornire un breve compendio agli insegnanti in modo che possa essere possibile rileggere con calma gli argomenti trattati durante il corso di ICT Lab. Esso è strutturato in tre parti: una prima parte teorica riguardante la didattica; una seconda parte che illustra gli strumenti utilizzabili e una terza parte che fornisce un'ampia documentazione per chi volesse approfondire le proprie conoscenze sugli strumenti proposti durante il corso.

Alla fine del documento è indicata la bibliografia dei testi citati e dei quali si consiglia la lettura.

Si ritiene, infine, doveroso scusarsi con il lettore per la gran fretta con la quale i presenti appunti sono stati scritti. Purtroppo non è stato possibile preparare con maggior calma il presente documento che, pertanto, potrebbe contenere errori e/o imprecisioni. Possono essere notificati al seguente indirizzo e-mail:

santino.bandiziol@gmail.com

Udine, 10/11/2019

*Santino Bandiziol*



**Parte I**

**LA DIDATTICA**



# Capitolo 1

## L'errore

---

*Non hai mai commesso un errore  
se non hai mai tentato qualcosa di nuovo.*  
Albert Einstein

Quando aveva pochi mesi, Carletto era un bel bambino paffutello e sempre sorridente. Mamma e papà lo coprivano di attenzioni e di stimoli e lui guardava e ascoltava. Guardava ammirato la mamma che stava dritta sulle gambe e non era costretta a spostarsi a quattro zampe. Ascoltava il papà che emetteva quei suoni strani e talvolta ripetitivi. Carletto guardava e ascoltava. Mai una volta che la mamma gli spiegasse come stare dritti sulle gambe. Mai una volta che gli dicesse: "Sposta il baricentro leggermeente sul piede d'appoggio e alza l'altro". Mai una volta che il papà gli spiegasse i rudimenti della grammatica italiana. Mai una volta che gli dicesse: "Prima il soggetto, poi il predicato...". Niente di tutto questo.

Ma Carletto una sua tecnica la elaborò ugualmente, quella che gli inglesi chiamano *trial and error*, che traducendo significa grossolanamente "fai dei tentativi e impara dagli errori". E dopo aver tanto guardato e ascoltato, Carletto incominciò a fare le sue prime prove. Ogni tanto - anzi, spesso - cadeva e diceva qualche strafalcione, ma ogni volta imparava qualcosa di nuovo e migliorava. Questa "rudimentale" tecnica permise a Carletto, in pochi anni, di diventare padrone della lingua italiana e di camminare, correre e saltare in maniera tanto spericolata da far venire il batticuore, talvolta, alla mamma.

All'età di quattro anni, mamma e papà lo iscrissero ai "primi calci" della scuola di calcio locale. A sei anni era già iscritto nei "pulcini" e partecipava a tutti gli allenamenti, imparando i fondamentali del nobile sport. Ma come? Sempre con la stessa tecnica: facendo centinaia di colpi di testa, di stop di petto, di lanci e di passaggi. Insomma, provando e riprovando, finché gli errori si

ridussero sia in quantità che in qualità. L'errore era fedelmente al suo fianco ad ogni allenamento e con pazienza e fermezza gli diceva che aveva sbagliato lo stop o clamorosamente mancato la palla. In tal modo l'errore era diventato suo amico inseparabile.

Poi, un giorno, quell'amicizia si incrinò. Successe a scuola, all'inizio della prima elementare, quando la maestra, con faccia per nulla sorridente, disse a Carletto che aveva sbagliato. "Che c'è di male?" pensò Carletto, "ho imparato una cosa nuova, adesso". Ma la maestra aveva una faccia strana, come se Carletto avesse fatto qualcosa che non avrebbe dovuto fare. Col tempo le cose peggiorarono. Carletto continuava a fare errori - ma li aveva sempre fatti, non c'era mica nulla di nuovo! - e la maestra col tempo mostrava sempre meno pazienza. Gli aveva anche dato qualche giudizio negativo e l'errore non sembrava più quell'amico fedele di qualche tempo fa.

Anche mamma e papà non sembravano tanto contenti degli errori che Carletto faceva a scuola, tanto che incominciò a pensare che aveva qualcosa che non andava. "Forse faccio troppi errori", pensava, "forse non sono bravo come gli altri bambini". Anche il *mister* non gli sembrava più lo stesso. Quando gli diceva di tenere il corpo più avanti, non sembrava che giudicasse il gesto tecnico, ma gli pareva proprio che il giudizio fosse rivolto a lui, a Carletto.

Oggi Carlo è adolescente e frequenta le scuole superiori. A scuola va male e lui è convinto di essere stupido. A scuola ha capito che bisogna studiare per il voto, non per apprendere. Quando cerca di capire le cose, gli altri gli mettono fretta e lo giudicano. In famiglia sono urla ogni volta che porta a casa un votaccio e gli insegnanti gli parlano solo di voti, di bocciature, di doveri, di colpe, di errori, errori, errori. Insomma un disastro. Ora Carlo odia l'errore e chi lo ha inventato. Ovvio: come si fa ad avere un errore per amico?

## 1.1 Un errore per amico

Naturalmente la storiellina di Carletto è inventata, ma è verosimile e rispecchia, nella sostanza, la situazione di molti studenti. Finché noi docenti non saremo capaci di porre le basi per un apprendimento genuino e profondo, sarà molto difficile ottenere risultati degni di nota. Ciò che si otterrà sarà che Carletto svilupperà delle strategie per portare a casa dei voti positivi e non delle strategie di apprendimento.

Carlo ha capito già alle elementari che ciò che conta è il voto e non il reale sapere maturato, ovvero le reali competenze maturate. Se il nostro protagonista dirà a casa che ha preso un votaccio impronunciabile in matematica, lo aspetteranno litigi e rimproveri. Se, invece, Carlo racconterà alla famiglia che ha preso una sufficienza aiutandosi, però, con il telefonino, verrà probabilmente rimproverato per la condotta non proprio irreprensibile dal punto di vista etico, ma le urla genitoriali saranno molto probabilmente mitigate dalla benevola valutazione riportata. Inoltre, Carlo è perfettamente autorizzato, dal suo punto di vista, di affinare ulteriormente la strategia scolastica evitando di raccontare proprio tutto in famiglia, sorvolando, ad esempio, lo spiacevole dettaglio del telefonino.

Insomma, quello che Carlo avrà elaborato durante la sua carriera scolastica sarà una raffinata strategia per evitare gli insuccessi.<sup>1</sup> Probabilmente, però, non avrà appreso molto. Dunque che fare?

Credo sia fondamentale ripristinare un corretto rapporto con l'errore, che può essere solamente un rapporto di "amicizia". L'errore è un fondamentale compagno di studi e un fedele alleato durante l'apprendimento. Caricare l'errore di altri significati è dannoso per lo studente e un boomerang nel medio-lungo termine per il docente.

Deve essere data la possibilità allo studente di riflettere sull'errore e di reconsiderarne la natura benigna. Si pensi al musicista, ad esempio. Certi passaggi vengono provati e riprovati centinaia forse addirittura migliaia di volte prima di essere accettati qualitativamente come validi. Ad ogni esecuzione viene valutata la carica espressiva del passaggio, la sua totale coerenza con lo spartito, la fluidità e la ritmicità del susseguirsi delle note e degli accordi. Tutto ciò senza tuttavia mortificarne la personale interpretazione. Anche l'atleta percorre la stessa faticosa strada. Si pensi al judoka. Una tecnica viene "tirata" decine di migliaia di volte prima di essere padroneggiata con maestria. E anche dopo, ad alti livelli, viene studiata per centinaia di volte ad ogni sessione di allenamento. L'errore è sempre lì, fedele servitore e imparziale giudice dell'esecuzione del musicista o dell'atleta.

Quest'amicizia fra esecutore ed errore stimola una mentalità di crescita,<sup>2</sup> che va al di là del voto e della valutazione scolastica. Uno studente così formato sarà maggiormente aperto alla critica e più orientato all'apprendimento genuino.

### 1.1.1 Favorire l'accettazione dell'errore

L'idea ICT Lab di Avanguardie Educative contiene in sé tutte le premesse per favorire una serena accettazione dell'errore. Per meglio comprendere il perché di tale affermazione è bene rileggere il pensiero di Seymour Papert a tal proposito:

*«Tipicamente, durante una lezione di matematica, la reazione del bambino di fronte a un errore è di tentare di dimenticarlo più in fretta possibile. Ma nell'ambiente di sviluppo LOGO, il bambino non è criticato a causa di un errore nel tracciato (della tartaruga. NdA). L'azione di debugging è una componente normale del processo di comprensione del programma. Il programmatore è incoraggiato a studiare il malfunzionamento piuttosto che dimenticare l'errore commesso.»<sup>3</sup>*

La citazione va spiegata. Negli anni '70 e '80 Seymour Papert insieme al suo team di ricerca, sviluppò al MIT di Boston un linguaggio di programmazione indirizzato ai bambini denominato LOGO. Permetteva di movimentare un

<sup>1</sup>L'espressione tecnica è "obiettivo di evitamento", che è la traduzione un po' cacofonica di "avoidance goal". L'argomento è trattato molto bene in ELLIOT, HULLEMAN, *Achievement Goals* in ELLIOT, DWECK, YEAGER, *Handbook of Competence and Motivation* [10], pagg.43-60 e in BOSCOLO, *La fatica e il piacere di imparare* [5], pagg. 29-56.

<sup>2</sup>*Growth Mindset* in inglese. La teorica del *Mindset* è Carol Dweck. Si veda a tal proposito DWECK, *Mindset* [8].

<sup>3</sup>PAPERT, *Mindstorms* [18], pag.61. Traduzione dell'autore.

cursore triangolare - la tartaruga - sullo schermo di un PC mediante istruzioni molto semplici: avanti, indietro, ruota, ecc. Detto linguaggio è stato usato per fini didattici per molti anni finché è stato superato dai linguaggi iconici ancora più immediati e spendibili nella scuola primaria, uno fra tutti *scratch* (sviluppato anch'esso al MIT). L'approccio computazionale e il comportamentismo hanno aperto un'era e permesso la nascita di una branca della didattica - il pensiero computazionale, appunto - che mantiene intatta la sua efficacia anche ai tempi odierni.

Quindi la robotica educativa e il pensiero computazionale in generale possono essere ottimi strumenti per veicolare una serena accettazione dell'errore da parte dello studente. L'azione, però, non può e non deve fermarsi a tale livello. È compito del docente far prendere consapevolezza agli allievi di come venga trattato l'errore durante le esercitazioni di programmazione e di robotica educativa ed estendere tale metodo progressivamente anche ad altre materie.

Si tratta, naturalmente, di un processo lento. Tanto più lento quanto più tormentata è stata l'esperienza scolastica pregressa dello studente e quanto più l'ambiente in cui lo studente è cresciuto lo ha orientato ad una mentalità di crescita o a una mentalità fissa. Vale la pena approfondire un po' il concetto.

### 1.1.2 Due mentalità

Si è accennato nella sezione 1.1 alla teoria del *Mindset*, elaborata negli anni '80 da una giovane ricercatrice, Carol Dweck. È bene premettere che in quegli anni sono state elaborate molteplici teorie in campo psicologico, molte delle quali strettamente collegate fra loro, alcune delle quali hanno incontrato particolare favore di pubblico assurgendo agli onori della cronaca e scalando le classifiche dei *bestseller*: l'Autoefficacia di Bandura,<sup>4</sup> la Self-Determination Theory di Deci e Ryan,<sup>5</sup> le Intelligenze Multiple di Gardner,<sup>6</sup> l'Intelligenza Emotiva di Goleman,<sup>7</sup> la già citata teoria degli *achievement goals* di Elliot e Hulleman,<sup>8</sup> la Grinta della Duckworth,<sup>9</sup> la Mindfulness di Kabat-Zinn,<sup>10</sup> ecc.

La teoria del *Mindset* classifica lo studente secondo due opposti tipi di mentalità: la *Growth Mindset* (mentalità di crescita) e la *Fixed Mindset* (mentalità fissa). Il primo tipo di mentalità - la mentalità di crescita - presuppone che ciascuno di noi possa crescere e migliorarsi, per cui un eventuale insuccesso è legato ad una non sufficientemente efficace preparazione dell'azione fallita. Il secondo tipo di mentalità - la mentalità fissa - presuppone che ciascuno di noi sia più o meno dotato da madre natura e che tali doti siano sostanzialmente immutabili, per cui un eventuale insuccesso è inscindibilmente legato alla persona.

Se ne deduce che lo studente con mentalità di crescita accoglie volentieri le critiche, le valuta come critiche al proprio operato e le utilizza per miglio-

<sup>4</sup>BANDURA, *Autoefficacia. Teoria e applicazioni* [2].

<sup>5</sup>DECI, RYAN, *Self-Determination Theory* [6].

<sup>6</sup>GARDNER, *Formae Mentis. Saggio sulla pluralità dell'intelligenza* [11].

<sup>7</sup>GOLEMAN, *Intelligenza Emotiva* [12].

<sup>8</sup>ELLIOT, HULLEMAN, *Achievement Goals* in ELLIOT, DWECK, YEAGER, *Handbook of Competence and Motivation* [10], pagg.43-60.

<sup>9</sup>DUCKWORTH, *Grinta. Il potere della passione e della perseveranza* [7].

<sup>10</sup>KABAT-ZINN, *Full Catastrophe Living* [15].

rarsi. Lo studente con mentalità fissa, invece, non accetta volentieri le critiche, in quanto critiche alla propria persona immutabile e non all'altezza della situazione. Né ora né mai.

La robotica educativa può essere un valido strumento per far crescere gli studenti di mentalità fissa. Quando tale studente sbaglia, ha la naturale tendenza a ricercare in autonomia l'errore che ha provocato il malfunzionamento del programma e ciò non viene percepito come critica. Il ruolo del docente consiste nello sfruttare tale situazione e far prendere consapevolezza allo studente sulla propria capacità di utilizzare l'errore per correggersi e migliorare. Tale riflessione da parte dello studente è fondamentale e solitamente deve essere indotta e stimolata dall'insegnante. Naturalmente ciò non prefigura alcun miracolo, ma permette allo studente di credere un po' di più nelle proprie capacità di crescita e incrina la convinzione di non essere all'altezza della situazione.

## 1.2 Un esempio concreto

Quasi ogni anno, da qualche lustro a questa parte, do ai miei allievi del terzo corso il seguente problema.<sup>11</sup> Fornisco loro un piccolo robot dotato di due ruote motrici e una libreria di funzioni per permettere loro di movimentarlo senza troppa fatica. La libreria contiene delle funzioni che permettono al robot di avanzare, di indietreggiare, di svoltare a destra a sinistra, ecc. Nel ristretto arco di tempo di 30-40 minuti, gli allievi sono in grado di movimentare più o meno a piacere il robot, facendogli compiere semplici traiettorie sul pavimento.

Quando i ragazzi hanno maturato un minimo di maestria nel controllo del robot, chiedo loro di movimentarlo in modo tale che tracci sul pavimento un quadrato di un certo numero di *step* di lato. La cosa riesce solitamente piuttosto facile agli studenti che, oltretutto, si divertono.

Al termine della suddetta esercitazione aumento la difficoltà delle consegne e chiedo agli studenti di tracciare sul pavimento un triangolo rettangolo avente angoli interni, ad esempio, di  $30^\circ$ ,  $60^\circ$  e  $90^\circ$  e lati a piacere (ma coerenti con gli angoli dati). Gli studenti si buttano di solito a capofitto nel problema e succede qualcosa di inatteso.

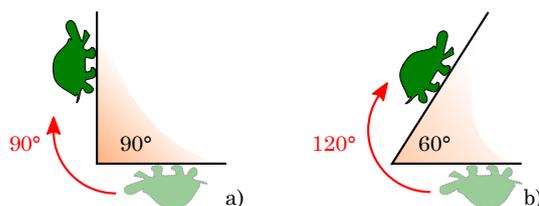


Figura 1.1: Angoli interni ed esterni

Quando il robot - rappresentato come una tartaruga nel disegno, in onore alla *turtle* di Papert - ha terminato di tracciare la base (ad esempio il cateto maggiore), deve ruotare di un certo angolo per percorrere l'ipotenusa. Generalmente, gli studenti impongono che la rotazione del robot sia pari all'angolo

<sup>11</sup>L'esercitazione prende spunto dal bel libro di Abelson e Disessa "La geometria della tartaruga" [1]. Purtroppo tale libro si trova ormai solamente in lingua inglese.

interno del triangolo dato. Se tale angolo è di  $90^\circ$ , l'angolo interno e l'angolo esterno coincidono e non vengono rilevati errori, ma se l'angolo interno è, ad esempio, di  $60^\circ$ , ruotare il robot di tale angolo è un errore clamoroso. La effettiva rotazione del robot deve essere di  $180-60 = 120^\circ$  e non  $60^\circ$ .

Quello che gli studenti vedono tracciare sul pavimento non è un triangolo rettangolo ma una poligonale aperta simile a quella di illustrata nella figura sottostante.

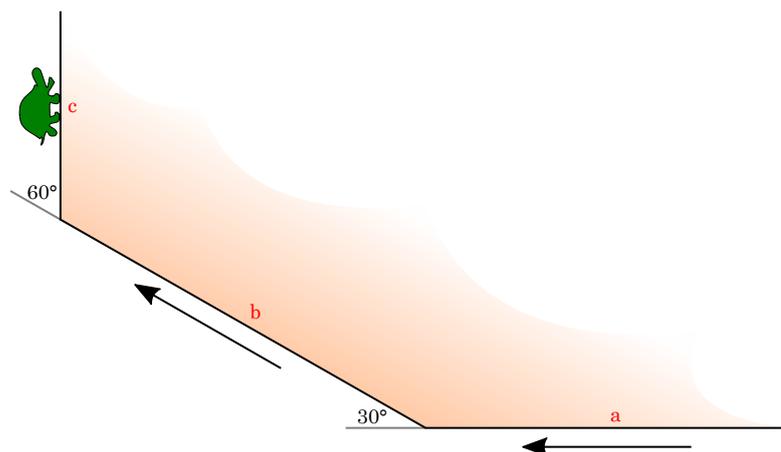


Figura 1.2: Percorso effettivo

Un simile risultato li coglie solitamente di sorpresa, ma invariabilmente, da anni, gli studenti si comportano sempre allo stesso modo: non chiedono aiuto all'insegnante, ma affrontano la nuova e inaspettata difficoltà in autonomia.

Qualche anno fa, con il permesso degli studenti, ho posto un registratore acceso sul banco di un gruppo di studenti durante l'esercitazione e ho analizzato poi con calma l'iter risolutivo messo in pratica dai ragazzi. La documentazione che segue è stata fatta seguendo le indicazioni fornite da Mara Kreshevsky in "Visible Learners".<sup>12</sup>

Di seguito viene fornita la trascrizione della registrazione audio relativa allo scambio di opinioni avvenuto all'interno di uno dei gruppi che hanno partecipato all'esercitazione.

Per facilitare l'individuazione delle frasi alle quali sono riferiti i successivi commenti, sono state aggiunte delle note colorate in rosso. Inoltre, sono state usate le seguenti abbreviazioni: P = Professore, A1 = Allievo 1, A2 = Allievo 2, A3 = Allievo 3, A4 = Allievo 4.

### 1.2.1 La trascrizione

Tempo: 12' 54" (dall'inizio della lezione)

P: Allora, posso darvi la seconda esercitazione?

<sup>12</sup>KRECHEVSKY, MARDELL, RIVARD, WILSON, *Visible Learners* [17]. Il *framework* di riferimento relativo alla documentazione è MLTV, una delle idee di Avanguardie Educative. L'argomento è trattato in altro documento.

A: Sì. Certo.

(L'insegnante visualizza mediante proiettore il testo della seconda esercitazione. Seguono commenti, spostamenti di sedia e preparativi.)

Tempo: 13' 50"

A1: Dobbiamo fare un triangolo... triangolino. Il lato lungo sono (sic) 100. C'è l'altezza che sono (sic) ... Ah, no ... l'ipotenusa è 87 ...

A2: Aspetta, aspetta. Cos'è 100?

A1: Allora, 100 è la parte lunga.

A3: La base.

A1: Sì, la base.

A2: E l'altezza?

A1: L'altezza è 50 e l'ipotenusa 87.

A3: No, l'ipotenusa 100.

A4: L'ipotenusa è 100.

A1: È vero, scusate.

A3: Come fa la base a essere più lunga dell'ipotenusa? (Nota 1)

A2: 100, 100, 50?

A1: Sì. No, 100, 87, 50.

A3: 100, 87, 50.

A4: Fai il disegno. (Nota 2)

A1: Fai un triangolo rettangolo...

A3: Sì, così ...

A4: 100 lì ... 80 lì ...

A1: Parte da 90... Sai com'è: la somma degli angoli interni fa 180. (Nota 3)

A3: Quanto fa?

A1: 180.

A2: E facciamo: Avanti 87. (I ragazzi stanno scrivendo il programma in linguaggio C) (Nota 4)

A3: Lo facciamo partire da ...

A2: Ruota 30. (Nota 5)

A2: Avanti 100 ... Ruota 60 (Nota 6)

A3: Aggiungi ...

A2: Avanti ...

A3: Occhio! Fai copia-incolla dei due ...

A2: Avanti ... 50 ... Ruota 90.

A3: Ruota 90.

A1: Perché hai messo ... ?

A2: Ecco, qua possiamo mettere lo Stop, tranquillamente. Mettiamo?

A3: Metti Stop.

A4: No, perché per provarlo è meglio.

A3: No, in realtà non serve lo Stop, perché ... se per caso ...

A1: Avanti ... Ruota ... Avanti ... Ruota ... Avanti ... Sì, ok. Metto su...

A2: Vai.

A1: Aspetta che collego ...

(Gli studenti collegano il cavo USB e caricano il programma)

Tempo: 16' 26"

A2: Staccalo. Proviamo a metterlo giù per terra anche adesso? (Nota 7)

(I ragazzi provano il programma, ma non funziona correttamente.)

Tempo: 17' 02''

A2: Sono gli angoli sbagliati ... No ... Si fa ... Il triangolo ... Si fa il triangolo così e dopo riinizia il giro ...

A1: Esatto.

A2: E quindi si torna ...

A3: C'è un Ruota che è sbagliato. (Nota 8)

A2: C'è un Ruota sbagliato.

A4: È un meno 30 gradi ... cioè, meno ... (Nota 9)

A3: Ahhhh ...

A4: Capito?

A1: Ah, già ...

A4: Perché deve girare nel verso opposto ...

A2: È meno 30?

A4: Eh, non so se è meno 30 o meno 90 o meno ... cioè ... Scegliamo un punto di partenza

A1: Il 90.

A3: Il 90. Parte da lì.

A4: Ok, va avanti, quindi ...

A3: Fa 87 ...

A4: Fa 87.

A3: Poi deve girare di ... non di 30

A4: Eh, deve girare di 180 ...

A3: Deve girare di quest'angolo qua

A2: Sì ... sì ...

A3: Cioè di 180 meno ...

A4: 180 meno 30. (Nota 10)

A2: No, no ... aspetta ... ah, sìsìsìsì

(Il programma viene corretto ponendo angoli negativi)

A2: Cos'era, 87? Dopo, Avanti 100 ... Qua è 100 ...

A1: Sì ...

A2: In teoria ...

A3: Dobbiamo anche fare ... tipo ... 30, 60 ...

A2: 90.

A3: 270.

A3: Carica.

A2: Vai.

(Segue il collaudo del programma. Di nuovo il programma non funziona. I ragazzi tornano al banco e discutono.)

A4: Deve fare 180 meno 30. (Nota 11)

A1: Quindi 150.

A3: Perché un giro completo fa 180 ... Perché 90 più 90 ...

A2: Cioè, arriva qua ... e deve fare ...

A1: Come 1500?

A2: Sì, sì, ho sbagliato. E qua 180 meno 60 ...

(Il programma viene modificato correttamente)

A3: Provalo, dai.

Tempo: 19' 13"

(Gli studenti collaudano il programma che adesso funziona correttamente, per la gioia di tutto il gruppo.)

### 1.2.2 Le note

1. La base può benissimo coincidere con l'ipotenusa. Ciò che è interessante è che vi è un tentativo di analisi del testo.
2. La nota precedente è confermata dalla nota presente. Anche se applicato in maniera estemporanea, vi è un indubbio tentativo da parte del gruppo di impostare il problema seriamente.
3. Può essere visto come un tentativo di *Think*,<sup>13</sup> anche se insufficiente.
4. L'analisi del problema è già terminata. È durata complessivamente 50 secondi, cioè meno di niente.
5. Questo è l'errore che gli studenti solitamente fanno: la Tartaruga-Robot ruota di un angolo pari all'angolo interno della poligonale, non dell'angolo esterno.
6. L'errore viene ripetuto anche per il secondo angolo.
7. Normalmente si fa un primo collaudo di massima con il cavo USB, per risparmiare le batterie, ma la fretta è tale da provare il software subito dal vero. Naturalmente il collaudo fallisce, perché la tartaruga palesemente non chiude la poligonale.
8. Le prime intuizioni si orientano subito, giustamente, sulle rotazioni della Tartaruga-Robot.
9. La risposta è chiaramente intuitiva ed è rassicurante (infatti raccoglie l'approvazione dei compagni) perché "mantiene" l'angolo di 30° e ne cambia solamente di segno. Si noti che non viene aggiunta alcuna spiegazione all'ipotesi, né nessuno la chiede. Questa nota sarà oggetto di successiva riflessione.
10. Qui c'è un ulteriore passo, anch'esso intuitivo, verso la soluzione, ma cade nel vuoto. I ragazzi fanno un secondo collaudo con gli angoli negativi, ma fallisce anche questo.
11. Finalmente la soluzione corretta. Anche in questo caso, però, a nessuno passa per la testa di chiedere spiegazioni. Il terzo collaudo funziona.

### 1.2.3 Una riflessione importante

La trascrizione mette in evidenza una serie di errori metodologici degli allievi, ciò è fuori dubbio, ma quel che si vuole sottolineare è di altra natura e riguarda la nota numero 9.

Lo studente A4 ha un'intuizione sbagliata, non suffragata da alcuna argomentazione logica, ma gli studenti A3 e A1 si dichiarano immediatamente d'accordo.<sup>14</sup> Lo studente A3, in particolare, dichiara di aver capito il "ragionamento" del compagno di classe dopo circa 600-700 ms. Meno di un secondo.

Il docente esperto riconosce immediatamente la suddetta situazione: viene detto qualcosa da parte di un allievo o da parte dell'insegnante e dopo un

<sup>13</sup>Gli studenti conoscono le *thinking routine*, che però non sono state usate durante l'esercitazione. Si veda al tal proposito RITCHHART, CHURCH, MORRISON, *Making Thinking Visible* [19].

<sup>14</sup>Si noti che l'allievo A2 rimane in silenzio.

tempo brevissimo, assolutamente insufficiente anche solo per una riflessione molto superficiale, degli studenti si dichiarano d'accordo. Perché?

In tale circostanza gli allievi A3 e A1 hanno evidenziato una mentalità fissa. La teoria del *Mindset* sostiene che, nell'ottica della *Fixed Mindset*, prendersi tempo per riflettere significa ammettere di essere poco intelligenti (le persone intelligenti capiscono al volo: solo le persone poco dotate devono riflettere). Si noti che il comportamento di A3 e A1 non è un comportamento consapevole: comportarsi da "persone intelligenti" dopo un po' diventa automatico e inconsapevole.

La situazione appena descritta è stata commentata in classe (badando bene di non fare nomi) in modo da rendere consapevoli i ragazzi di tale comportamento. Si tratta di un atteggiamento diffusissimo e orientato a celare il proprio non-sapere,<sup>15</sup> che rappresenta esattamente ciò che non deve mai succedere in una scuola.

Per comprendere l'origine di tutto ciò basta rileggere la storiella di Carletto. L'errore viene punito a scuola, quindi bisogna cercare di nascondere le proprie lacune il più possibile, elaborando strategie di mimetizzazione scolastica piuttosto che chiedere apertamente spiegazione al docente (oltretutto infoltendo in tal modo la schiera degli studenti che sono costretti a pensare per capire quello che dice il professore!).

Che i ragazzi considerino la riflessione silenziosa un'attività per persone poco intelligenti lo si deduce anche dal prossimo esempio.

### 1.3 Un secondo esempio

Dopo che gli allievi hanno avuto modo di prendere confidenza con la movimentazione del robot, propongo loro un'altra esercitazione con un contenuto teorico maggiore e una richiesta di astrazione più evidente.

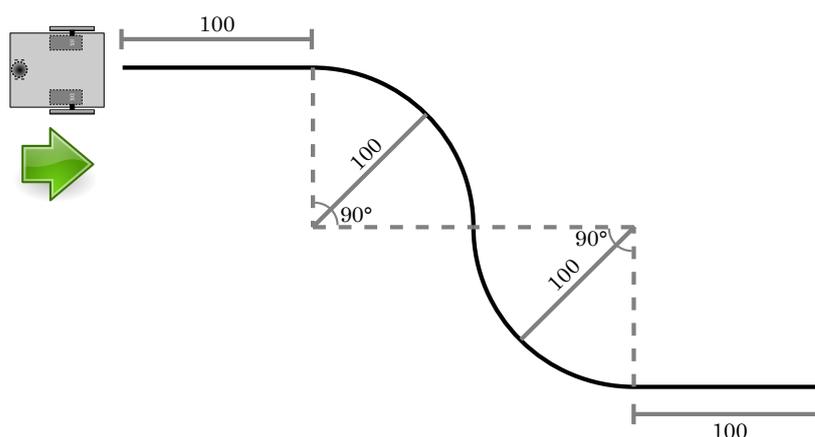


Figura 1.3: Percorso Tartaruga-Robot

<sup>15</sup>Si tratta del solito "obiettivo di evitamento" a cui si è già accennato nel paragrafo 1.1.

Chiedo agli allievi di calcolare la velocità angolare di ciascuna ruota motrice del robot in modo da tracciare sul pavimento la traiettoria evidenziata in fig. 1.3, essendo noti i dati relativi alla lunghezza dell'interasse, il diametro delle ruote, la massima velocità angolare che i due motori possono sviluppare e la distanza coperta dal robot per ogni passo.

Lo studio è quindi, volutamente, interamente teorico, in modo che i ragazzi imparino, pian piano, a destrutturare i problemi che vengono loro proposti e a trasportarli dal piano semantico a quello simbolico.

Anche in questo caso si propone una riflessione sulla trascrizione della registrazione audio fatta durante l'esercitazione. Si sono usate più o meno le stesse abbreviazioni utilizzate nell'esempio precedente: P = Professore, A1 = Allievo 1, A2 = Allievo 2, A3 = Allievo 3 e infine R = Ricercatrice INDIRE.

### 1.3.1 La trascrizione

Si riporta di seguito la trascrizione dell'attività di risoluzione del problema dato. Chi volesse risparmiare tempo può evitare di leggerla tutta: la parte effettivamente significativa è quella dalla nota 3 in poi.

Tempo: 00:10 (dall'inizio dei lavori)

A1: Allora, c'è scritto che deve fare 100 passi. Poi, in qualche modo, deve fare una rotazione di 90 ...

A2: Questa ruota deve fare un suo arco e questa deve farne uno maggiore ... [incomprensibile] ... Se qui c'è un arco di 100 (sic), qui dovrà farne uno maggiore. (Nota 1)

A1: Questa potrebbe essere la linea centrale, la linea guida, magari, così ci rendiamo conto della ...

A3: Questa qui sta ferma e questa qui ruota. Giusto?

A1: In teoria dovrebbero ruotare tutte e due.

A2: Sì, tutte e due in senso ...

R: Come si chiama questa classe? Seconda ...

A1: Terza ELI/A

R: Grazie

(L'allievo A3 rilegge lentamente il testo in inglese. Si capisce che sta tentando di approfondirne la comprensione. Interviene l'insegnante P che chiede di alzare un po' la voce)

Tempo: 03:00

A2: Allora ... Deve fare 2 mm per ogni passo, quindi qui dovrebbe fare 25 centimetri ...

A1: Sì, 25 centimetri ...

A2: Si può usare la distanza fra le ruote?

A1: Eh, dobbiamo usarla! La distanza dovrebbe essere di 11,5 ... Prof., la distanza fra le ruote è di 11,5?

P: Sì, sì, ma voi potete chiamarla "b", potete dare un valore ... Potete anche fare 10, cifra tonda. Non è quello il problema ... prendiamo un metro e misuriamo la distanza fra le ruote.

A3: Io direi di calcolare quant'è questo qua ... quello della circonferenza diviso quattro. Sappiamo che il raggio è 100.

P: State attenti che voi avete la tendenza di saltare subito alle conclusioni. Seguite questo. Voi avete la tendenza a risolvere subito tutto. Per piacere, no. (Nota 2)

Tempo: 04:43

(Segue un silenzio piuttosto lungo. Evidentemente i ragazzi sono un po' disorientati in seguito all'intervento del docente e non sanno bene cosa fare.)

Tempo: 05:00

A2: Allora ... la formula della circonferenza è  $2\pi r$ , quindi diviso 4 viene  $1/2\pi r$  ... Per 100 viene ... quindi  $50\pi$ .

A1: Sono 157 ...

A3: Passi.

A2: Sono 314 millimetri.

A3: E quanto equivale in millimetri? 314?

A1: 314.

A2: Quindi in totale è ... 200 per 2 ...

A1: Perché per 2?

A2: Perché ...

A1: Ah, sì, ok. E poi ... 314 per 2.

A3: Non so quanto può aiutare, ma questa roba è. La distanza che deve percorrere è questa. (Nota 3)

Tempo: 06:58

(Altro silenzio lungo. Forse i ragazzi stanno pensando a che strategia usare.)

Tempo: 07:24

A1: Quindi deve fare un totale di 1028 passi. (Nota 4)

A2: Deve rispettare queste due curve con tutte ... le due le ruote devono fare archi diversi nello stesso periodo di tempo. (Nota 5)

A1: Secondo me deve fare ... solo che ...

A2: Devi fare la metà della distanza delle ruote e trovi ... dal centro alla singola ruota. Dopo devi fare il raggio meno questa distanza per trovare questo arco e poi il raggio più questa distanza per trovare quest'altro. E poi sai quanto deve fare ogni singola ruota nello stesso periodo di tempo.

Tempo: 08:07

(I ragazzi si rendono conto di aver risolto concettualmente il problema e iniziano a fare i calcoli che, alla fine, risulteranno essere corretti.)

### 1.3.2 Le note

1. Uno dei ragazzi, lo stesso che avrà poi l'intuizione giusta, ha identificato subito il problema, ma poi non è riuscito a concretizzare il pensiero in una formulazione chiara. L'intuizione andrà subito persa. Il gruppo in questo frangente non ha funzionato bene: i compagni di classe avrebbero potuto soffermarsi e riflettere con un po' di calma sull'idea proposta;

2. il docente ha deciso di intervenire perché le *thinking routine* non venivano nemmeno considerate.<sup>16</sup> “Seguite questo” era l’esortazione a seguire quanto indicato nel testo, ovvero l’espressa richiesta di usare le *thinking routine*. Evidentemente, per i ragazzi, risolvere il problema è più importante che affrontarlo correttamente! L’intervento del docente ha sortito, comunque, uno scarso effetto;
3. i ragazzi hanno correttamente elencato i dati in loro possesso, anche se non sono stati capaci di evitare di fare qualche calcolo. Sembra quasi che gli studenti sentano il bisogno di effettuare dei calcoli, per avere l’impressione di risolvere qualcosa;
4. in realtà si tratta di millimetri, non di passi;
5. lo stesso ragazzo nominato nella nota 1 ha avuto di nuovo la stessa intuizione. Stavolta, complice il tempo avuto per riflettere, lo studente è riuscito a formalizzare esaustivamente il proprio pensiero. Il ragazzo ha capito che le due ruote devono percorrere nello stesso tempo archi di lunghezza diversa, il che significa che devono avere due velocità angolari differenti, facilmente calcolabili. Da qui in poi il problema è stato risolto concettualmente in pochi secondi. Anche i calcoli effettuati successivamente sono risultati corretti.

### 1.3.3 Un’altra riflessione

Si pone in evidenza un fatto: i ragazzi sono pervenuti alla soluzione dopo solamente un minuto di effettiva riflessione. Nei primi sette minuti gli studenti si sono affannati a fare calcoli più o meno inutili e semplicemente “riempitivi” che servivano, cioè, a dar loro l’impressione di “fare qualcosa” ovvero di risolvere il problema. Pur non potendo negare che i primi sette minuti sono senz’altro serviti a qualcosa, il primo pensiero razionale e argomentato è emerso dopo una semplice e serena riflessione, peraltro anche molto breve.

Sembra, ancora una volta, che riflettere sia un indice di scarsa intelligenza per qualche studente. Il bravo studente è “costretto” a fornire risposte fulminee, che sottolineino l’acuta intelligenza del ragazzo. In realtà testimoniano, una volta di più, una mentalità non tesa alla crescita personale, ma all’apparire bravi e intellettualmente dotati.

## 1.4 Conclusioni

Cambiare la cultura dell’errore nella scuola superiore italiana è un’impresa titanica. Significa trovare resistenze fra i colleghi - e passi - ma anche inspiegabilmente fra le famiglie e fra gli studenti. Ciò nondimeno, si tratta di un passaggio fondamentale per preparare il terreno ad un corretto e genuino apprendimento. Non si può pensare di vincere le sfide che il futuro ci prepara con le armi spuntate di una fiacca memorizzazione di alcuni concetti. A tal proposito si riporta il pensiero di Zygmunt Bauman, tratto da un suo intervento all’Univer-

<sup>16</sup>Era stato chiesto agli studenti di utilizzare la *thinking routine Think-Suppose-Prove*, espressamente pensata e progettata per una classe ad indirizzo tecnico. Gli studenti hanno però ignorato sostanzialmente la richiesta del professore.

sità di Padova, in occasione della Conferenza Annuale del Coimbra Group nel 2011:<sup>17</sup>

*«Penso che noi tutti potremmo convenire, senza ulteriori indugi, che la missione dell'istruzione, sin da quando essa fu formulata dagli Antichi con il nome di paidèia, era, rimane e probabilmente continuerà a rimanere quella di preparare i giovani alla vita. Se così è, allora l'istruzione, inclusa l'istruzione universitaria, si trova ora ad affrontare la crisi più profonda e critica nella sua storia così ricca di momenti difficili: una crisi che colpisce non soltanto una specifica consuetudine ereditata o acquisita, ma la sua vera raison d'être. Ora ci si attende che i giovani siano preparati a vivere in un mondo che - in pratica, ancorché non in teoria - rende nulla e vuota l'idea stessa "dell'essere preparato" - ovvero, adeguatamente qualificato e specializzato, e non colto di sorpresa dagli eventi e dalle tendenze mutevoli.»*

Alla luce delle sferzanti parole di Bauman, si deve convenire che la scuola è chiamata a interrogarsi severamente e fornire risposte coraggiose per poter affrontare gli anni che verranno. Ciò richiede un cambiamento di paradigma da parte del mondo della scuola, a cominciare dalla sua parte più reattiva, ovvero il corpo docente, che non è più procrastinabile. Dare il via a tale cambiamento modificando la cultura dell'errore che domina la scuola italiana è certamente un buon inizio.

---

<sup>17</sup>BAUMAN, *Le sfide all'istruzione nella modernità liquida* [3].

## Capitolo 2

# L'apprendimento

---

L'*incipit* del discorso di Bauman illustrato alla fine del precedente capitolo bene illustra la necessità di orientare l'azione scolastica verso una maggiore flessibilità da parte dello studente. Ciò significa muoversi con sicurezza in un mondo in cui l'informazione è spesso sovrabbondante e talvolta addirittura distorta. Significa avere la capacità di comprendere criticamente e fare propri, il più velocemente possibile, nuovi concetti.

Ciò coinvolge, soprattutto ma non solo, nelle materie ad indirizzo tecnico-scientifico, la capacità di ragionare logicamente in termini astratti.<sup>1</sup> Si tratta di una capacità che va allenata con perseveranza e che non è innata nell'uomo.<sup>2</sup> Significa allenare gli studenti alla destrutturazione del problema dato, affinché sia facilmente associabile alla teoria studiata in matematica, in fisica, ecc. e destreggiarsi con una certa maestria con il ragionamento simbolico, cosa che già di per sé costituisce un ostacolo non sempre facilmente superabile.

Quello che però si vuole sottolineare nel presente capitolo è l'importanza dell'atteggiamento del docente, che gioca un ruolo fondamentale. Una presenza "invadente", del tipo "so tutto io", rafforza negli studenti l'idea di non essere all'altezza (*Fixed Mindset*) con le conseguenze già succintamente illustrate nel capitolo precedente. Una buccia di banana piuttosto diffusa è rappresentata dalla convinzione del docente che gli studenti "non ragionino" abbastanza.

Nel prossimo paragrafo si vuole sostenere la tesi che il ragionamento è un'arte di difficile assimilazione e che noi docenti abbiamo le stesse identiche difficoltà dei nostri studenti. Noi insegnanti abbiamo una capacità a volte sorprendente di risolvere i problemi - anche non strettamente legati alla nostra materia - ma non perché siamo maestri del ragionamento astratto, ma perché abbiamo un bagaglio esperienziale enorme, che ci permette, attraverso ana-

---

<sup>1</sup>Per chi volesse approfondire l'argomento, un buon libro di trattazione scientifica è HOLYOAK, MORRISON, *The Cambridge Handbook of Thinking and Reasoning* [14].

<sup>2</sup>Daniel Kahneman sostiene che è piuttosto difficile frenare il pensiero associativo (predominante) per favorire il pensiero logico-simbolico, al quale non siamo naturalmente predisposti. Si veda a tal proposito KAHNEMAN, *Pensieri lenti e veloci* [16].

logie e associazioni, di affrontare correttamente le situazioni scolastiche più disparate. Ma lo studente, per questioni di età, è intrinsecamente sprovvisto di tale enorme esperienza ed è costretto a servirsi del solo ragionamento. Siccome non ne conosce la sottile arte, si serve del più comodo pensiero associativo, spesso con risultati disastrosi. Attenzione, però: quando l'esperienza non ci aiuta, abbiamo tutti la tendenza a usare il pensiero associativo. Il rompicapo proposto nel prossimo paragrafo ha la pretesa di porre in evidenza che anche noi docenti abbiamo spesso delle difficoltà nell'usare il ragionamento logico di fronte a situazioni nuove. Ovviamente, chi conosce già il problema o è già abituato al ragionamento logico non beneficerà di tale evidenza.

## 2.1 Un carico particolare

Il seguente rompicapo è tratto, e leggermente rielaborato, da una raccolta "non ufficiale e non autorizzata" di indovinelli ispirati e in parte riconducibili ad Albert Einstein.<sup>3</sup>

Una nave trasporta un carico particolare da una sponda all'altra di un lago. Il carico è formato da un certo numero di barre di piombo, da un serbatoio colmo d'acqua e da una certa quantità di tronchi d'abete. A metà del lago il comandante rileva un grave guasto all'imbarcazione e decide di buttare in acqua l'intero carico.



Figura 2.1: Un carico particolare

Prima butta nel lago il piombo, poi, in un successivo momento, l'acqua stivata nel serbatoio e infine il carico di legname. Cosa succede al livello del lago - si abbassa, si alza o resta uguale? - dopo che è stato buttato in acqua il carico di piombo? E dopo che è stato buttato il carico d'acqua (senza il serbatoio)? Infine, cosa succede al livello dell'acqua dopo che è stato buttato nel lago il carico di legname? Si tenti di affrontare il problema in autonomia.

Si ponga attenzione che non si è interessati alla correttezza della risposta, ma al tipo di ragionamento fatto.

<sup>3</sup>DEDOPULOS, *Einsteins Rätsel Universum* [9], pag.45.

### 2.1.1 La soluzione

Naturalmente ci sono diversi modi di affrontare il problema. Per il momento è importante sottolineare che le uniche conoscenze necessarie sono il Principio di Archimede e le correlazioni fra i pesi specifici di piombo, acqua e legno (ad esempio, legno d'abete).

Il Principio di Archimede, studiato sui banchi della scuola media, recita:

*“Un corpo immerso in un fluido riceve una spinta dal basso verso l'alto pari al peso del fluido spostato.”*

Anche la correlazione fra i vari pesi specifici non dovrebbe essere un grosso problema: Il piombo non galleggia nell'acqua, quindi pesa più dell'acqua e l'acqua pesa più del legno d'abete, dato che quest'ultimo galleggia nell'acqua. Chiarito ciò si può tentare di affrontare il problema.

La prima cosa da fare è chiarire cosa chiede il problema e la seconda tentare di destrutturarlo, dopo aver individuato le conoscenze necessarie per affrontarlo correttamente.

Viene chiesto cosa succede al livello del lago quando un corpo viene tolto dalla nave e buttato in acqua, questo è chiaro. Non è ancora chiaro come possa il Principio di Archimede essere utile - ammesso che lo sia - alla risoluzione del problema. Forse è necessario destrutturarlo e non pensare più a navi, laghi e carichi, ma a fluidi, contenitori e corpi.

Quando un corpo viene immerso in un fluido, quest'ultimo viene spostato, per cui il suo livello si innalza. Ciò è facilmente comprensibile se immaginiamo un piccolo contenitore riempito con dell'acqua: quando immergiamo un corpo nel liquido, il livello di tale liquido si innalza nel contenitore. Supponiamo ora una situazione un po' più complessa. Un contenitore più piccolo e magico - totalmente privo di massa, quindi privo di peso e di spessore - viene posto sull'acqua e dentro il contenitore magico poniamo un corpo. Detto corpo farà sì che il contenitore privo di massa si immerga nel liquido e così facendo farà alzare il livello nel contenitore grande. L'innalzamento del livello dipenderà dal peso del corpo: più esso sarà pesante e più si immergerà il contenitore magico, maggior acqua verrà spostata e più si innalzerà il livello.

Aggiungiamo ancora un po' di complessità al problema. Mettiamo nel contenitore magico del piombo. Quest'ultimo (carico di piombo) farà immergere il contenitore privo di massa (nave) nel liquido facendo innalzare il livello nel contenitore (lago), come rappresentato in fig.2.2a. Il problema è completamente destrutturato e tra parentesi si è mantenuta l'equivalenza con la rappresentazione semantica del problema.<sup>4</sup>

Se il piombo viene sollevato - sempre magicamente - in modo che non venga immerso nell'acqua né gravi sul contenitore piccolo, il livello cala e il contenitore senza massa né peso sfiorerà il pelo dell'acqua, come evidenziato in fig.2.2b. Si noti che in entrambe le figure è possibile contare, in virtù del reticolo tracciato, il numero di quadratini indicanti la presenza d'acqua e quindi, indirettamente, il volume del liquido. Contando detti quadratini si vince che

<sup>4</sup>L'importanza di riformulare simbolicamente - ovvero mediante le nozioni teoriche imparate in matematica, fisica, ecc. - problemi presentati semanticamente - ovvero fortemente attinenti alla vita reale - è ben evidenziata in EVANS, *Deductive Reasoning* in HOLYOAK, MORRISON, *The Cambridge Handbook of Thinking and Reasoning* [14]. pagg.169-184.

il volume d'acqua rimane costante nelle due figure, com'è ovvio che sia. Se ora si immerge il piombo nell'acqua, questo non galleggerà e si adagerà sul fondo e il livello tornerà a salire un po', *ma non fino al livello di fig.2.2a*. Questo perché, essendo il peso specifico del piombo maggiore di quello dell'acqua, il volume d'acqua spostato per galleggiare - quindi per ottenere la parità di peso - è maggiore del volume del piombo. Quando si getta il piombo in acqua,

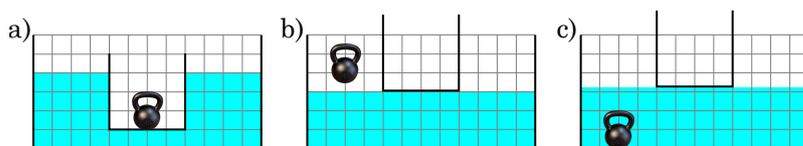


Figura 2.2: Il carico di piombo

l'unico volume spostato è quello della massa di piombo e quindi insufficiente per far alzare l'acqua al livello di fig.2.2a. Riproducendo i tre disegni su carta millimetrata e dando al corpo di piombo una forma rettangolare è possibile verificare che il volume d'acqua rimane costante. Abbiamo, in tal modo, risposto alla prima domanda.

Il carico d'acqua rende più facile l'analisi della situazione. Le due figure sottostanti evidenziano il livello quando il carico d'acqua è nel contenitore piccolo (fig.2.3a) e quando il contenitore è stato svuotato (fig.2.3b). Siccome il contenitore magico non ha né peso né spessore è immediato verificare che i due livelli restano uguali.

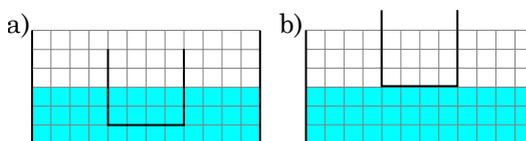


Figura 2.3: Il carico d'acqua

In tal modo abbiamo risposto anche alla seconda domanda del problema.

L'ultima domanda riguarda il carico di legna, ovvero un materiale con peso specifico minore dell'acqua. Quando la legna viene posta nel contenitore privo

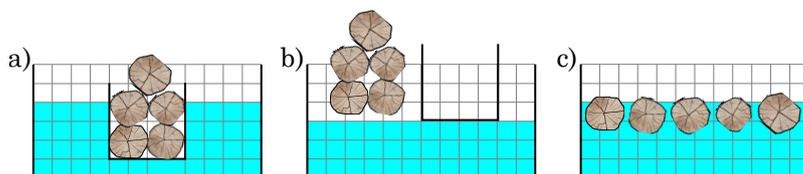


Figura 2.4: Il carico di legna

di massa, viene spostato un volume d'acqua il cui peso è pari al peso della legna. La legna galleggia, infatti, perché riceve una spinta dal basso verso l'alto pari al proprio peso (e al peso dell'acqua spostata). Quando la legna viene

buttata in acqua viene spostato un egual volume d'acqua, ossia tale da pesare quanto la legna, quindi anche in questo caso i due livelli saranno uguali.

Siamo quindi in grado di formulare una regola generale. Se si butta nel lago un carico con peso specifico maggiore di quello dell'acqua, il carico affonda e il livello dell'acqua sarà inferiore a quello precedente, perché il volume dell'acqua spostata è minore. Se si butta nel lago un carico con peso specifico minore o uguale a quello dell'acqua, i due livelli restano uguali, perché il volume d'acqua spostato è sempre lo stesso.

### 2.1.2 Com'è andata?

Con ogni probabilità, il docente in qualche maniera ce l'ha fatta a venire a capo del problema. Perlomeno potrebbe aver affrontato correttamente il rompicapo, probabilmente dimenticando la rappresentazione semantica del problema e orientandosi verso quella simbolica, magari semplificando qua e là quelle variabili non pertinenti al problema simbolico.

Quello che il rompicapo ha voluto porre in evidenza è che probabilmente anche il docente ha dovuto spendere qualche metaforica goccia di sudore, pur possedendo le conoscenze necessarie per affrontare il problema. Cosa significa tutto ciò? Diverse cose:

1. Ragionare è difficile. Richiede tempo, calma e la convinzione profonda di possedere le conoscenze necessarie. Noi, spesso, chiediamo ai nostri allievi di ragionare quando sono sotto stress: solitamente durante interrogazioni e compiti in classe. In tali situazioni mancano tutte e tre le precondizioni. Lo stress attiva il sistema limbico e in particolare l'amigdala, per cui lo studente viene spinto naturalmente<sup>5</sup> ad usare il pensiero associativo piuttosto che quello logico-matematico. Con i danni che verifichiamo ogni giorno in classe.
2. La conoscenza non basta, bisogna saperla usare. Quante volte abbiamo detto ai nostri allievi che la conoscenza è una chiave che apre tutte le porte? È vero, ma la chiave bisogna saperla usare. Tenere la chiave in tasca non serve assolutamente a nulla. Fuor di metafora, non dobbiamo preoccuparci che i ragazzi sappiano risolvere un integrale indefinito, ma dobbiamo preoccuparci che sappiano usarlo quando serve. Dobbiamo essere onesti e ammettere che noi docenti ci preoccupiamo molto che i nostri allievi acquisiscano delle conoscenze, ma ci preoccupiamo molto poco che i nostri allievi le sappiano usare. E soprattutto non lo insegniamo.
3. È importante ricondurre un problema in forma simbolica. Questo significa destrutturarlo e ricondurlo a quella teoria che abbiamo studiato a scuola. È inutile studiare che l'anacoluto è la rottura della regolarità sintattica della frase se poi il bel titolo del libro di Marcello D'Orta ("Io, speriamo che me la cavo") non è riconosciuto come tale. In molti casi i problemi scolastici sono formulati in forma simbolica: "Dato un triangolo rettangolo ...". Questo non aiuterà lo studente a usare le proprie conoscenze nella vita reale, perché, essendo il problema già destrutturato, non imparerà mai a destrutturarlo. Ma a scuola non si studiano le navi e i loro carichi: si studia il Principio di Archimede. Quindi il problema va ricondotto in

---

<sup>5</sup>Nel senso di "secondo natura".

una forma che permetta allo studente di utilizzare le proprie conoscenze. Spetta a noi docenti insegnarlo.

4. Noi insegnanti siamo molto bravi a risolvere i problemi, perché abbiamo un'esperienza enorme, accumulata in decenni di insegnamento. E quando siamo in difficoltà con il ragionamento (tutti facciamo fatica a ragionare!), ci aiutiamo con l'esperienza. Ma i nostri allievi sono privi di tale esperienza e a loro non è stato quasi mai insegnato a ragionare. Dobbiamo essere pazienti e insegnare loro la difficile arte del ragionamento. Dobbiamo insegnare loro che il pensiero associativo è utile nella foresta, ma spesso dannoso fra i banchi di scuola.<sup>6</sup>
5. Non sprechiamo le occasioni. Quando uno studente fa una domanda, ci sono docenti che sanno frenare l'impulso a far bella figura e rispondono: "Bella domanda! Non lo so. Vediamo insieme..." e diventa un'occasione per far vedere come si ragiona - anche barando, è vero, ma il fine giustifica i mezzi - e che ci si può arrivare anche senza essere geni o docenti. Un simile atteggiamento, oltre tutto, favorisce anche il rafforzamento della mentalità di crescita, sulla cui importanza non serve dilungarsi.

## 2.2 Un problema da terza elementare

Si vuole ora presentare un secondo problemino, adatto ai docenti delle scuole primarie e delle medie inferiori. Provocatoriamente sono uso dire ai colleghi che si tratta di un problema da terza elementare. Si tratta di una provocazione, ma fino ad un certo punto. Vediamo dunque il problema.

### 2.2.1 La somma delle cifre

Dato un numero di  $n$  cifre, con  $n$  non noto, di cui non si conosce il valore, si vuole eseguire la somma delle singole cifre. Di solito pongo il problema ad un allievo, dopo averlo messo con le spalle contro la lavagna e scrivendo il numero in modo che solo il resto della classe possa vederlo.

Con riferimento alle sole quattro operazioni aritmetiche, l'allievo può fare domande alla classe, ma senza fare in modo che, direttamente o indirettamente, possa conoscere il numero o il numero di cifre. Ad esempio, non può chiedere quanto vale il numero moltiplicato per uno, perché in tal modo conoscerebbe il numero.



Figura 2.5: Somma le cifre

<sup>6</sup>Non vorrei essere frainteso. Il pensiero associativo è utilissimo e non va demonizzato, ma non può essere l'unico usato a scuola. Alla lavagna bisogna essere capaci di argomentare con coerenza e di usare l'evidenza logica. Ciò significa un uso predominante del lobo frontale e delle forme di pensiero consapevoli, non solo associative.

Si deve considerare che, in realtà, l'algoritmo viene implementato su un PC, dovendo scomporre un numero nelle sue cifre. Il PC "non è capace" di isolare le singole cifre decimali, per il semplice motivo che il numero è espresso in base binaria. Detto ciò, si invita ad affrontare il problema.

### 2.2.2 La soluzione

La soluzione del problema è effettivamente piuttosto banale e coinvolge solamente due operazioni aritmetiche: la divisione intera e l'addizione.

Si valuti il seguente algoritmo:

1. si azzeri una variabile che dovrà contenere la somma delle cifre;
2. si sommi alla somma delle cifre il resto della divisione per dieci del numero dato;
3. si divida il numero dato per dieci. Il quoziente diventerà il nuovo numero da considerare.
4. se il quoziente vale zero, l'algoritmo termina e il risultato è nella somma delle cifre,
5. altrimenti si deve tornare al punto 2.

Commentiamo l'algoritmo eseguendolo passo-passo.

1a. La prima azione da fare, *una tantum*, consiste nell'azzerare la variabile destinata a contenere la somma delle cifre. Potremmo chiamare tale variabile "somma".

2a. A questa variabile bisogna sommare *il resto della divisione per dieci del numero dato*. Siccome il numero originale è 9254, si ha:

$$9254 : 10 = 925 \quad \text{con resto} = 4 \quad (2.1)$$

La suddetta divisione ha prodotto un quoziente pari a 925 e un resto pari a 4. Il resto va sommato a "somma", per cui, al termine del punto 2a dell'algoritmo si ha:

$$\text{somma} = 0 + 4 = 4 \quad (2.2)$$

3a. Il numero 9254 va diviso per 10 e il quoziente diventerà il nuovo numero a cui applicare l'algoritmo.

$$9254 : 10 = 925 \quad (\text{nuovo numero}) \quad (2.3)$$

4a. Siccome il quoziente dell'operazione precedente è maggiore di zero (vale 925) si deve tornare al punto 2.

2b. "Somma" vale 4. A questa variabile bisogna sommare il resto della divisione per dieci del nuovo numero. Siccome il numero ora vale è 925, si ha:

$$925 : 10 = 92 \quad \text{con resto} = 5 \quad (2.4)$$

La suddetta divisione ha prodotto un quoziente pari a 92 e un resto pari a 5. Il resto va sommato a "somma", per cui, al termine del punto 2b dell'algoritmo si ha:

$$\text{somma} = 0 + 4 + 5 = 9 \quad (2.5)$$

3b. Il numero 925 va diviso per 10 e il quoziente diventerà il nuovo numero a cui applicare l'algoritmo.

$$925 : 10 = 92 \quad (\text{nuovo numero}) \quad (2.6)$$

4b. Siccome il quoziente dell'operazione precedente è maggiore di zero (vale 92) si deve tornare al punto 2.

2c. "Somma" ora vale 9. A questa variabile bisogna sommare il resto della divisione per dieci del nuovo numero. Siccome il numero ora vale è 92, si ha:

$$92 : 10 = 9 \quad \text{con resto} = 2 \quad (2.7)$$

La suddetta divisione ha prodotto un quoziente pari a 9 e un resto pari a 2. Il resto va sommato a "somma", per cui, al termine del punto 2c dell'algoritmo si ha:

$$\text{somma} = 0 + 4 + 5 + 2 = 11 \quad (2.8)$$

3c. Il numero 92 va diviso per 10 e il quoziente diventerà il nuovo numero a cui applicare l'algoritmo.

$$92 : 10 = 9 \quad (\text{nuovo numero}) \quad (2.9)$$

4c. Siccome il quoziente dell'operazione precedente è maggiore di zero (vale 9) si deve tornare al punto 2.

2d. "Somma" ora vale 11. A questa variabile bisogna sommare il resto della divisione per dieci del nuovo numero. Siccome il numero ora vale è 9, si ha:

$$9 : 10 = 0 \quad \text{con resto} = 9 \quad (2.10)$$

La suddetta divisione ha prodotto un quoziente pari a 0 e un resto pari a 9. Il resto va sommato a "somma", per cui, al termine del punto 2d dell'algoritmo si ha:

$$\text{somma} = 0 + 4 + 5 + 2 + 9 = 20 \quad (2.11)$$

3d. Il numero 9 va diviso per 10 e il quoziente diventerà il nuovo numero a cui applicare l'algoritmo.

$$9 : 10 = 0 \quad (\text{nuovo numero}) \quad (2.12)$$

4d. Siccome il quoziente dell'operazione precedente vale zero, l'algoritmo termina e il risultato è contenuto in "somma", che vale 20.

### 2.2.3 Commento alla soluzione

Al termine della spiegazione della soluzione, solitamente si eleva un coro: "Ma questo non è un problema da terza elementare!".

Insisto: è un problema da terza elementare. Gli strumenti necessari a risolvere il problema sono tipici di tale classe della scuola primaria. Non servono concetti superiori. È sufficiente un atteggiamento razionale, una discreta conoscenza dell'addizione e una discreta conoscenza della divisione intera. Non serve altro.

È innegabile che lo studente medio ha studiato tali operazioni matematiche e possiede le conoscenze relative. Cosa impedisce a detto studente di risolvere il problema? Avanzo una timida ipotesi: la capacità di utilizzare le proprie conoscenze???

Si è già detto nel paragrafo 2.1.2 che noi docenti amiamo dire ai nostri discenti che la conoscenza è la chiave che apre tutte le porte. Propongo una piccola variante all'aforisma: "La conoscenza è la chiave che apre tutte le porte... Se la usiamo!". Avere in tasca la chiave che apre una determinata porta e non usarla, significa che la porta rimane chiusa. Ineluttabilmente.

Avanzo una seconda timida ipotesi: siamo sicuri che noi docenti insegniamo ai nostri allievi a usare le conoscenze che trasmettiamo loro? Personalmente temo che molti di noi si limitino a trasmettere le conoscenze, lasciando ai ragazzi la difficile impresa di imparare ad usarle. Ma si tratta, appunto, di un'impresa per niente facile. Dal punto di vista tecnico, abbiamo visto che ciò significa destrutturare il problema e trasportarlo dall'ambito semantico a quello simbolico. Non si tratta di una cosa facile. E così difficile che, qualche volta, si riesce a mettere in difficoltà un docente esperto con un problema di terza elementare.

Ritengo che il problema illustrato - che ho presentato ai miei allievi una ventina di anni fa! - sia utilissimo per far riflettere docenti e discenti sulla difficoltà e sulla bellezza del ragionamento. Ricordiamocene la prossima volta che diremo ai nostri allievi: "Ma ragiona, per favore!".

## 2.3 Il lavoro di gruppo

Il lavoro di gruppo può essere una notevole risorsa per gli studenti, a patto che venga gestito in maniera efficiente. Si tenga conto che anche se il clima in classe è idilliaco, vi è sempre timore da parte degli studenti a fare "domande sciocche" o a chiedere chiarimenti su un argomento fatto qualche mese addietro. In quei casi, il ragazzo ha la percezione di essere venuto meno ai suoi doveri di studente e preferisce non esporsi in tal senso. Tali freni inibitori sono minori fra compagni di classe, con i quali ci si sente più liberi di fare domande imbarazzanti, per cui il lavoro di gruppi, in tali casi, può essere di grande aiuto per gli studenti più deboli o più timidi.

Vi sono, però, ulteriori motivi che rendono il lavoro di gruppo una buona pratica in classe. Oggi vi è la tendenza a lavorare in *team* nelle industrie e conoscere i meccanismi e le dinamiche del lavoro di squadra può solo essere utile. La probabilità che nella vita lavorativa dello studente non vi sia spazio per il lavoro di gruppo è assai remota e lo sarà sempre di più man mano che passano gli anni. È quindi corretto che la scuola ne tenga conto e che prepari gli studenti a tale pratica.

Il lavoro di gruppo è anche formativo dal punto di vista umano. Se ben svolto comunica una visione molto democratica del lavoro e rafforza le relazioni interpersonali. Il ragazzo in difficoltà trae solitamente giovamento dal gruppo di lavoro, esattamente come trae giovamento chi aiuta il compagno di classe in difficoltà. I docenti sanno bene che spiegare un concetto aumenta la padronanza del concetto stesso.

Si è, però, più volte sottolineato che il lavoro di gruppo deve essere gestito in maniera efficiente, senza spiegare cosa significa. Vi è un pericolo di cui si deve tener conto, ossia che lo studente "bravo" monopolizzi le attività e le diriga da leader senza lasciare spazio agli studenti "inutili" e in difficoltà. Da questo punto di vista il lavoro di gruppo può essere addirittura pericoloso, perché rischia di spingere ancora più ai margini lo studente in difficoltà.

L'azione che si rende necessaria è quindi una "democratizzazione" del lavoro di gruppo. Il docente deve riservarsi la possibilità di modificare quei gruppi che non sono sufficientemente omogenei dal punto di vista delle conoscenze o che funzionano male per problemi di compatibilità interpersonali. Fatto ciò, il docente avrà cura di nominare due figure particolari: un *timekeeper*, avente funzioni di controllare i tempi e un portavoce, che avrà il compito di prendere appunti durante la discussione di gruppo. Infine il docente dovrà formulare l'argomento da discutere in gruppo in modo tale da permettere a tutti i componenti del gruppo di partecipare alla discussione. Questo significa che un eventuale problema complesso deve essere preventivamente scomposto dal docente in sottoproblemi più semplici da affrontare.

Ciò fatto, il *timekeeper* dà la parola al primo studente per un'unità di tempo, ad esempio un minuto, e tale studente *non dovrà essere interrotto per nessun motivo* dal gruppo. Eventuali obiezioni potranno essere elevate - in maniera educata e costruttiva - solo al proprio turno e non prima. Si badi che il *timekeeper* dovrà vigilare in maniera ferrea sul tempo e potrà arrivare a togliere la parola a chi parla oltre il tempo concesso. Durante le esposizioni il portavoce prenderà appunti, che serviranno a riassumere le idee migliori.

In tal modo si ottengono due effetti: tutti parlano democraticamente per lo stesso tempo e tutti rispettano le idee altrui. Gli studenti bravi vengono frenati un po', è vero, ma a tutto vantaggio del gruppo e dei ragazzi più deboli. I ragazzi in difficoltà potranno finalmente dire la loro e contribuire con le loro idee all'avanzamento dei lavori. In breve gli studenti, soprattutto se il docente lo farà notare, si accorgeranno che un gruppo che funziona non è semplicemente la somma delle parti, ma è molto di più. Idee apparentemente strampalate faranno venire in mente a qualcuno delle idee più concrete e attuabili. Il punto di vista dello studente in difficoltà potrà aggiungere ricchezza all'idea, oppure potrà obbligare a fornire ulteriori chiarimenti. Nei *team* ben amalgamati si scoprirà presto che avere nel gruppo un elemento un po' più debole degli altri è una ricchezza, perché obbliga a fornire chiarimenti e spiegazioni aggiuntive.

Man mano che tutti gli elementi del gruppo saranno intervenuti più volte, alcune idee si rafforzeranno e altre, più deboli, verranno abbandonate. Qualche elemento potrà fare proprio il punto di vista di un compagno e così via finché, pian piano, il gruppo convergerà su due-tre idee buone. Tali idee potranno essere poi oggetto di nuova discussione, finché il gruppo convergerà sulle effettive soluzioni.

Democratizzare il lavoro di gruppo ha effetti benefici per tutti, sia per gli studenti in difficoltà che per quelli bravi. All'interno di un gruppo di lavoro ben amalgamato tutti possono dare il proprio contributo, anche se in maniera diversa e con tempi diversi. Le prime volte il docente dovrà accertarsi che le regole del lavoro di gruppo vengano effettivamente messe in pratica, in particolare modo quella che prevede di non interrompere chi sta parlando. Quando le regole saranno state interiorizzate dal gruppo, attendere il proprio turno diventerà naturale.

Si rammenti che la Robotica Educativa trova la sua naturale collocazione nel lavoro di gruppo, per cui è bene curare le dinamiche al suo interno siano gestite nel modo più efficace possibile.

## 2.4 Il ruolo del docente

Il docente non è il protagonista della nostra storia: lo è lo studente. Il docente fa da "spalla" al protagonista e ha il compito di agevolare il lavoro dell'allievo. Lo fa in mille modi.

Il docente ha cura del clima in classe. Sta attento che nessuno studente venga emarginato e che non vi siano "fazioni" contrapposte. Sottolinea l'importanza della solidarietà fra compagni di classe e di come l'atto di solidarietà sia sempre ripagato nel suo stesso essere. Quando lo studente "bravo" spiega qualcosa allo studente in difficoltà, entrambi ne escono rafforzati. Chi spiega è costretto a chiarire ad altri ciò che ha compreso e questo richiede uno sforzo che rafforza la comprensione del concetto. Chi riceve la spiegazione può abbreviare il tempo di comprensione di un argomento e può fare quelle domande che troverebbe imbarazzanti se fatte a un docente. Se il clima è sereno e solidale è più facile apprendere.

Al docente non va bene tutto. Creare un clima costruttivo in classe non significa lassismo. Il docente è ben vigile affinché la classe non si abbandoni al dolce far niente. Ma se ciò dovesse succedere, non è una disgrazia, anzi. Significa avere l'occasione di aprire una discussione sull'operato della classe, sul proprio operato o su entrambi. Ciò significa anche che l'azione del docente è un'azione *a rischio*.<sup>7</sup> Non nel senso che la classe fa da cavia, ma nel senso che non si possono togliere tutte le incertezze dalla vita della classe. Apprendere significa mettersi in gioco, esplorare, rischiare, appunto, e questo vale sia per il discente che per il docente.

Il docente propone sfide che gli studenti possono affrontare. Senza sfida non c'è motivazione ad apprendere. La sfida, però, deve essere ben dimensionata, perché non deve scoraggiare lo studente. Ciò significa anche sfide personalizzate, se necessario. Una sfida ben dimensionata accresce la motivazione dello studente e una sfida mal dimensionata la diminuisce. Ma la totale assenza di sfida annoia lo studente. E ricordiamoci che l'importante non è che la sfida venga vinta, ma che venga affrontata e possibilmente con metodo.

---

<sup>7</sup>Che il docente debba assumersi dei rischi è molto ben spiegato in BIESTA, *The beautiful risk of education* [4].

Il docente cerca di smorzare il clima competitivo in classe. Apprendere non è una gara<sup>8</sup> ma è una conquista che procede per ciascuno studente per vie e velocità diverse. In una classe a basso tasso di competitività si respira una minor ansietà<sup>9</sup> ed è più facile apprendere.

Il docente evita, se può, di valutare. La valutazione ha effetti negativi sia quando premia che quando punisce,<sup>10</sup> perché distoglie l'attenzione dello studente dall'apprendimento genuino per reindirizzarla verso la valutazione. In regime di valutazione l'apprendimento è più superficiale e teso alla memorizzazione piuttosto che alla comprensione.

Il docente cerca di rendere la propria materia interessante. Ci sono docenti capaci di far odiare la propria materia agli studenti e ci sono docenti che sono capaci di farla amare. Entrare nella ristretta cerchia degli insegnanti del secondo gruppo non è molto facile, ma bisogna evitare come la peste di entrare nel primo gruppo. La Robotica Educativa rappresenta un modo leggero e divertente di far ragionare gli studenti. Il problema è che la Robotica non si presta facilmente per essere adattata a qualsiasi materia. Qualche modesto suggerimento verrà dato nel prossimo paragrafo.

## 2.5 La Robotica e le materie curricolari

In alcune Nazioni la Robotica Educativa e più in generale il Pensiero Computazionale è diventata materia curricolare sin dalla scuola primaria.<sup>11</sup> In Italia il Pensiero Computazionale può essere impiegato con relativa facilità nelle materie tecniche e con qualche difficoltà nelle materie umanistiche. Ipotizziamo qualche suggerimento per le varie materie.

**Matematica** Il già citato libro di Abelson e Disessa [1] permette una dettagliata esplorazione della matematica attraverso la tartaruga del MIT.<sup>12</sup> Viene esplorata la geometria piana e parte di quella solida. Propone esempi di suddivisione del piano mediante frattali o mediante le curve di Sierpinsky o di Hilbert. Contiene cenni di fisica e di analisi vettoriale, nonché problemi classici come i labirinti (algoritmo di Pledge), i cammini chiusi ed esempi di ricorsione. Si tratta, praticamente, di una fonte pressoché inesauribile di idee e proposte.

<sup>8</sup>Adesso vanno tanto di moda le gare di matematica, chimica, elettronica, fisica, neuroscienze, ecc. Hanno indubbiamente molti pregi, ma stimolano la motivazione degli studenti che sono già bravi e sottolineano il divario fra quelli bravi e quelli che non ce la fanno. Da questo punto di vista rappresentano un danno verso gli studenti in difficoltà.

<sup>9</sup>BANDURA, *Autoefficacia. Teoria e applicazioni* [2].

<sup>10</sup>Per una dettagliata esposizione dell'argomento si veda DECI, RYAN, *Self-Determination Theory* [6]. Una trattazione più sintetica si ha in DECI, RYAN, *Motivation, Personality, and Development Within Embedded Social Contexts: An Overview of Self-Determination Theory* in RYAN, *The Oxford Handbook of Human Motivation* [20]. Il filosofo cinese Chuang Tzu diceva, 2400 anni fa, che "le ricompense e le punizioni sono la forma più bassa di educazione".

<sup>11</sup>Negli Stati Uniti e nel Regno Unito tale scelta è già legge dello Stato.

<sup>12</sup>Come già detto la versione italiana del libro non è più in commercio. Si trova ancora la versione in lingua inglese: ABELSON, DISESSA, *Turtle Geometry*, The MIT Press, 1986.

**Fisica** La Fisica può essere studiata efficacemente attraverso il Coding. Nelle classi dove si fa Informatica (nei Licei Tecnologici, ad esempio) e si conosce un po' di C++, è possibile usare un ambiente di sviluppo gratuito come quello di Qt per sviluppare semplici simulazioni o semplici studi. Può risultare utile il seguente articolo di Andrea Disessa: *DISSA, Computational Literacy and "The Big Picture" Concerning Computers in Mathematics Education* [21], che riporta anche feconde riflessioni nel campo della fisica.

**Elettronica** In tale campo le possibilità sono letteralmente illimitate. Esiste anche una letteratura vastissima che verrà illustrata nelle prossime sezioni.

**Biologia** Un'idea originale potrebbe essere quella di utilizzare gli algoritmi genetici per sviluppare degli studi in campo genetico. Un testo di grande utilità in tal senso - anche se qualche adattamento si rende indubbiamente necessario - è GOLDBERG, *Genetic Algorithms in Search, Optimization and Machine Learning* [13].

**Italiano** Anche la lingua italiana può essere studiata mediante il Coding. Alcuni software di sviluppo di qualche anno fa elaboravano le frasi di interfaccia macchina-uomo "assemblandole" partendo da dei blocchi funzionali costituiti da soggetti, predicati e complementi. Un altro esperimento fatto anni fa nella mia scuola è stato quello di sviluppare un algoritmo che scrivesse autonomamente il profilo dello studente avendo come base di partenza i voti nelle diverse materie. Si tratta di un'esercitazione che potrebbe avere non pochi risvolti educativi per gli studenti.

**Informatica** Anche in Informatica, come in Elettronica, le applicazioni del Pensiero Computazionale sono praticamente infinite. Basta un minimo di fantasia.



**Parte II**

**GLI STRUMENTI**



## Capitolo 3

# Mindstorm

---

Il presente capitolo è dedicato al Mindstorm, il robot della LEGO<sup>®</sup>, utilizzato, tradizionalmente, dagli allievi più grandi della scuola del I ciclo. Le versioni alle quali si farà principalmente riferimento saranno quella in dotazione all'Istituto Malignani di Udine, ovvero il Mindstorm education 9797, versione base in tecnologia NXT e quella attualmente presente sul mercato, ovvero il Mindstorm education EV3, in tecnologia, appunto, EV3. La figura sottostante illustra la versione NXT.

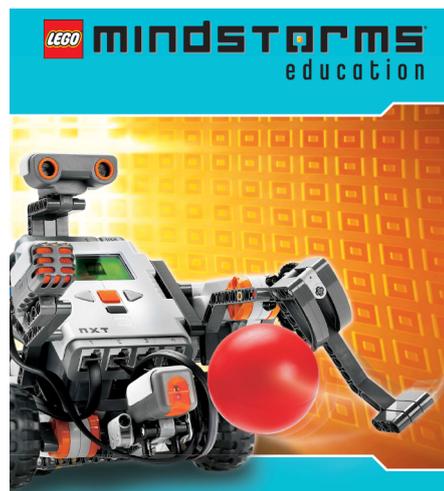


Figura 3.1: Mindstorms Education

## 3.1 I Mindstorms

Per parlare di Mindstorm si deve fare un po' di chiarezza, dato che il sito ufficiale non aiuta in tal senso e, parlando del robot della LEGO<sup>®</sup>, è facile fare confusione.

Esistono sostanzialmente tre generazioni di Mindstorm:

- **Mindstorm RCX.** Si tratta della prima generazione del robot della LEGO<sup>®</sup>, la più vecchia, ed è stato prodotto a partire dal 1998, diventando obsoleto con l'avvento della tecnologia NXT nel 2006. Era basato sul microcontrollore della Renesas H8/300 che si programmava mediante un'interfaccia ad infrarossi ed era frutto di una collaborazione con il MIT dell'era papertiana. L'ambiente di sviluppo era il Robolab, basato su tecnologia LabView della National Instruments;<sup>1</sup>
- **Mindstorm NXT.** Evoluzione piuttosto netta della precedente versione. Nacque nel 2006 equipaggiato con un microcontrollore ARM7 a 32 bit e connessione Bluetooth. Anche questa versione si programmava con un software basato su tecnologia LabView;<sup>2</sup>
- **Mindstorm EV3.** EV significa "evoluzione" della NXT ed è stato introdotto sul mercato nel 2013. Si tratta quindi della terza generazione di Mindstorm. L'EV3 utilizza un processore ARM9 a 32 bit e ha aggiunto uno slot SD per facilitare la programmabilità. È interfacciabile con applicazioni Android e iOS ed è commercializzato in 5 versioni differenti, che si distinguono non per la tecnologia, ma solamente per la dotazione dei mattoncini e dei sensori/attuatori.<sup>3</sup>

Nel presente capitolo si parlerà prevalentemente del Mindstorm EV3, che permette di costruire una serie impressionante di robot. Di seguito si elencano solamente i principali.

### 3.1.1 Tracker

Si tratta del robot consigliato dalla LEGO<sup>®</sup> per iniziare. Il nome è ottenuto



Figura 3.2: Track3r

sostituendo la lettera "E" con la cifra "3", ottenendo in tal modo un effetto

<sup>1</sup>WIKIPEDIA [30].

<sup>2</sup>*Ibid.* [30].

<sup>3</sup>*Ibid.* [31].

rovesciato e richiamando la terza generazione di robot LEGO®. È un cingolato capace di muoversi su tutti i tipi di terreno e possiede quattro utensili intercambiabili.<sup>4</sup>

### 3.1.2 Reptar

Tutti gli altri robot della serie EV sfruttano la cifra “3” come “E” rovesciata. Il seguente robot si chiama R3ptar (Reptar) e assomiglia ad un cobra. È alto 35



Figura 3.3: R3ptar

cm ed è capace di strisciare come un vero serpente. Sente i suoni e reagisce, se toccato, come un vero cobra. Pare sia particolarmente amato dai bambini, che non si lasciano impressionare dalle movenze e dalle fattezze. Impressiona invece gli adulti, o perlomeno l'autore: si veda il video indicato in bibliografia.<sup>5</sup>

### 3.1.3 Race Truck

È progettato da Laurens Valk, un giovanissimo ingegnere olandese già piuttosto conosciuto nell'ambiente della robotica educativa. Si tratta di un camion



Figura 3.4: Rac3truck

telecomandato con tanto di marce al quale è possibile attaccare un rimorchio. Può funzionare nativamente da *line follower*, essendo in grado di seguire una linea bianca tracciata sul terreno.

Ci sono molti tutorial in rete che insegnano come programmare un *line follower*. Si veda a tal fine la sezione dedicata ai tutorial Mindstorm.

---

<sup>4</sup>MINDSTORM TRACKER [35].

<sup>5</sup>R3PTAR [37].

### 3.1.4 Dinorex

È progettato da Lasse Stenæk Lauesen, altro giovane ingegnere, stavolta danese. Si tratta di un dinosauro - non un Tyrannosaurus Rex, però, come suggerisce il nome, ma di un Triceratops - capace di muoversi su quattro arti e di accorgersi della presenza di ostacoli sul proprio cammino. Anche in questo

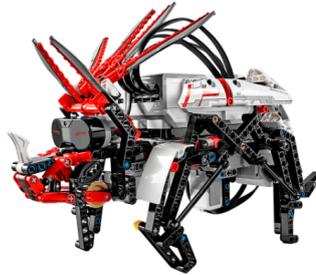


Figura 3.5: Dinor3x

caso ci sono dei simpatici video in rete.<sup>6</sup>

### 3.1.5 Everstorm

È il più truculento dei Mindstorm presentati. La LEGO<sup>®</sup> lo presenta come il



Figura 3.6: Ev3rstorm

“più avanzato. Dotato di bazooka e tripla lama rotante”. Ha la particolarità, come altri Mindstorm d'altronde, di essere controllabile mediante smartphone o tablet.

## 3.2 L'elettronica di Mindstorm

Si è accennato al fatto che le diverse versioni di Mindstorm differiscono fra loro per la tecnologia usata. Si intendeva con ciò indicare il tipo di microcontrollore usato per movimentare il robot. Nelle precedenti sezioni si sono viste le “forme” che i diversi Mindstorm possono assumere, ma quello che è veramente importante è come possa essere utilizzato in classe a scopi educativi.

---

<sup>6</sup>DINOR3X [36].

Ciò dipende in gran parte dall'elettronica a corredo e dall'ambiente di sviluppo, ovvero da quel software integrato che permette all'allievo di scrivere il programma, verificarlo, toglierne i malfunzionamenti e caricarlo infine nel Mindstorm. Non c'è limite a ciò che si può realizzare e ci sono dei video abbastanza sorprendenti a testimoniarlo.<sup>7</sup>

Nell'EV3 il caricamento avviene mediante un collegamento USB fra il PC, mediante il quale il software viene sviluppato, e il cervello del Mindstorm. Quest'ultimo ha la forma di un "mattoncino" di dimensioni relativamente grandi, come testimoniato nella figura 3.7.

In figura sono visibili le 4 connessioni RJ11 - indicate con 1, 2, 3 e 4 - che permettono di collegare altrettanti sensori del Mindstorm all'unità centrale. Sulla parte opposta, dal lato del display, vi sono altre 4 connessioni per altrettanti motori, più la connessione mini USB per il collegamento via PC.



Figura 3.7: Mattoncino EV3

L'EV3 Brick si connette con motori e sensori, per poter muoversi ed esplorare il mondo esterno. I principali attuatori e sensori di Mindstorm EV3 sono i seguenti:

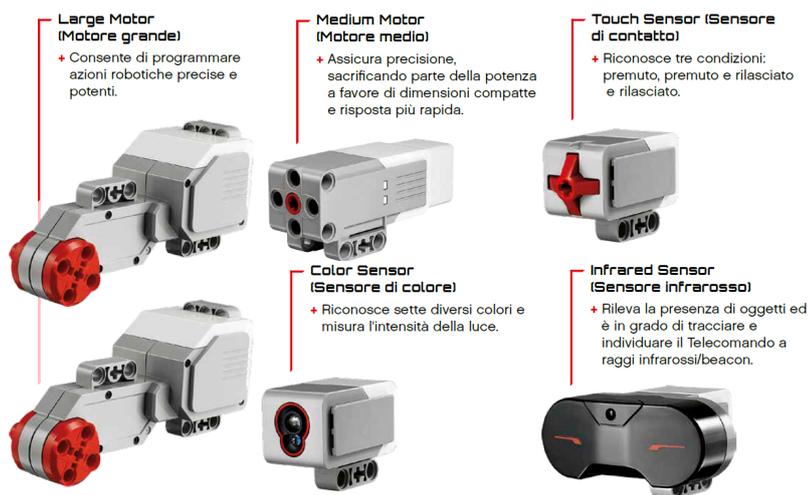


Figura 3.8: Motori e sensori

<sup>7</sup>RUBIKS CUBE SOLVER [38].

### 3.3 La programmazione di Mindstorm

Ora che abbiamo visto a grandi linee quale sia l'elettronica in dotazione al Mindstorm EV3 è naturale chiedersi come programmarla. I manuali di installazione ci insegnano come montare le varie parti del robot e come collegare elettricamente e meccanicamente l'elettronica, ma ciò non basta. Prima o poi si deve incominciare a scrivere delle istruzioni - non importa se con un linguaggio di programmazione iconico o testuale - utilizzando un *editor*, verificarne la correttezza sintattica e semantica attraverso un *compilatore* ed un *debugger* e, se il programma funziona esattamente come ci si aspettava, scaricarlo definitivamente nel Mindstorm.

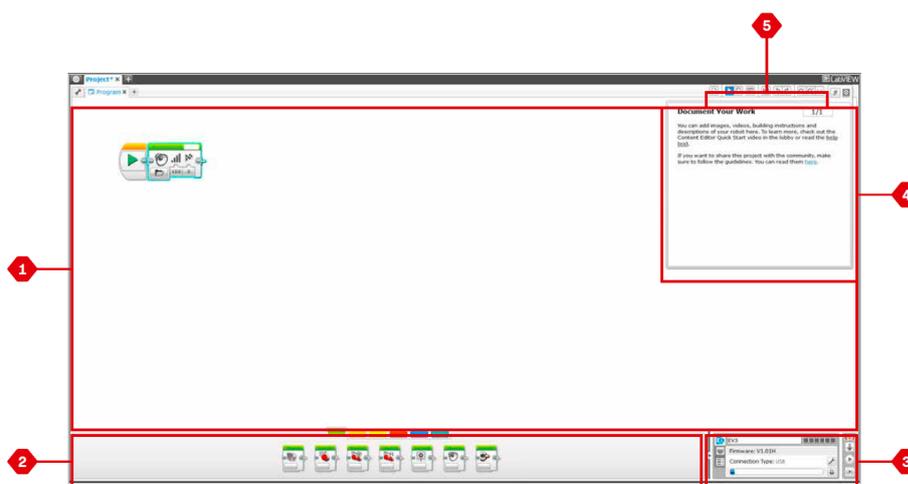


Figura 3.9: Interfaccia di programmazione

Per fare ciò si utilizza l'interfaccia di programmazione illustrata in fig. 3.9. La parte indicata con il numero 1 rappresenta l'area di programmazione, dove si utilizzerà il linguaggio iconico; la parte 2 contiene la cosiddetta *tavolozza di programmazione*, ovvero le icone da selezionare e porre nel dovuto modo nell'area di programmazione; la parte 3 è la cosiddetta pagina hardware che gestisce la comunicazione PC-EV3; la parte 4 è il *content editor* che raggruppa tutti i documenti del progetto;<sup>8</sup> la parte 5 è la barra degli strumenti del tutto simile a quella di un qualsiasi altro software per PC.

#### 3.3.1 Le tavolozze di programmazione

Le tavolozze di programmazione sono complessivamente sei e si distinguono per colore. Ogni colore rappresenta una tipologia distinta di elementi di programmazione, come di seguito indicato.

**Blocchi verdi** Sono i blocchi di azione. Essi controllano le azioni che il Mindstorm può eseguire, come, ad esempio, azionare dei motori, emettere un suono o una luce.

<sup>8</sup>Ogni programma va visto come un progetto, contenente sia le istruzioni che la documentazione a corredo.

**Blocchi arancioni** Sono i blocchi di flusso. Essi rappresentano le strutture di controllo del linguaggio. Mediante essi si possono eseguire dei cicli, prendere delle decisioni, rimanere in attesa di un evento e così via.

**Blocchi gialli** Sono i blocchi dei sensori. Permettono la lettura dei singoli sensori, correlando in tal modo le azioni del Mindstorm con il mondo esterno.

**Blocchi rossi** Sono i blocchi dei dati. Permettono la scrittura e la lettura delle variabili nonché l'esecuzione di operazioni logiche e matematiche fra di esse.

**Blocchi blu** Sono i blocchi di gestione dei file, delle connessioni e di altre operazioni relativamente evolute.

### 3.3.2 L'area di programmazione

Componendo i vari blocchi secondo un criterio stabilito dal programmatore si costruisce il programma che verrà poi compilato e caricato nell'EV3. Questa serie di azioni sta alla base della programmazione ed è ciò che distingue un programma da un altro. Un esempio di una porzione di programma è la seguente:

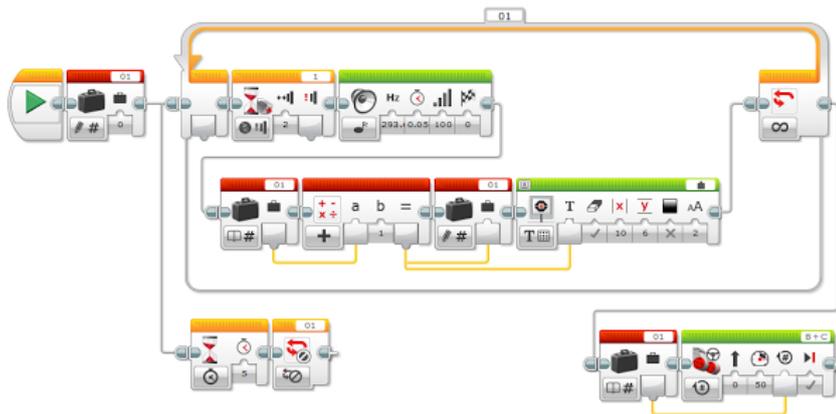


Figura 3.10: Esempio di programma

A prima vista può apparire complessa, soprattutto agli occhi del neofita, ma con la guida di un docente e costruendo il programma un po' alla volta, verificandolo e testandolo sul campo, si giunge a formare programmi anche relativamente complessi.

### 3.3.3 Enchanting: un linguaggio alternativo

L'esempio del paragrafo precedente sconsiglia l'applicazione del Mindstorm nelle classi frequentate dai più piccini. Sono stati sviluppati, però, altri linguaggi iconici, tutto sommato più semplici e adatti ai bambini delle prime classi delle scuole del I ciclo. Un esempio è *Enchanting*,<sup>9</sup> un linguaggio di pro-

<sup>9</sup>CALLEGARIN [22].

grammazione *open source*, anch'esso iconico, sviluppato in Canada<sup>10</sup> sulla base di linguaggio BYOB (Build Your Own Blocks), quindi assolutamente simile a Scratch, per intenderci. È stata annunciata in diverse occasioni anche la versione per EV3, ma non se ne rileva alcun riscontro pratico. Esiste anche un manuale di riferimento del linguaggio e dell'installazione in versione eBook,<sup>11</sup> ma il *download* sembra non funzionare sempre a dovere. L'autore ha dovuto ritentare diverse volte prima di poter scaricare il documento.

### 3.4 Riassumendo

Il LEGO® Mindstorm è uno strumento estremamente potente e versatile da utilizzare in classe. È ideale per bambini dai 9-10 anni in su, potenzialmente fino all'ultimo anno delle superiori. Ci si può limitare a far avanzare il robot creando delle figure geometriche sul terreno, fino a controllare processi di notevole complessità. Uno dei video che viene fornito insieme al kit Mindstorm NXT mostra un Mindstorm di una certa complessità che analizza un cubo di Rubik e lo risolve nell'arco di qualche minuto. Un video analogo è indicato in bibliografia, nella sezione dedicata a YouTube.<sup>12</sup>

Per quanto riguarda le classi inferiori, può essere utilizzato, ma in forma piuttosto primitiva. Consigliabile, in tal caso, l'uso di linguaggi più semplici, quali, ad esempio *Enchanting*. L'esperienza fatta dal professor Callegarin<sup>13</sup> può essere utile per il docente che intende utilizzare detto linguaggio. Si consiglia la lettura del documento riportato in bibliografia per chi fosse interessato ad un linguaggio alternativo.

---

<sup>10</sup>ENCHANTING ROBOCLUB [24].

<sup>11</sup>MONASH UNIVERSITY [27].

<sup>12</sup>YOUTUBE [38].

<sup>13</sup>CALLEGARIN [22].

# Capitolo 4

## Arduino

---

Arduino è un sistema formato da una *piattaforma hardware* e da un *ambiente di sviluppo software*. Nasce nel 2005 ad Ivrea dalla mente di Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis ed è distribuito come hardware e software libero, ovvero *open source*. Il nome è tratto dal bar di Ivrea frequentato dai progettisti, che a sua volta è intitolato ad Arduino d'Ivrea, marchese d'Ivrea dal 990 al 999 e re d'Italia dal 1002 al 1004.<sup>1</sup>

In virtù del basso costo<sup>2</sup> e di una vastissima libreria libera ha avuto una fulminea diffusione, prima negli ambienti hobbistici e subito dopo in quelli educativi. Il presente capitolo cercherà di illuminare quello che ormai è diventato un meraviglioso labirinto, in cui anche gli esperti fanno talvolta fatica ad orientarsi.

### 4.1 Le schede

Alla data di pubblicazione del presente opuscolo risultano essere 16 le diverse versioni di Arduino.<sup>3</sup> Si precisa che si sta parlando della sola scheda a microcontrollore, non delle schede di espansione che è quasi impossibile da elencare e che superano sicuramente il centinaio.

Arduino, solitamente, è utilizzato nelle scuole ad indirizzo tecnico o perlomeno nelle scuole superiori. Ciò è dovuto al fatto che Arduino non è un robot pronto all'uso, ma solamente il sistema elettronico. Ad esso va aggiunta la parte meccanica - solitamente un telaio dotato di due ruote motrici ed una ruota basculante - la sensoristica e i motori. Dato il costo obiettivamente basso - si dovrebbe riuscire ad assemblare tutto, comprando qua e là i singoli pezzi, con meno di €100 - se la scuola potesse contare su qualche ATA o collega con

---

<sup>1</sup>WIKIPEDIA [28].

<sup>2</sup>Ormai c'è un numero impressionante di diverse schede Arduino e di schede di espansione ma, molto grossolanamente, il costo di ciascuna scheda si aggira intorno alla ventina di Euro.

<sup>3</sup>*Ibid.* [28].

un po' di esperienza in campo robotico, sarebbe sicuramente una soluzione conveniente.

Le prossime tre sezioni parleranno di versioni particolari di Arduino: Arduino Uno, Arduino Lily Pad e Arduino Yun.

### 4.1.1 Arduino Uno

Arduino Uno, illustrato nella figura sottostante,<sup>4</sup> è stata una delle versioni di maggior successo nel campo della robotica educativa. Costa circa € 20 e si trova molto facilmente *on-line*. Chi ha la pazienza di attendere una lentissima spedizione via mare, può ordinare la scheda da qualche esportatore cinese a meno di metà prezzo.



Figura 4.1: Arduino Uno

Si tratta di una scheda equipaggiata con microcontrollore ATmega328 a 8 bit in tecnologia RISC.

### 4.1.2 Arduino Lilypad

Arduino Lilypad è pensato per la tecnologia *wearable*, ovvero *indossabile*. Per tale motivo esiste anche una versione flessibile - rappresentata in fig. 4.2<sup>5</sup> - particolarmente indicata per tale uso. Il costo è assai variabile: la versione non flessibile costa meno di € 10, mentre quella flessibile si aggira intorno ai € 20.



Figura 4.2: Arduino Lilypad

Anche in questo caso il microcontrollore usato è il ATmega328 a 8 bit.

<sup>4</sup>Foto SNOOTLAB [32].

<sup>5</sup>Foto WIKIPEDIA [28].

### 4.1.3 Arduino Zero Pro

Si tratta di uno degli ultimi arrivati in casa Arduino (vedi fig. 4.3).<sup>6</sup> Dal punto di vista tecnologico presenta una grossa novità: la possibilità di utilizzare un *debugger step by step*. Si tratta di una particolarità che facilita la rilevazione e correzione degli errori nei programmi. Si tratta di una novità molto attesa e gradita dai programmatori. A parte il prezzo: circa € 50.



Figura 4.3: Arduino Zero Pro

Utilizza un microcontrollore Atmel SAM D21 a 32 bit.

### 4.1.4 Gli Shield Arduino

Una delle particolarità di Arduino sono gli *shield*, ovvero le schede di espansione. Come già accennato sono moltissime, ma hanno tutte una particolarità: sono *stackable*, ovvero impilabili una sull'altra, come indicato nella figura sottostante.<sup>7</sup>

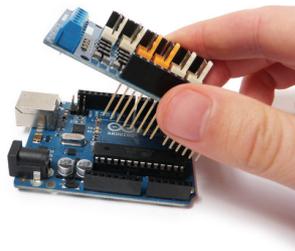


Figura 4.4: Arduino Motor Shield

L'impilamento può essere effettuato a più livelli, ponendo schede una sopra l'altra a seconda del bisogno. Naturalmente ciò comporta un aumento della complessità della programmazione di Arduino e richiede un minimo di esperienza o di spirito d'intraprendenza.

I più diffusi sono gli *shield* motori, che permettono un interfacciamento diretto a dei motori passo-passo o a dei servo motori. Esistono *shield* di attuazione (motori, display, relè, ecc.) e *shield* di sensoristica (ultrasuoni, inclinometri,

<sup>6</sup>Foto ARDUINO [23].

<sup>7</sup>Foto SARAFAN [25].

bluetooth, ecc.) che coprono qualsiasi necessità hobbistica o didattica. I costi dipendono dall'espansione, ma sono in linea con la *main board*.

## 4.2 L'ambiente di sviluppo

Come già detto, sono molti i linguaggi di programmazione utilizzabili con Arduino e di conseguenza sono diversi anche gli *ambienti di sviluppo*. Per ambiente di sviluppo si intende un software che raggruppa diversi *tool*: l'*editor* per scrivere il programma; il *debugger*;<sup>8</sup> il *tool* per programmare il microcontrollore, ecc.

Di seguito si fornisce qualche informazione utile per potersi orientare nella lettura della documentazione. Si sottolinea che le presenti pagine non vogliono essere un *tutorial* o un'introduzione ad Arduino. Nel capitolo dedicato alla documentazione vi è materiale in abbondanza e di ottima qualità, per cui si rimanda a detto capitolo e al materiale in esso indicato. A seconda del sistema operativo usato (Windows, Linux o Mac) può essere necessario interpretare detta documentazione con un po' di elasticità.

Quando si lancia l'IDE (*Integrated Developing Environment*) di Arduino, si apre una finestra simile alla seguente:

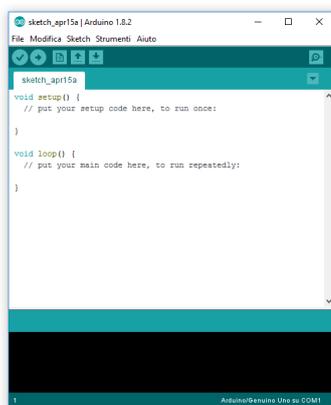


Figura 4.5: Arduino IDE

A questo punto, se è stato precedentemente installato ArduBlock,<sup>9</sup> si può decidere se programmare utilizzando un linguaggio evoluto e professionale, adatto a studenti delle scuole superiori - il C/C++ o, meglio, un suo adattamento - oppure un linguaggio più adatto ad allievi delle scuole del I ciclo, ovvero ArduBlock.

<sup>8</sup>Purtroppo solamente l'Arduino Zero Pro possiede un *debugger* professionale. Le altre versioni ne risultano prive. Dal punto di vista didattico ciò è un grave limite.

<sup>9</sup>Si veda MAFFUCCI [26] per una guida all'installazione di ArduBlock.

### 4.2.1 ArduBlock

ArduBlock è un *add-on* di Arduino. Viene visto dall'IDE di Arduino come un programma esterno che può essere utilizzato all'interno dell'ambiente di sviluppo.

Se l'installazione di ArduBlock è stata eseguita correttamente, nel menù Strumenti comparirà la voce ArduBlock, come evidenziato nella sottostante figura.

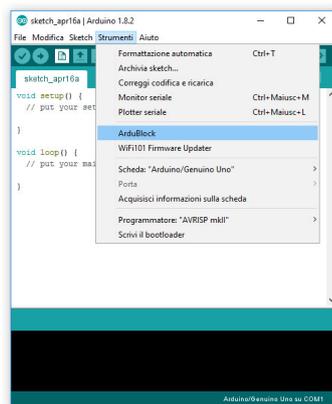


Figura 4.6: Arduino IDE (ArduBlock)

Richiamando tale voce si ha accesso all'IDE di ArduBlock, simile, per molti versi, a Scratch:

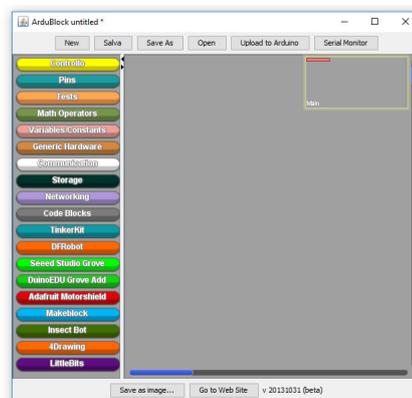


Figura 4.7: ArduBlock IDE

Da questo momento in poi si utilizza un ambiente di sviluppo parallelo all'Arduino classico e che non prevede la programmazione testuale classica. Risulta quindi adatto - almeno in una prima fase introduttiva - anche ai più piccoli.

Anche ArduBlock ha, però, le sue regole e la relativa documentazione va studiata attentamente se si vuole operare con efficienza con l'ambiente iconico. La prima regola che va rispettata - regola che si ritroverà anche nella pro-

grammazione classica di Arduino - è la suddivisione del programma in **setup()** e **loop()**. Un qualsiasi programma va, infatti, sempre iniziato con il tassello **program**, contenente, appunto, le due funzioni:

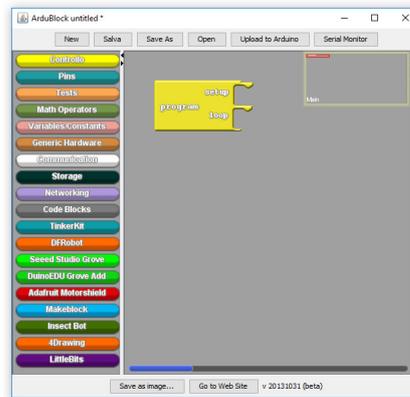


Figura 4.8: ArduBlock (setup-loop)

Compiuta questa operazione fondamentale, il resto è del tutto simile a Scratch. Il significato della tipologia dei singoli gruppi di icone va studiato ed assimilato prima di poter operare efficacemente.

La documentazione acclusa offre indicazioni di studio, sia in lingua inglese che in lingua italiana.

#### 4.2.2 C/C++

L'ambiente testuale di sviluppo è basato su linguaggio C/C++. Anch'esso uti-

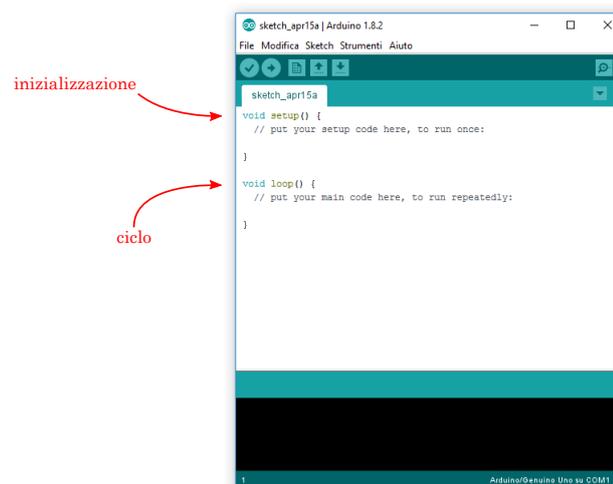


Figura 4.9: Arduino IDE (C++)

lizza la regola **setup-loop** e si coglie l'occasione per illustrarla in maggior det-

taglio. Un qualsiasi programma è sempre suddivisibile in due grandi blocchi: un blocco di inizializzazione e un ciclo infinito.<sup>10</sup>

Si deve quindi tenere a mente che una volta eseguite le istruzioni contenute nella funzione **loop()** esse vengono rieseguite automaticamente. Non vi è quindi alcuna necessità - peraltro inefficiente e in controtendenza rispetto alla filosofia Arduino - di porre un ciclo senza uscita nella funzione **loop()**.

Una volta terminata la scrittura del software è possibile scaricare il programma compilato nel microcontrollore di Arduino mediante un cavetto USB. Purtroppo, come già detto, la versione classica di Arduino non possiede alcun *debugger*, per cui la ricerca degli errori di programmazione va effettuata mediante stampa dei valori delle variabili in determinati *stati-macchina*. Tale operazione può risultare noiosa e inefficiente, ma è l'unico modo possibile per correggere gli errori software.

Se tale pratica dovesse risultare inappropriata didatticamente, ci si dovrebbe orientare verso la versione Zero Pro di Arduino.

L'ambiente C/C++ è comunque alquanto impegnativo e dovrebbe essere riservato solamente alle scuole superiori. Il linguaggio è infatti piuttosto complesso e va studiato attentamente se si vogliono ottenere risultati significativi. La scelta dell'ambiente di lavoro è importante e si deve tenere bene a mente quali obiettivi si vogliono perseguire prima di operare una scelta. A tal proposito si propone una riflessione che si considera importante nella prossima sezione.

### 4.3 Taglia e Incolla

In tutta la prima parte delle presenti pagine si sono sottolineati gli obiettivi che la robotica educativa dovrebbe perseguire. In moltissimi casi, però, si riscontra tutt'altro atteggiamento sia negli allievi/studenti che nei docenti. La ricerca del risultato - soprattutto quello spettacolare - diventa prevalente sulla effettiva crescita dello studente e si abbandonano velocemente i buoni propositi per il *tutto e subito*.

La rapida diffusione di Arduino non è dovuta al fatto che compia miracoli in campo didattico, ma al fatto che offre una vastissima libreria - a volte professionale, a volte meno - che può essere utilizzata senza sforzo alcuno e che offre risultati immediati e spettacolari. In tal modo il mondo degli hobbisti si è trovato nella possibilità di sviluppare programmi in maniera creativa e appassionata. In ciò, naturalmente, non si intravede alcun male. È esattamente ciò che è accaduto con l'avvento del PC, quando tutti, all'improvviso si sono sentiti esperti di computer.

Il problema reale è che tutto ciò *distoglie lo studente dal vero obiettivo: crescere*.

---

<sup>10</sup>Nei programmi scritti per PC il ciclo non è mai realmente infinito, nel senso che prevede sempre la possibilità di essere terminato, ad esempio mediante una voce di menù Esci o similare. Nei dispositivi *stand-alone*, quale ad esempio un robot non collegato al PC, il ciclo è realmente infinito, nel senso che non si può interrompere se non togliendo alimentazione al sistema. Si pensi, ad esempio, ad un orologio da polso, ad un telecomando o ad un televisore. Infatti quando si spegne un televisore in realtà non lo si spegne completamente, ma rimane in *stand by*. Ciò significa che un microcontrollore esegue - appunto, all'infinito - un micro-programma che rimane in ascolto di un comando proveniente dal telecomando.

Chiunque è capace di tagliare e incollare pezzi di codice o un'intera libreria nel proprio programma e ciò dà soddisfazione alla propria creatività, ma non aiuta a sviluppare quella capacità razionale che gli sarà utile nella vita adulta. In tal modo si spreca del tempo prezioso e si fa passare un'idea profondamente errata: che si possa eliminare ogni sforzo dallo studio. In realtà, proprio perché Arduino è per tutti, saper usare Arduino - con il metodo *taglia e incolla* - serve a ben poco. È come spacciarsi per tecnici elettronici perché si sa usare il telecomando della TV.

Per questo è importante che il docente tenga ben fermo il timone e non perda di vista la rotta che porta verso la crescita dell'allievo, affinché il giovane non diventi *nave senza nocchiere in gran tempesta*.

## 4.4 Riassumendo

I grandi vantaggi di Arduino sono i costi ridotti al minimo e la sua grande diffusione, con conseguente grande disponibilità di documentazione e di librerie. Gli svantaggi sono legati al fatto che il robot vada assemblato in proprio e che non esiste una vera e propria robotica educativa basata su Arduino. Inoltre, gran parte della documentazione si riferisce a librerie in linguaggio C/C++, non raramente commentate in inglese. Tutto ciò lo rende utile nelle scuole superiori e scarsamente utilizzabile nelle scuole del I ciclo, a meno che il docente non sia un esperto, o quasi, in tal campo.

In presenza di un *team* di docenti affiatati e con sufficienti conoscenze di C/C++, è possibile, però, preparare un piano didattico molto valido anche per i più piccoli.

## **Parte III**

# **LA DOCUMENTAZIONE**



## Capitolo 5

# Mindstorm e il Web

---

Per chi volesse utilizzare Mindstorm in classe, appare consigliabile orientarsi sull'ultima versione del robot della LEGO<sup>®</sup>. L'azienda dei mattoncini, infatti, pare scarsamente interessata a fornire assistenza sui robot della prima e della seconda generazione, che risultano essere introvabili nel sito ufficiale.

Si decide, pertanto, di riportare con prevalenza i link ai diversi tipi di materiale utilizzando il sito ufficiale della LEGO<sup>®</sup>, ma di migrare anche su siti non ufficiali all'occorrenza, in modo da fornire aiuto anche a coloro che utilizzano - o intendono utilizzare - versioni precedenti, ad esempio per motivi economici.

L'utente medio di Mindstorm è generalmente interessato a quattro tipi di informazione riguardo al robot:

- mattoncini di espansione e nuove versioni di Mindstorm;
- tutorial;
- librerie;
- miscellanea (linguaggi, eventi, ecc.).

Il presente capitolo esplorerà il Web alla ricerca delle suddette informazioni.

### 5.1 Hardware

Il sito ufficiale di Mindstorm ha una sezione chiara ed esaustiva di tutti i mattoncini in produzione. La sezione si chiama "Pick a Brick" e contiene veramente tutto quanto prodotto dalla ditta danese:

<https://shop.lego.com/it-IT/Pick-a-Brick#shopxlink>

Ogni singolo mattoncino, piccolo o grande che sia, è identificato graficamente e riporta a fianco il relativo prezzo. È possibile ruotare il pezzo da varie angolazioni per verificarne l'esatta forma, come pure è possibile sceglierne il colore. L'unica obiezione che si potrebbe muovere è data dall'assenza di quote meccaniche che identifichino il mattoncino nella sua effettiva grandezza. Il

problema non dovrebbe essere, però particolarmente pressante per un occhio un po' allenato al mondo LEGO®. La consegna è gratuita a partire da ordini maggiori di € 55,00.

Una sezione a parte è dedicata ai *mattoncini di potenza*. Non si tratta di blocchetti della Basilicata, ma di attuatori un po' particolari, quali servo motori, telecomandi, ecc. L'elenco completo è illustrato al seguente link:

<https://shop.lego.com/it-IT/Power-Functions>

Infine, da segnalare un servizio di assistenza clienti molto ben strutturato. È possibile contattare la ditta praticamente per qualsiasi tipo di problema: mattoncini rotti, mancanti, istruzioni di montaggio, ecc. Il link è il seguente:

<https://www.lego.com/it-it/service>

## 5.2 Tutorial

I tutorial della LEGO® hanno una particolarità: non contengono una sola parola. La sezione "Costruisci un robot" è assolutamente completa, anche se ad una prima superficiale occhiata può apparire scarna e incompleta. Diamole un'occhiata.

### 5.2.1 Costruisci un robot

Tutti i robot venduti in kit sono elencati e "cliccando" su uno di essi si accede ad una relativamente scarna pagina informativa, al cui centro appare un bottone blu, con la scritta "Costruisci ..." dove al posto dei puntini di sospensione c'è il nome del robot. Premendo detto bottone viene scaricato automaticamente la guida al montaggio in versione Pdf. Si tratta di guide di quasi un centinaio di pagine, molto ben fatte e progressive che, appunto, sono completamente grafiche e numerate.

Il montaggio è codificato in cosiddette *mission* per poter suddividere il lavoro e renderlo più semplice. Un esempio di "tavola" di montaggio è la seguente:

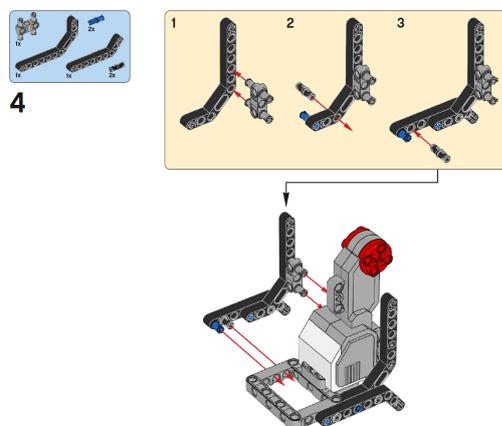


Figura 5.1: Tavola 4 Spik3r

Le tavole accompagnano gradualmente il costruttore al montaggio del robot e sono assolutamente estremamente efficaci. Si deve ricordare che l'assemblatore è il bambino, non il docente. A tal fine, infatti, sulla scatola di montaggio è chiaramente indicata l'età minima richiesta all'assemblatore.

Il sistema è estremamente efficace, per cui non si ritiene necessario dilungarsi oltre.

### 5.2.2 Video

I video forniti dalla LEGO® non sono da considerarsi esaustivi e i tutorial veri e propri non sono piuttosto pochi, ma di qualità: alla data di pubblicazione del presente opuscolo il sito ufficiale contiene solamente 7 video-tutorial. Due di essi sono indicati come video in lingua inglese, ma in realtà, i video inglesi sono quattro e i restanti tre sono pieni di meravigliosi suoni tecnologici, ma totalmente privi di parole espresse nella lingua di Dante.

Comunque la raccolta di video si trova al seguente indirizzo:

<https://www.lego.com/it-it/mindstorms/videos>

Si noti che accedendo ai video-tutorial, si viene automaticamente reindirizzati alla sezione "Impara a programmare", per cui si ritiene inutile documentare tale sezione.

## 5.3 Librerie

Le librerie di Mindstorm sono alquanto povere. Vengono fornite le informazioni base per imparare a programmare i robot e poco altro. Il sito da cui incominciare è il seguente:

<https://www.lego.com/it-it/mindstorms/downloads/download-software>

Il software di installazione contiene alcune *mission* software che guidano il neofita nella programmazione essenziale. Una volta completate si accede al livello superiore, che rimanda al già noto "Impara a programmare":

<https://www.lego.com/it-it/mindstorms/learn-to-program>

Non si può dire, quindi che il sito ufficiale abbondi di materiale software. Per ottenere ulteriori informazioni su programmi già pronti, si deve entrare nella sezione "App" e scaricare la "App EV3 Programmer", che contiene il software di sviluppo e i programmi per i robot indicati alla sezione "Robot dei fan". Di seguito i link:

Robot dei fan  
App EV3 Programmer

## 5.4 Miscellanea

Anche le librerie non ufficiali non sono straricche. Un buon sito dove trovare un po' di materiale è GitHub:

<https://github.com/>

Si tratta di un sito di hosting per progetti software.<sup>1</sup> Gli sviluppatori che decidono di condividere i loro lavori li memorizzano in questa *repository* gigantesca e li rendono disponibili per altri utenti. Ciascun programmatore decide se rendere pubblico o proteggere il proprio software. In caso di software protetto, GitHub funge da vetrina commerciale, mentre negli altri casi il software è libero ed eventualmente solo alcuni diritti sono riservati.<sup>2</sup>

Due esempi di software Mindstorm sono indicati di seguito. I due programmatori hanno usato rispettivamente i linguaggi Elixir e .NET:

<https://github.com/jfcloutier/ev3>  
<https://github.com/BrianPeek/legoev3>

I tutorial non ufficiali, in compenso, sono un po' più numerosi e dettagliati. Uno particolarmente ben fatto, non per la versione EV3 ma per la NXT, è il tutorial "NXT Programming for Beginners" di Neil Rosenberg:

[http://www.rocwnc.org/Beginning\\_NXT\\_Programming\\_Workshop.pdf](http://www.rocwnc.org/Beginning_NXT_Programming_Workshop.pdf)

Si tratta di un documento molto ben fatto, ma coperto da copyright. L'autore ne permette, però, l'uso gratuito per finalità non a scopo di lucro.

---

<sup>1</sup>WIKIPEDIA [29].

<sup>2</sup>Tipicamente è richiesto il riconoscimento della paternità del software e l'impedimento a trarne guadagno.

## Capitolo 6

# Arduino e il Web

---

Arduino è abbinato ad una documentazione tecnica impressionante. Si tratta soprattutto di librerie in C++, ma c'è anche molto altro e in tutte le lingue. Analogamente a quanto fatto nel capitolo precedente, si suppone che l'utente medio di Arduino sia generalmente interessato a quattro tipi di informazione:

- schede di espansione per Arduino e nuove versioni della *mainboard*;
- tutorial;
- librerie;
- miscellanea (linguaggi, eventi, ecc.).

Anche in questo caso, il presente capitolo esplorerà il Web alla ricerca delle suddette informazioni.

### 6.1 Hardware

Il sito ufficiale di Arduino offre una lista dei propri *reseller* - che è impressionante - al seguente indirizzo web:

<https://www.arduino.cc/en/Main/Buy>

Come si nota la lista dei rivenditori autorizzati è davvero infinita. I più attenti si saranno accorti che in Italia non viene venduto Arduino, ma Genuino. Si tratta di una triste storia legale fra gli inventori della celebre scheda a microcontrollore. Si preferisce soprassedere sul *gossip*. Gli interessati possono informarsi cliccando sul sottostante link. A tal fine si è cercata una fonte oggettiva e attendibile e la scelta è caduta sul Sole 24 ore (consultato alle ore 11:12 del 21 Aprile 2017):

Sole 24 ore - Arduino, ecco le ragioni dello scontro

Ovviamente, tornando a cose meno tristi, non può mancare un link al sito ufficiale di Arduino:

<https://www.arduino.cc/>

## 6.2 Tutorial

I tutorial sono uno dei punti di forza di Arduino, permettendo ad una semplice scheda elettronica di diventare un *totem*. Per tal motivo è necessario fare un po' d'ordine: si distinguerà in tutorial italiani, in lingua inglese e in video di ambedue le lingue.

### 6.2.1 Tutorial italiani

Alla data del 21 aprile 2017 l'autore ha contato 149 tutorial solamente sul sito ufficiale di Arduino, raggiungibili al seguente link:

<http://playground.arduino.cc/Italiano/Tutorials>

Per gli *absolute beginners* il sito ufficiale di Arduino riserva una sezione speciale a loro dedicata, con tanto di traduzione in italiano:

<http://playground.arduino.cc/Italiano/Newbie>

Consultate questo sito e non vi sentirete soli e abbandonati. C'è veramente tutto. C'è qualche utile indicazione anche nella pagina italiana del sito ufficiale:

<http://playground.arduino.cc/Italiano/HomePage>

Dai siti non ufficiali si ottengono comunque ottime documentazioni. Una versione carina, dedicata a chi non è del mestiere, è la seguente:

<http://www.zeppelinmaker.it/il-manuale-di-arduino-guida-completa/>

Ci si deve registrare e lasciare la propria mail, dopodiché si riceve sulla mail il manuale in allegato. Si tratta di un manuale che copre tutte le necessità del *newbie* in maniera chiara ed esaustiva. Davvero un tutorial per principianti ben fatto. Esiste anche il volume secondo edito da Edizioni LSWR e disponibile anche su Kindle. Una rassegna cartacea di tutorial per tutti i gusti è disponibile su Amazon.it al seguente link:

Tutorial Arduino ed altro

I professori Tiziana Marsella e Romano Lombardi hanno pubblicato i seguenti due tutorial, anch'essi molto buoni:

Elementi base del linguaggio di programmazione di Arduino  
Arduino ed applicazioni

Un ultimo sito interessante, contenente molti progetti e tutorial pronti per l'uso è il seguente:

<http://www.progettiarduino.com/progetti-e-tutorial.html>

### 6.2.2 Tutorial in lingua inglese

I tutorial in lingua inglese sono di qualità altrettanto buona e sono in numero notevolmente più alto di quelli italiani. Si elencano per dovere di completezza, ma non dovrebbero essere strettamente necessari, data la relativa completezza della documentazione in italiano. Se, invece, il lettore è anche docente CLIL, allora i seguenti siti sono fatti su misura.

Il primo tutorial che si presenta è quello di TutorialsPoint:

<http://www.tutorialspoint.com/arduino/>

Il suindicato sito dovrebbe essere già sufficiente per accontentare anche i palati più raffinati, ma chi non si accontenta si propongono i seguenti tre siti:

Learn Arduino - Contiene progetti, video e molto altro materiale

SparkFun - Un colosso dell'elettronica. Ottimo per docenti CLIL

Arduino Lessons - Semplice e introduce anche il linguaggio Phyton

### 6.2.3 Video italiani

Esiste una notevole quantità di video in rete. Esiste, però, anche il non secondario problema di visionare tutto ciò che si consiglia e ciò diventa decisamente *time consuming*. E purtroppo l'attività può riservare anche qualche sorpresa.

Molto allettanti sarebbero stati i video di Progetti Arduino, con un simpatico *youtuber* che illustra Arduino in un corso per principianti da 52 lezioni. Purtroppo il preparato signore scivola in maniera piuttosto *naïf* su un aspetto che non si può far finta di non vedere: utilizza un Arduino falsificato con tanto di marchio - anch'esso falsificato - in bella vista. Peccato, perché i video erano di buona fattura.

Allora si è deciso di optare per due simpatici studenti che propongono una *playlist* di 15 video, anch'essi piuttosto belli. La visione è stata interrotta al 13<sup>esimo</sup> filmato, intitolato "Come costruire un detonatore wireless". Naturalmente il contesto era il tradizionale fuoco d'artificio di capodanno, ma rimane ugualmente sconveniente.

Infine, sperando non sia sfuggito nulla, si propongono i seguenti video, anch'essi 15, di buona fattura:

MG Elettronica - Arduino Tutorial ITA

### 6.2.4 Video in lingua inglese

Per quanto riguarda i video in lingua inglese si è deciso di affidarsi ad un sito di qualità, senza dover verificare la bontà del tutorial video per video. Si è scelto il sito della Khan Academy, un colosso assoluto dell'organizzazione educativa senza scopo di lucro. Il solo sito presentato contiene un centinaio di video di qualità su Arduino.

[https://www.khanacademy.org/search?page\\_search\\_query=arduino](https://www.khanacademy.org/search?page_search_query=arduino)

Interessanti anche i video ufficiali di Arduino, non tanto incentrati sui tutorial, ma sul CTC (*Creative Technologies in the Classroom*). La *playlist* non è fornitissima ma conviene tenerla sott'occhio:

Arduino YouTube

## 6.3 Librerie

Si parla di librerie quando ci si riferisce a gruppi di istruzioni già scritte da altri in linguaggio testuale. A tal fine il linguaggio C++ si presta meravigliosamente, dato che uno dei capisaldi della programmazione ad oggetti è la classe. Le librerie, quindi, sono costituite da un insieme di classi che insieme propongono al programmatore una possibile soluzione ad un problema dato.

Ad esempio esiste la libreria per i motori passo-passo, rappresentata da un'unica classe. Mediante questa classe il programmatore può gestire i motori senza dover scrivere una sola riga di programma, ma utilizzando semplicemente quanto già scritto da altri.

Affinché una libreria possa essere usata è necessario includerla nel programma. A tal fine il sito ufficiale dedica una sezione - in inglese - a come una siffatta azione debba essere eseguita:

<https://www.arduino.cc/en/guide/libraries>

Una volta imparato come si includono le librerie si deve scegliere la libreria di nostro interesse al seguente indirizzo:

<https://www.arduino.cc/en/reference/libraries>

Si nota che le librerie sono suddivise per categoria. Alcune sono standard, il che significa che sono utilizzabili in qualsiasi *piattaforma Arduino*, ovvero in qualsiasi versione della scheda hardware (Uno, Duemilanove, Yun, ecc.). Appartengono a questa categoria, ad esempio, le seguenti librerie:

- **WiFi**. Gestisce la connessione a Internet mediante shield WiFi;
- **Stepper**. Gestisce motori passo-passo;
- **Servo**. Gestisce motori servo;
- **SD**. Permette la lettura e/o la scrittura di *memory card SD*;
- **LiquidCrystal**. Gestisce display a cristalli liquidi;
- ecc. ...

Altre librerie sono espressamente dedicate a certe versioni di Arduino. In tal caso si deve rispettare l'abbinamento. Ad esempio, la libreria **Audio** può essere usata solo Arduino Due; la libreria **Esplora** può essere usata solo con Arduino Esplora; la libreria **Robot** può essere usata solamente con Arduino Robot, e così via.

Per un elenco completo delle versioni Arduino e, soprattutto, per un corretto abbinamento versione-librerie, si deve fare riferimento all'elenco dei prodotti, al seguente link:

<https://www.arduino.cc/en/Main/Products>

Esistono, naturalmente, molte altre librerie, ma non sono ufficiali, per cui se ne sconsiglia l'uso.

## 6.4 Miscellanea

L'ultima sezione è dedicata alla raccolta di quelle voci che si richiamano un po' meno spesso: *language reference*, *l'Arduino store*, *l'IDE*, *Arduino per la scuola*, il

Forum, ecc. A costo di essere noiosi: sono moltissimi i siti nel Web che supportano le voci appena elencate. Alcuni siti sono molto professionali, altri meno; alcuni non hanno fini di lucro, altri sono strutturati per vendere Arduino anche a Banzi & Co. Per fortuna, il sito ufficiale di Arduino è molto ben strutturato, per cui non vi è grande necessità di visitare siti non ufficiali o di cui si fatica a capirne la professionalità. Di seguito una rapida carrellata degli indirizzi più utili del sito ufficiale.

#### 6.4.1 *Language reference*

Soprattutto i neofiti possono talvolta essere in imbarazzo quando sono alle prese con il linguaggio C/C++. In tal caso è utile la sezione *Language Reference*, accessibile al seguente indirizzo:

<https://www.arduino.cc/en/Reference/HomePage>

Se il compilatore dell'IDE rileva qualche errore e non si è in grado di risolverlo da soli, può essere utile fare riferimento a tale sezione in modo da confrontare le sintassi. Spesso eventuali errori si possono risolvere accedendo a tale sezione.

#### 6.4.2 *Arduino Store*

Ai tempi di Amazon un *Arduino Store* può risultare inutile o superfluo. Invece è utilissimo perché elenca i prodotti esistenti in maniera ordinata, efficiente ed esaustiva. Lo *store* è accessibile al seguente indirizzo:

<https://store.arduino.cc/>

#### 6.4.3 **Software e IDE**

Gli IDE vengono continuamente migliorati. Lo sviluppatore farebbe quindi bene a verificare periodicamente la disponibilità di nuovi ambienti di sviluppo, che garantiscono non solo *features* aggiuntive e migliorative, ma anche ambienti di sviluppo via via più stabili e robusti. L'indirizzo ove verificare la disponibilità di nuovi software o ambienti di sviluppo è il seguente:

<https://www.arduino.cc/en/Main/Software>

#### 6.4.4 *Arduino Educational*

Ovviamente Arduino ha una sezione dedicata alla scuola, con materiale dedicato agli studenti e materiale dedicato ai docenti. Particolarmente interessante il CTC Network (*Creative Technologies in the Classroom*), ovvero la rete di scuole utilizzando tecnologie creative.

<https://www.arduino.cc/en/Main/Education>

Molto bello anche l'Arduino Project Hub. Contiene centinaia di progetti che si possono esaminare e portare in classe modificandoli o meno. Sono una vera e propria miniera a cielo aperto:

<https://create.arduino.cc/projecthub>

### 6.4.5 Il Forum

Ovviamente non poteva mancare il forum ufficiale di Arduino. Naturalmente ce ne sono moltissimi e di buona qualità, ma citarli tutti o anche solo una minima parte è impresa ciclopica. Di seguito viene fornito il link al forum ufficiale di Arduino:

<https://forum.arduino.cc/>

## 6.5 Riassumendo

Arduino è un Universo a sé. Pretendere di esplorarlo tutto è praticamente impossibile. Conoscere bene il solo sito ufficiale risolve il 99% dei problemi che si possono incontrare. Certo, ha una pecca: gran parte del sito è in lingua inglese. Chi non è a proprio agio con la lingua di Shakespeare può effettivamente sentirsi pesce fuor d'acqua. In tal caso ci si deve rivolgere ai siti italiani. Sono tanti anch'essi. Ciò che è meno evidente è la qualità certificata tipica dei siti ufficiali, ma con un po' di esperienza e di buon senso si può navigare anche in quel mare un po' più agitato, ma anche più conosciuto.





# Bibliografia

- [1] Abelson Harold, Disessa Andrea, *La geometria della tartaruga*, Franco Muzzio & C. Editore, 1986
- [2] Bandura Albert, *Autoefficacia. Teoria e applicazioni*, Edizioni Centro Studi Erickson, 2000
- [3] Bauman Zygmunt, *Le sfide all'istruzione nella modernità liquida*, Padova University Press, 2011
- [4] Biesta Gert, *The beautiful risk of education*, Routledge, 2013
- [5] Boscolo Pietro, *La fatica e il piacere di imparare*, UTET Università, 2012
- [6] Deci Edward, Ryan Richard, *Self-Determination Theory*, The Guilford Press, 2017
- [7] Duckworth Angela, *Grinta. Il potere della passione e della perseveranza*, Giunti Editore, 2017
- [8] Dweck Carol, *Mindset*, Robinson, 2017
- [9] Dedopulos Tim, *Einsteins Rätsel Universum*, Ullmann Medien GmbH, 2015
- [10] Elliot Andrew, Dweck Carol, Yeager David, *Handbook of Competence and Motivation*, The Guilford Press, 2017
- [11] Gardner Howard, *Formae Mentis. Saggio sulla pluralità dell'intelligenza*, Feltrinelli, 2013
- [12] Goleman Daniel, *Intelligenza Emotiva*, BUR Saggi, 2001
- [13] Goldberg David, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1989
- [14] Holyoak Keith, Morrison Robert, *The Cambridge Handbook of Thinking and Reasoning*, Cambridge University Press, 2005
- [15] Kabat-Zinn Jon, *Full Catastrophe Living*, Bantam Dell Pub Group, 2013
- [16] Kahneman Daniel, *Pensieri lenti e veloci*, Mondadori, 2014
- [17] Krechewsky Mara, Mardell Ben, Rivard Melissa, Wilson Daniel, *Visible Learners*, Jossey-Bass, 2013

- [18] Papert Seymour, *Mindstorms. Children, Computers, and Powerful Ideas*, Basic Books, 1980
- [19] Ritchhart Ron, Church Mark, Morrison Karin, *Making Thinking Visible*, Jossey-Bass, 2011
- [20] Ryan Richard, *The Oxford Handbook of Human Motivation*, Oxford University Press, 2012

## Articoli

- [21] Disessa Andrea, *Computational Literacy and “The Big Picture” Concerning Computers in Mathematics Education*, Graduate School of Education, University of California at Berkeley, 2017
- [22] Callegarin Giuseppe, *Programmare NXT con Enchanting: un’esperienza e valutazioni*, raggiungibile all’indirizzo  
[http://www.robocupjr.it/3/wp-content/uploads/2012/03/01\\_Callegarin\\_.pdf](http://www.robocupjr.it/3/wp-content/uploads/2012/03/01_Callegarin_.pdf)

## Sitografia

- [23] Arduino, *Arduino Zero Pro*, raggiungibile all’indirizzo  
<http://www.arduino.org/products/boards/arduino-zero-pro>
- [24] Enchanting Robotclub, *Enchanting*, raggiungibile all’indirizzo  
<http://enchanting.robotclub.ab.ca/tiki-index.php>  
 ATTENZIONE: il sito è indicato “non sicuro” dai browser.
- [25] Sarafan Randy, *Arduino Motor Shield Tutorial*, raggiungibile all’indirizzo  
<https://www.instructables.com/id/Arduino-Motor-Shield-Tutorial/>
- [26] Maffucci Michele, *Programmare Arduino con ArduBlock*, raggiungibile all’indirizzo  
<http://www.maffucci.it/2014/09/27/programmare-arduino-con-ardublock/>
- [27] Monash University, *Robotics with Enchanting and LEGO® NXT*, raggiungibile all’indirizzo  
<http://monash-blockbooks.appspot.com/#enchanting>
- [28] Wikipedia, *Arduino (hardware)*, raggiungibile all’indirizzo  
[https://it.wikipedia.org/wiki/Arduino\\_\(hardware\)](https://it.wikipedia.org/wiki/Arduino_(hardware))
- [29] Wikipedia, *GitHub*, raggiungibile all’indirizzo  
<https://it.wikipedia.org/wiki/GitHub>

- [30] Wikipedia, *LEGO® Mindstorms*, raggiungibile all'indirizzo  
[https://it.wikipedia.org/wiki/LEGO\\_Mindstorms](https://it.wikipedia.org/wiki/LEGO_Mindstorms)
- [31] Wikipedia, *LEGO® Mindstorms EV3*, raggiungibile all'indirizzo  
[https://it.wikipedia.org/wiki/LEGO\\_Mindstorms\\_EV3](https://it.wikipedia.org/wiki/LEGO_Mindstorms_EV3)

## Immagini

- [32] Flickr Creative Commons,  
Autore: *Snootlab*,  
Titolo dell'opera: *Arduino Uno*,  
pubblicata con licenza CC-BY  
<https://www.flickr.com/photos/snootlab/6052455554/>
- [33] LEGO®, *Mindstorm Dinorex*, tratto da  
<https://www.lego.com/it-it/mindstorms/build-a-robot/dinor3x>
- [34] LEGO®, *Mindstorm Reptar*, tratto da  
<https://www.lego.com/it-it/mindstorms/build-a-robot/r3ptar>
- [35] LEGO®, *Mindstorm Tracker*, tratto da  
<https://www.lego.com/it-it/mindstorms/build-a-robot/track3r>

## YouTube

- [36] YouTube, *Dinor3x*,  
visualizzabile all'indirizzo  
<https://www.youtube.com/watch?v=l7ij64qnNLo>
- [37] YouTube, *EV3 R3ptar Robot*,  
visualizzabile all'indirizzo  
<https://www.youtube.com/watch?v=tPKTyTbpoBQ>
- [38] YouTube, *Rubiks Cube Solving Robot*,  
visualizzabile all'indirizzo  
<https://www.youtube.com/watch?v=uWkZ51yxvis>