

ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA  
CAMPUS DI CESENA

---

Scuola di Scienze  
Corso di Laurea in Ingegneria e Scienze Informatiche

STUDIO E SVILUPPO PROTOTIPALE DI UN  
FRAMEWORK SU PIATTAFORMA SNAP!  
PER ATTIVITÀ DI APPRENDIMENTO E  
CODING IN SCUOLE PRIMARIE E  
SECONDARIE

*Elaborato in*  
PROGRAMMAZIONE DI SISTEMI EMBEDDED

*Relatore*

Prof. ALESSANDRO RICCI

*Presentata da*

YLENIA BATTISTINI

*Co-relatori*

Dott.ssa. LAURA TARSITANO

Ing. ANGELO CROATTI

---

II Sessione di Laurea  
Anno Accademico 2018 – 2019



# PAROLE CHIAVE

Imparare ad Imparare  
Pensiero Computazionale  
Seymour Papert  
Mitchel Resnick  
Progetto MIA



*Alla mia famiglia,  
a chi ha sempre creduto in me  
e a me stessa.*



# Indice

<b>Abstract</b>	<b>ix</b>
<b>Introduzione</b>	<b>xi</b>
<b>1 Coding e Pensiero Computazionale</b>	<b>1</b>
1.1 Pensiero Computazionale - Jeannette Wing . . . . .	1
1.2 Seymour Papert . . . . .	4
1.2.1 Il Costruzionismo . . . . .	5
1.2.2 Il linguaggio LOGO . . . . .	6
1.2.3 Il Micromondo . . . . .	8
1.3 Mitchel Resnick - MIT Lifelong Kindergarten . . . . .	10
1.3.1 Scratch . . . . .	11
1.3.2 Le 4 "P" . . . . .	12
1.4 Coding . . . . .	13
<b>2 PROGETTO MIA</b>	<b>17</b>
2.1 Caratteristiche principali . . . . .	17
2.2 Le fasi di progetto . . . . .	19
2.3 Strategia di intervento . . . . .	21
2.4 Enti coinvolti . . . . .	24
2.4.1 Fondazione ENAIP Forlì-Cesena . . . . .	25
2.4.2 FabLab Romagna . . . . .	28
2.4.3 Tecnopolo Forlì - Cesena . . . . .	29
2.4.4 Criad Coding . . . . .	30
2.4.5 Scuola Primaria 3 Circolo Didattico Cesena . . . . .	32
2.4.6 Scuola Secondaria di Primo grado Comprensivo SMS Viale della Resistenza - Cesena . . . . .	33
2.4.7 Scuola Secondaria di Secondo grado IIS "Pascal-Comandini" di Cesena FC. . . . .	33
2.5 Lavorare in aula . . . . .	35
2.6 Obiettivo della tesi . . . . .	36

<b>3</b>	<b>STAZIONE METEOROLOGICA: hardware e software</b>	<b>39</b>
3.1	HARDWARE . . . . .	40
3.1.1	Si7021 Temperatura ed umidità . . . . .	41
3.1.2	BMP280 Pressione atmosferica . . . . .	42
3.1.3	SGP30 Concentrazione di CO2 e TVOC . . . . .	43
3.1.4	PMS5003 Qualità dell'aria . . . . .	43
3.1.5	Led RGB WS2812B . . . . .	46
3.1.6	Servo Motore . . . . .	47
3.2	REST API . . . . .	47
<b>4</b>	<b>Snap! Build Your Own Blocks</b>	<b>53</b>
4.1	La nascita di Snap! . . . . .	53
4.2	User Interface . . . . .	55
4.3	Scrivere un programma in Snap! . . . . .	56
4.4	Programmazione Object Oriented con Sprites . . . . .	60
<b>5</b>	<b>Sviluppo prototipale di un Framework su piattaforma Snap! per attività di apprendimento e coding</b>	<b>63</b>
5.1	Framework su piattaforma Snap! . . . . .	64
5.2	Idea iniziale . . . . .	64
5.3	Progettazione e Sviluppo . . . . .	65
5.4	Diversi livelli di astrazione . . . . .	75
5.4.1	Scuola Primaria . . . . .	78
5.4.1.1	Scienze . . . . .	79
5.4.1.2	Storia . . . . .	80
5.4.1.3	Geografia e Musica . . . . .	82
5.4.1.4	Italiano . . . . .	86
5.4.2	Scuola Secondaria di Primo Grado e Scuola Secondaria di Secondo Grado . . . . .	89
5.5	Possibili ampliamenti . . . . .	90
	<b>Conclusioni</b>	<b>91</b>
	<b>Ringraziamenti</b>	<b>93</b>
	<b>Bibliografia</b>	<b>95</b>



## Elenco delle figure

3.1	Rest Api . . . . .	49
3.2	Rest Api . . . . .	50
4.1	Schermata Snap! . . . . .	55
4.2	Esempio di script base in Snap! . . . . .	56
4.3	Tipologie di input per la creazione dei blocchi . . . . .	59
4.4	Schermata di esportazione blocchi a scelta . . . . .	59
4.5	Idea di script che sfrutta OOP con l'invio di un messaggio . . . . .	61
5.1	Editor di blocchi-Configura stazione meteo ip. . . . .	67
5.2	Editor di blocchi-Leggi temperatura. . . . .	68
5.3	Editor di blocchi-Leggi temperatura(Comando). . . . .	69
5.4	Editor di blocchi-Accendi led RGB. . . . .	70
5.5	Editor di blocchi-Gioco di colori. . . . .	71
5.6	Editor di blocchi-Invia lista di dati. . . . .	74
5.7	Tre diversi script per il monitoraggio. . . . .	77
5.8	Script Storia- Sprite termometro . . . . .	80
5.9	Script Storia-Sprite Girandola . . . . .	81
5.10	Script Storia-Stage . . . . .	82
5.11	Script Geografia-Nubi Script . . . . .	83
5.12	Script Geografia-Sole Script . . . . .	84
5.13	Script Geografia-Foglia Script . . . . .	85
5.14	Script Geografia-Neve Script . . . . .	85
5.15	Script Italiano-Sprite Mario . . . . .	87
5.16	Script Italiano-Sprite Goccia . . . . .	88
5.17	Script Italiano-Sprite Nubi . . . . .	88
5.18	Script Italiano-Sprite Stormo . . . . .	88



# Abstract

*”Perché programmare? E’ diventato un luogo comune identificare i giovani come nativi digitali a causa della loro apparente agilità nell’interagire con le tecnologie digitali. Ma il fatto che siano considerati dei nativi digitali li rende davvero fluenti con le nuove tecnologie? Anche se interagiscono con i media digitali tutto il tempo, pochi sono davvero in grado di creare i propri giochi, le proprie animazioni o simulazioni. In effetti è quasi come se potessero leggere senza però saper scrivere. Ciò che manca è quindi la capacità di sfruttare tali tecnologie per progettare, creare e inventare cose nuove, per i quali è necessario il coding. [...] Programmare espande notevolmente la portata di ciò che puoi creare con il computer e di ciò che puoi imparare. In particolare, il coding supporta il Pensiero Computazionale, grazie al quale poter imparare importanti strategie di problem-solving e progettazione e i cui benefici vanno ben oltre ai domini dell’informatica. Infine, poiché la programmazione coinvolge la creazione di rappresentazioni esterne dei tuoi processi di problem-solving, fornisce l’opportunità di riflettere sul proprio pensiero, nonché riflettere sull’opinione che abbiamo di noi stessi”[7] <sup>1</sup>.*

Il computer può diventare uno strumento per fare scuola.

L’utilizzo didattico dell’elaboratore implica un cambio di atteggiamento mentale nei suoi confronti, poiché esso, può diventare un ambiente utile per fornire ai bambini nuove possibilità di apprendere, di pensare, di migliorare le proprie capacità di approccio ai problemi costruendo le proprie conoscenze attivamente. *”Dovrebbe essere il bambino a programmare il computer e non il computer a programmare il bambino”*, queste le parole di Seymour Papert [1] creatore di uno dei linguaggi di coding più conosciuti: LOGO. Papert considera il bambino come un vaso da riempire ed *”il computer come creta con cui costruire una scultura”*[9]. Rimane il paradosso *”Perché non insegniamo ai bambini ad imparare, pensare e giocare?”*. [20]

---

<sup>1</sup><https://www.youtube.com/watch?v=Ok6LbV6bqaE>



# Introduzione

Il pensiero Computazionale è il " *processo mentale che sta alla base della formulazione dei problemi e delle loro soluzioni*" [2]. Lo studio del pensiero computazionale è prezioso non solo per gli appassionati e/o programmatori del mestiere ma per qualsiasi studente.

L'insegnamento dell'informatica non consiste solo nel formare migliori sviluppatori software, per questo motivo, è necessario introdurre nuovi percorsi di educazione all'informatica nelle scuole.

In passato diversi studiosi hanno proposto framework in grado di rendere più efficace l'apprendimento di tali tematiche.

Nel 2006, Jeannette Wing definisce il pensiero computazionale come: "*il processo necessario per la formulazione e soluzione di problemi in forme comprensibili da agenti in grado di processare informazioni*". Il pensiero computazionale si propone di applicare le nozioni acquisite in discipline eterogenee e nella quotidianità.

Secondo il Costruzionismo di Seymour Papert il computer costituisce un nuovo mezzo di apprendimento tramite cui gli studenti diventano creatori del proprio processo di crescita: *il ruolo del computer è come quello della creta con cui costruire una scultura*.

Il linguaggio LOGO creato e ideato dallo stesso Papert permette di applicare questi concetti costruendo, manipolando, sbagliando, ricostruendo e testando. L'errore non è più visto in ottica negativa, ma come un aspetto costruttivo del processo di apprendimento.

Il Capitolo 1, dopo aver illustrato il pensiero computazionale, si concentra sul coding spiegando, non solo come è "nato" ma, anche come sta cambiando il metodo di apprendimento didattico nelle scuole italiane. Successivamente presenta la figura di Seymour Papert, del costruzionismo includendo la storia del linguaggio LOGO. In ultima nota il primo capitolo include la figura di Mitchel Resnick, successore di Papert e creatore del linguaggio Scratch.

Il Capitolo 2 è un capitolo prettamente teorico in quanto spiega nel dettaglio l'ambiente Snap! e la modalità di utilizzo dei blocchi per la creazione degli script.

Il Capitolo 3 entra nel vivo del progetto MIA descrivendone le specifiche a livello territoriale mettendo in evidenza tutti gli enti coinvolti ed il loro relativo contributo. Questo capitolo racconta l'apporto di questa tesi che mira a realizzare una libreria Snap! identificandone diversi livelli di astrazione da utilizzare all'interno del progetto permettendo agli alunni di programmare e progettare la propria stazione meteorologica. Il Capitolo 4 entra nello specifico della stazione meteorologica in prevalenza dell'hardware spiegando come è stata costruita, quali sono i sensori e gli attuatori utilizzati illustrando le rest api realizzate per poterci comunicare.

Il Capitolo 5 è il capitolo principale della tesi in quanto espone nello specifico la libreria partendo dall'idea iniziale per poi arrivare alla progettazione e allo sviluppo ponendo l'accento sui vari livelli di astrazione realizzati consentendo a tutti gli studenti di potersi esprimere e poter lavorare nel modo migliore e più conforme alle loro capacità e alla loro età.

In ultimo sono presenti le conclusioni ed i ringraziamenti a tutti coloro che mi hanno portato alla realizzazione di questa tesi.

# Capitolo 1

## Coding e Pensiero Computazionale

Negli ultimi anni si parla tanto di pensiero computazionale e coding, sempre più spesso fanno parte dei programmi didattici nelle scuole ma, realmente che cosa significano questi due termini?

Il pensiero computazionale è una strategia di pensiero chiara, logica e operativa per la risoluzione dei problemi non solo scientifici ma, anche quotidiani in modo personale e creativo.

Molte scuole indicano tra le attività per i ragazzi ore di coding e di pensiero computazionale. In Italia la legge del 2015 ha collocato il pensiero computazionale tra gli strumenti culturali per la cittadinanza, cioè tra quelle abilità e competenze che realizzano la piena cittadinanza.

Il pensiero computazionale merita di essere coltivato e applicato in modo interdisciplinare perché costituisce una sorta di fertilizzante che prepara il terreno sia per l'uso consapevole della tecnologia sia per comprendere gli aspetti logici e la struttura profonda delle attività che si svolgono.

Il pensiero computazionale e il coding stanno rivoluzionando i metodi d'insegnamento e di apprendimento già a partire dalla scuola dell'infanzia: si tratta di una nuova e difficile sfida per i docenti, che non devono farsi trovare impreparati, dovendo quotidianamente rapportarsi ai nuovi "studenti nativi digitali".

### 1.1 Pensiero Computazionale - Jeannette Wing

Il termine pensiero computazionale è un stato portato all'attenzione della comunità scientifica in un articolo del 2006 scritto dalla scienziata informatica Jeannette Wing .

Wing si è laureata in ingegneria elettrica e informatica presso il MIT nel giugno 1979; in un articolo breve, un "viewpoint", la Wing propone il pensiero computazionale come una competenza originale e tipica dell'informatica, utile a tutti. Questo ha generato un vivace dibattito internazionale e la rimessa in discussione del curriculum della scuola. Non succede spesso che si rimetta in discussione l'impianto del curriculum scolastico; il pensiero computazionale è riuscito non solo a porre il problema, ma anche ad ottenerne l'introduzione nel curriculum a partire dalla scuola primaria in Inghilterra, Australia, Francia, Polonia e Finlandia.

Il pensiero computazionale fornisce un quadro potente per studiare l'informatica. La sua applicazione, però, va ben al di là della materia stessa ma, riguarda il processo di riconoscimento di aspetti della computazione nel mondo che ci circonda, e dell'applicazione di strumenti e tecniche informatiche per capire e ragionare su sistemi e processi naturali, sociali e artificiali. La locuzione "pensiero computazionale" è stata usata per primo da Seymour Papert.

La prof.ssa Jeannette Wing ha reso popolare l'idea sostenendo che il pensiero computazionale dovrebbe far parte delle competenze di tutti i nuovi studenti universitari.

Esistono un certo numero di tecniche utilizzate per dimostrare e valutare il pensiero computazionale che sono l'equivalente "informatico" dei "metodi scientifici". Questi sono gli strumenti con cui il pensiero computazionale viene reso operativo in aula, sul posto di lavoro e a casa:

1. Riflettere,
2. Programmare (coding),
3. Progettare,
4. Analizzare,
5. Applicare.

Il pensiero computazionale si basa sul potere e sui limiti dei processi di elaborazione, siano essi eseguiti da un essere umano o da un macchina. I metodi ed i modelli computazionali ci danno il coraggio di risolvere problemi e sistemi di progettazione che nessuno di noi sarebbe stato in grado di affrontare da solo.

Il pensiero computazionale rappresenta una competenza fondamentale per tutti, non solo per gli informatici.

Come per la lettura, la scrittura e l'aritmetica, dovremmo aggiungere il pensiero computazionale pensando alla capacità di analisi di ogni bambino. Il pensiero computazionale consiste nel risolvere i problemi, progettare sistemi e comprendere il comportamento umano, attingendo dai principi fondamentali



del mondo dell'informatica comprendendo una serie di strumenti mentali che ne riflettano la vastità.

Consente di riformulare un problema apparentemente difficile in uno che sappiamo come risolvere, ricorrendo anche ad una sua riduzione, inclusione, trasformazione o simulazione. Pensiero computazionale significa pensare in modo ricorsivo. E' un'elaborazione parallela; si tratta di riconoscere sia le virtù che i pericoli dell'implementazione dei linguaggi di programmazione (aliasing analysis , in inglese), quindi ponendo attenzione nel dare a qualcuno o a qualcosa di più che un nome.

Si tratta di giudicare un programma non solo per la correttezza e l'efficienza ma, anche per l'estetica e il design di un sistema che viene ottimizzato per fornirgli semplicità ed eleganza.

Il pensiero computazionale impiega l'astrazione e la decomposizione quando deve approcciare ad una funzione complessa e pesante o per progettare un sistema grande e complicato. In questo modo c'è una separazione delle competenze.

L'informatica intrinsecamente attinge al pensiero matematico, dato che, come tutte le scienze, le sue basi formali poggiano sulla matematica.

Attinge anche dal pensiero ingegneristico costruendo sistemi che interagiscano con il mondo reale. Non saranno solamente il software e gli artefatti hardware che produciamo ad essere fisicamente presenti ovunque toccando la nostra vita tutti i giorni ma, saranno i concetti computazionali quelli che, invece, useremo per affrontare e risolvere i problemi, per gestire la nostra vita quotidiana, per comunicare ed interagire con altre persone.

*” Molte persone equiparano l'informatica con la programmazione del computer. Alcuni genitori vedono solo una stretta gamma di opportunità di lavoro per i loro figli che si laureeranno in informatica. In tanti pensano che la ricerca di base in informatica sia compiuta e che bisogna concentrarsi solo sulla parte ingegneristica. Il computer è quasi sempre visto come uno strumento per insegnare, un ausilio didattico nelle mani del docente, piuttosto che un ambiente a disposizione di chi apprende. Il pensiero computazionale rappresenta una grande visione per guidare, nell'ambito della computer science, gli educatori scientifici, i ricercatori ed i professionisti per cambiare l'immagine che la società ha di questo settore. Piuttosto che l'informatica come materia o il computer come strumento per insegnare, occorre pensare all'informatica come comunicazione, linguaggio, capacità di modellare sistemi e situazioni complesse con strumenti che sono sì matematici, ma in un'accezione molto più ampia, legati alle capacità di base di ragionare, di risolvere problemi di qualsiasi tipo. In particolare, c'è la necessità di raggiungere il mondo delle scuole superiori, tra cui insegnanti, genitori e studenti, inviandogli due messaggi principali:*

- *I problemi scientifici intellettualmente stimolanti e coinvolgenti hanno bisogno di essere capiti e risolti.*
- *L'area di un problema e il dominio delle sue soluzioni sono limitate solo dalla nostra curiosità e creatività.*

*La comunicazione multimediale, e interattiva, la smisurata capacità di memorizzazione dei calcolatori consentirebbero ancora di più di "conoscere e apprendere la saggezza e gli insegnamenti di un maestro senza stargli accanto"[5]. Chi si laurea in Informatica può fare qualsiasi cosa. Così come chi si laurea in Lettere o Matematica può percorrere una moltitudine di differenti carriere. Idem per la computer science. Chi si laurea in informatica può proseguire una carriera in medicina, diritto, economia, politica, qualsiasi tipo di altro ambito scientifico o di ingegneria e anche nell'ambito dell'arte. I professori di informatica dovrebbero insegnare un corso per matricole denominato "I modi di pensare di uno scienziato informatico", rendendolo disponibile per tutti, non solo per i laureandi in informatica. Si dovrebbero coinvolgere anche gli studenti delle superiori allo studio dei metodi e dei modelli computazionali. Piuttosto che lamentarsi del calo di interesse per l'informatica o per il declino dei finanziamenti per la ricerca in informatica, dovremmo cercare di ispirare l'interesse del pubblico ad avventurarsi intellettualmente in questo campo. Perciò dobbiamo diffondere la gioia, i timori e la potenza dell'informatica, con l'obiettivo di rendere popolare il pensiero computazionale." Queste le parole di Jeannette Wing [2] direttrice del Data Science Institute presso la Columbia University, dove è anche professoressa di informatica.*

## 1.2 Seymour Papert

Il concetto di *Pensiero Computazionale* è stato introdotto per la prima volta dal matematico, filosofo, pedagogista e informatico statunitense nonché padre del *Costruzionismo* Seymour Papert nel 1980.

Seymour Papert, nato a Pretoria il 29 febbraio 1928, è stato un matematico, informatico e pedagogista sudafricano naturalizzato statunitense. Il lavoro di Papert inizia negli anni '60 presso i laboratori MIT di Boston come matematico e informatico ma, si interessa anche agli studi di Jean Piaget, psicologo e pedagogista. Piaget è considerato il fondatore dell'epistemologia genetica, ovvero dello studio sperimentale delle strutture e dei processi cognitivi legati alla costruzione della conoscenza nel corso dello sviluppo; egli si dedicò molto anche alla psicologia della crescita.

Piaget dimostrò innanzitutto l'esistenza di una differenza qualitativa tra le modalità di pensiero del bambino e quelle dell'adulto e, successivamente,

che il concetto di capacità cognitiva, e quindi di intelligenza, è strettamente legato alla capacità di adattamento all'ambiente sociale e fisico. Lo sviluppo del pensiero e della riflessione di Papert è segnato da due grandi incontri intellettuali e personali, quello con Jean Piaget, che avviene all'Università di Ginevra, dove lavora tra il 1958 e il 1963 e quello con Marvin Minsky, con il quale incessantemente lavora al MIT.

Papert sostiene che esistano due tipi di apprendimento che influenzano l'approccio ai problemi ed il modo di lavorare con il computer. Distingue fra coloro che hanno l'abitudine di pianificare il lavoro, i (planners) e coloro che invece procedono per tentativi ed errori (bricoleurs). Il modo di apprendere dei bricoleur è molto simile a quello dei bambini ed è costituito da un avvicinamento graduale al problema e di conseguenza alla sua soluzione.

Marvin Minsky, collaboratore di Papert, nel laboratorio di Intelligenza Artificiale di Cambridge enuncia il cosiddetto principio di Papert: "alcuni fra gli stadi più cruciali dello sviluppo mentale sono basati non sulla semplice acquisizione di nuove abilità, bensì sull'acquisizione di nuovi metodi per poter utilizzare ciò che già si conosce[...] poiché una mente non può crescere molto se si limita soltanto ad accumulare conoscenze, ma deve inventarsi anche dei modi per poter sfruttare al meglio le conoscenze già possedute" [5]. Papert sostiene che i principi del bricolage siano utilizzare le cose di cui si dispone, improvvisare, adattarsi.

La collaborazione con Piaget è quella a cui si deve l'interesse di Papert per lo sviluppo cognitivo dei bambini. Il suo lavoro con Piaget si focalizza sul tentativo di utilizzare una modellizzazione matematica delle modalità dello sviluppo cognitivo dei bambini. L'obiettivo di Piaget era capire come i bambini costruiscono la conoscenza e le strutture del pensiero.

### 1.2.1 Il Costruzionismo

Papert sostiene che il processo di apprendimento è un procedimento di costruzione di rappresentazioni più o meno corrette e funzionali del mondo con cui si interagisce. L'essere umano, a prescindere dall'età, ha bisogno di avere a disposizione materiali concreti affinché la conoscenza acquisita sia tanto più vicina alla realtà. Papert parte dall'osservazione di attività di alcune civiltà africane in cui i bambini costruivano case in scala o manufatti in giunco. Secondo Papert, la mente ha bisogno di materiali da costruzione appropriati, esattamente come un costruttore.

Il prodotto concreto può essere mostrato, discusso, esaminato, sondato e ammirato.

La lentezza dello sviluppo di un particolare concetto da parte del bambino non è dovuta alla maggiore complessità o formalità, ma alla povertà della cultura di quei materiali che renderebbero il concetto semplice e concreto.

Il bambino apprende così con l'aiuto di artefatti cognitivi. In particolare, Papert sostiene l'uso del computer come supporto all'istruzione e ambiente d'apprendimento che aiuta a costruirsi nuove idee. Il computer viene così usato come macchina per simulare. Partendo da Piaget, Papert fonde il Costruttivismo con le nuove tecnologie e con gli strumenti informatici, dando vita al **Costruzionismo**.

Il Costruzionismo ha anche la connotazione di "set da costruzioni", dove il termine set è da prendersi in senso letterale, come il set dei Lego. La costruzione di un concetto che avviene nella testa delle persone spesso si verifica in modo particolarmente felice quando è supportata dalla realizzazione di qualcosa di molto più concreto.

[...] *Il costruzionismo assegna una particolare importanza al ruolo delle costruzioni reali a supporto di quelle presenti nella mente*[...][20].

Il *Costruzionismo* di Papert posa sull'assunto che i bambini faranno meglio a scoprire da sé le specifiche conoscenze di cui hanno bisogno e questo li aiuterà ad acquisire altro sapere; si basa proprio sulla capacità di costruire modelli cognitivi funzionanti e di riutilizzarli in contesti simili.

Papert, per ovviare alla mancanza di una parola che indichi l'arte di apprendere coniò il termine **matetica**. La matetica è per l'apprendimento, ciò che l'euristica è per il problem solving.

L'uso di un computer come ambiente per costruire il proprio sapere offre all'alunno creatività e responsabilità permettendogli di sviluppare una serie di concetti e venendo a contatto con le proprie risorse intellettuali, mettendole in pratica in modo spontaneo ed intuitivo.

Questo è quello che effettivamente fa un programmatore. La programmazione è quell'attività che parte da un problema per arrivare alla soluzione.

Il ruolo dell'informatico è quello di analizzare, trovare un algoritmo di risoluzione e verificare se questo è corretto attraverso sistemi di debug.

Inizialmente tutto questo processo era racchiuso nel termine **programmazione** ma, successivamente ne è stata fatta una distinzione tra **Programming** e **Coding**.

### 1.2.2 Il linguaggio LOGO

Papert sostiene l'uso del computer come supporto all'istruzione e ambiente d'apprendimento che aiuta a costruirsi nuove idee.

Papert capisce che semplificare l'uso di istruzioni informatiche per potersi concentrare sulla costruzione di un algoritmo che risolva un problema demandando alla macchina la difficoltà relativa alla sintassi.

Egli ha ideato il linguaggio di programmazione LOGO, il primo e più rilevante tentativo di dare ai bambini e ai ragazzi la possibilità di comprendere e padroneggiare le tecnologie della programmazione e di utilizzarle come strumento per l'apprendimento.

L'ambiente creato da Papert ha la preziosa caratteristica per l'apprendimento, di non fare differenza fra istruzioni messe a disposizione dal linguaggio stesso e istruzioni personali create dall'utente. LOGO consente di costruire e ricostruire procedimenti cognitivi per individuare eventuali inconvenienti permettendo, al bambino di imparare a crearsi un ragionamento, piuttosto che accettarlo o scartarlo in blocco come giusto o sbagliato.

LOGO a differenza di tutti i linguaggi di programmazione esistenti è basato fortemente sulla grafica rendendo l'ambiente adatto anche ai bambini della scuola primaria.

Proprio questa caratteristica lo ha reso il linguaggio più famoso nell'ambito dell'educazione scolastica. La programmazione con LOGO non è mai fine a se stessa ma sempre pensata in relazione ad un dato progetto attinente alle discipline più svariate. LOGO è un linguaggio di programmazione formalmente molto rigoroso ma, allo stesso tempo semplice e visuale che consente con poche istruzioni di descrivere un algoritmo, orientato alla gestione delle liste e alla grafica della tartaruga, comprensibile ed utilizzabile anche da bambini delle scuole elementari.

Per comprendere il ruolo delle tecnologie nei processi di insegnamento e apprendimento, occorre riflettere non soltanto sulla tecnologia ma, anche sui ruoli dell'insegnante e dell'allievo. LOGO è infatti uno strumento che consente ai bambini di utilizzare il computer per ottenere rapidamente, utilizzando principi matematici e logici rigorosi, risultati concreti. In quest'ambiente il docente si trasforma in animatore della comunità, promotore di attività in cui i bambini progettano e imparano esplicitando e discutendo teorie sul mondo con cui interagiscono.

La "geometria della tartaruga" è un'area che Papert e i suoi colleghi hanno creato come area formale della matematica più significativa per i bambini. Nella geometria della tartaruga il computer ha un uso completamente diverso rispetto al tradizionale uso nella didattica. Viene utilizzato come mezzo espressivo. Invece di porsi il problema su "come insegnare la matematica scolastica", si pone su come ricostruirla o più in generale come ricostruire la conoscenza in modo tale che non sia necessario un grande sforzo per insegnare. La geometria della tartaruga si basa su tre principi:

- Principio di continuità: il primo passo per comprendere un concetto è quello di integrarlo con conoscenze pregresse;
- Principio di potenza: l'ambiente di apprendimento deve consentire a chi apprende di capire concetti carichi di significato, che non avrebbe mai pensato prima;
- Principio di risonanza culturale: ciò che viene appreso deve avere senso all'interno dell'ambiente sociale nel quale ci troviamo.

Secondo Papert, i bambini dovrebbero essere "indipendenti, responsabili del proprio apprendimento". Scrive Papert in *Mindstorms*[1]:

*Ricorro a Jean Piaget per il modello secondo il quale i bambini sono essi stessi costruttori delle loro strutture intellettuali. Sembra che i bambini abbiano un dono innato per imparare, poiché molto tempo prima di andare a scuola accumulano una gran quantità di conoscenze mediante un processo che io definisco «apprendimento piagetiano».*

Attraverso il coding, quindi, il bambino partecipa alla creazione e all'implementazione di "artefatti cognitivi" che gli consentono di imparare passo passo a risolvere problemi sempre più complessi riducendoli a tanti piccoli problemi semplici di cui conosce la soluzione.

Papert, infatti, sostiene che l'uso del computer per la creazione degli artefatti cognitivi aiuti anche la costruzione di nuove idee. Il computer, così, non viene usato come mero esecutore materiale di programmi già realizzati, quindi passivamente, ma viene utilizzato per decodificare ciò che il bambino ha codificato.

### 1.2.3 Il Micromondo

Nasce il concetto di **Micromondo**, ossia un ambiente di apprendimento appositamente realizzato per esplorare ed imparare una data disciplina e materia.

Con il termine micromondo si fa riferimento a una riproduzione del comportamento di un sistema reale, ad una simulazione digitale effettuata sul computer. Sono piccoli universi, realtà delimitate, luoghi "sicuri" da esplorare in cui è possibile creare applicazioni sotto forma di percorsi di apprendimento, animazioni, simulazioni geometriche ecc.[20]

L'approccio alla programmazione in LOGO, in particolare, consiste nel provare, testare, ricostruire quello che non va e riprovarlo. In tale contesto l'elemento dell'errore non deve essere più visto come qualcosa di bloccante per l'apprendimento ma, come parte costitutiva del processo di apprendimento e

di costruzione della conoscenza; un'importante opportunità per una migliore comprensione dei problemi. La ricerca e l'individuazione dell'errore, chiamato debugging, attraverso il quale comprendere e risolvere quanto c'è di sbagliato stabilisce con lui un rapporto, in un certo senso, amichevole.

In particolare, l'elemento dell'*errore* diventa la base di partenza per la ricerca e l'individuazione di esso grazie al quale poter comprendere e modificare il proprio ragionamento. Mettere in evidenza l'errore come un aspetto negativo produce energia ed interesse nel bambino, che ha bisogno di non venire rimproverato per l'errore commesso, ma va piuttosto messo nella condizione di controllare l'errore. Gli errori ci aiutano perché ci guidano a studiare ciò che è accaduto, capire cosa non funzionava e, attraverso la comprensione, sistemare le cose.

Un ambiente scolastico costruttivo è un ambiente nel quale l'insegnante apprende insieme al bambino. L'alunno attraverso l'errore può valutare da solo i propri progressi. A livello operativo l'impatto è che gli studenti vorrebbero affrontare il problema tutto in una volta ma, la scomposizione del problema permette di facilitarne la risoluzione, l'identificazione e la risoluzione di eventuali errori.

Il lavoro di Papert ha ispirato generazioni di educatori e ricercatori in tutto il mondo. Ha ricevuto numerosi premi, tra cui una borsa Guggenheim nel 1980, una borsa Marconi International nel 1981 e lo Smithsonian Award da Computerworld nel 1997. Nel 2001, Newsweek lo ha nominato "uno dei 10 principali innovatori della nazione nel campo dell'istruzione".

Morto a 88 anni, il 31 luglio 2016, nella sua casa a East Blue Hill, nel Maine Seymour viene ricordato così da uno speciale [6] <sup>1</sup>

*"I bambini adorano costruire oggetti. Così mi dissi: scegliamo un set di costruzioni e aggiungiamo tutto quello che serve per creare dei modelli ciberneticici. I bambini dovranno essere in grado di costruire una tartaruga dotata di motori e sensori e avere il modo di scrivere programmi logo per guidarla [...] l'unico limite deve essere quello della loro immaginazione e delle loro capacità tecniche. Nel caso dei primi esperimenti condotti, i motori e i sensori dovettero essere collegati a un computer tramite un'interfaccia. In tempi più recenti siamo riusciti a costruire computer abbastanza piccoli da poter essere inseriti nei modelli stessi. La differenza è sostanziale; ora l'intelligenza si trova all'interno del modello, non in un computer esterno. Inoltre i modelli possono essere autonomi. Possono muoversi a piacimento senza un cordone ombelicale. Tutto insomma appare più reale."*

---

<sup>1</sup><https://www.giuntiscuola.it/lavitascolastica/magazine/news/eventi/i-bambini-adorano-costruire-oggetti-ciao-a-seymour-papert/>

Anche sul sito del MIT appare uno scritto in memoria di Seymour Papert [4]<sup>2</sup> nel quale viene raccontata e descritta tutta la sua vita.

*”Con una mente di straordinaria gamma e creatività, Seymour Papert ha contribuito a rivoluzionare almeno tre campi, dallo studio di come i bambini hanno un senso del mondo, allo sviluppo dell’intelligenza artificiale, al ricco incrocio di tecnologia e apprendimento”,* afferma il Presidente del MIT L. Rafael Reif. *”Il timbro che ha lasciato sul MIT è profondo. Oggi, mentre il MIT continua ad espandere la sua portata e approfondire il suo lavoro nell’apprendimento digitale, sono particolarmente grato per la visione rivoluzionaria di Seymour e speriamo di sviluppare le sue idee per aprire le porte agli studenti di tutte le età, in tutto il mondo.”*

*”Seymour parlava spesso poeticamente, a volte in enigmi, come la sua famosa frase,*

*”Non puoi pensare di pensare senza pensare a qualcosa”.”*

### 1.3 Mitchel Resnick - MIT Lifelong Kindergarten

Michael Resnick, nato negli Stati Uniti nel 1956 si è laureato in Fisica nel 1978 presso l’Università di Princeton.

Attualmente si occupa di ricerca nel campo dell’apprendimento presso il MIT Media Lab, dove sviluppa nuove tecnologie e attività per impegnare persone di qualunque età ma, soprattutto i bambini, in esperienze creative di apprendimento.

Spesso si dice che i giovani di oggi siano ”nativi digitali” ma, secondo Mitchel Resnick questo termine non li descrive nel modo corretto. I giovani hanno familiarità e dimestichezza nel navigare in Internet, chattare, mandare messaggi e giocare ma questo non basta per renderli fluenti nell’utilizzo delle nuove tecnologie in quanto non riescono a creare e ad esprimersi utilizzandole.

*E’ come se riuscissero a leggere senza però saper scrivere*[7].

Uno degli obbiettivi fondamentali per Resnick è capire come poter aiutare i giovani a diventare fluenti in modo che possano scrivere con le nuove tecnologie. Questo significa che loro devono essere in grado di scrivere i propri programmi per il computer, o programmare.

Negli anni dell’uscita di Scratch molte più persone si sono rese conto dell’importanza di programmare, infatti sono nate centinaia di nuove organizzazioni e siti web per aiutare i giovani ad imparare come ad esempio Codecademy, CoderDojo, GirlswhoCode, blackgirlsCode, ecc. Anche a livello globale, ad

---

<sup>2</sup><https://www.media.mit.edu/posts/in-memory-seymour-papert/>



esempio il sindaco di New York Michael Bloomberg espresse il desiderio di imparare a programmare, come proposito per il 2012; qualche mese dopo l'Estonia decise che tutti i bambini in prima elementare avrebbero dovuto imparare a programmare e questo scatenò un dibattito negli Stati Uniti sulla necessità di tutti i bambini di imparare a programmare.

Quando si parla di programmare, molte persone pensano sia qualcosa che solo un piccolo gruppo di individui con abilità matematiche ed una formazione prettamente informatica possa fare ma, l'obbiettivo di Resnick è proprio quello di avvicinare la programmazione a tutti.

*"Se si impara a leggere si può anche leggere per imparare e la stessa cosa vale per la programmazione, se si impara a programmare, allora si potrà anche programmare per imparare"[8]*

### 1.3.1 Scratch

Mitchel Resnick, si può affermare, sia il successore di Papert e, proprio come per il suo mentore, l'attività di coding e, in generale, la realizzazione di opere cognitive per mezzo di questa è di fondamentale importanza all'interno dell'istruzione dei bambini e delle generazioni future.

Per perseguire gli obiettivi di diffusione del pensiero computazionale, del coding a scuola e, in generale, per lo studio del cosiddetto Creative Learning, Resnick ha istituito il team Lifelong Kindergarten fondando le proprie radici sui principi promulgati dal maestro Papert del Costruzionismo, realizzando un framework per l'insegnamento del pensiero computazionale e la valutazione dell'apprendimento conseguito.

Papert, col suo Costruzionismo, sosteneva che alla base di un migliore e più naturale apprendimento la mente umana avesse bisogno di creare artefatti cognitivi, ovvero rappresentazioni reali del mondo con cui interagire. Per sperimentare sul campo le teorie, Papert, ideò più di cinquanta anni fa, il linguaggio LOGO.

Come il suo predecessore, Resnick per promuovere la spirale dell'apprendimento creativo cominciò a lavorare ad un piccolo robot programmabile e sviluppò così il famoso linguaggio di programmazione a blocchi **Scratch**.

*"Quando qualcuno impara a programmare con Scratch impara allo stesso tempo importanti strategie per risolvere problemi, creare progetti e comunicare le proprie idee".*

L'idea di Scratch è che attraverso il gioco s'impari la logica e ci si appropri del pensiero computazionale inventando, creando, montando e smontando il proprio mondo interiore. Scratch è stato inventato e sperimentato al MIT

all'inizio degli anni 2000 ed è, per certi versi, l'erede di LOGO con alcune differenze chiave.

LOGO era stato progettato pensando alla matematica mentre Scratch è stato pensato come strumento per la libera espressione dei più piccoli. Scratch utilizza un linguaggio di programmazione di tipo grafico e visuale, a blocchi, come i blocchetti delle costruzioni Lego.

Scratch prevede un approccio orientato agli oggetti (denominati Sprite) ed un linguaggio di programmazione che consente di elaborare storie interattive, giochi, animazioni, arte e musica. Inoltre permette di condividere i progetti con altri utenti del web.

L'idea di questo linguaggio è che anche i bambini o le persone inesperte in linguaggi di programmazione possano imparare importanti concetti di calcolo matematico, a ragionare in modo sistematico, a pensare in modo creativo e a lavorare in gruppo.

### 1.3.2 Le 4 "P"

Scratch prende il nome da "scratching", una tecnica utilizzata dai disk jockey per mixare clip musicali e produrre diversi effetti sonori manipolando "graffiando" i dischi in vinile. Nel nome – Scratch – è descritto il suo funzionamento, poiché in effetti si possono mescolare diversi media (inclusi grafici, suoni e altri programmi) in modo creativo.

Per creare un programma, infatti, basterà trascinare dei blocchi di codice, preimpostati e contraddistinti dai diversi colori, all'interno dell'area di coding e "incastrarli" insieme seguendo un ordine logico, in modo da far muovere dei personaggi, farli interagire, animare una storia, creare un game, o qualsiasi altra animazione. Al MIT hanno identificato con quattro P l'evoluzione di Scratch.

#### **Projects, Peers, Passion, Play**

**Progettare:** fornire ai bambini l'opportunità di lavorare su progetti significativi (non solo su attività in stile puzzle), in modo tale da sperimentare il processo di trasformazione dall'idea iniziale fino alla creazione di qualcosa che possa essere condiviso con gli altri.

**Condividere tra pari:** incoraggiare la collaborazione e la condivisione e aiutare i bambini a imparare a costruire qualcosa partendo dal lavoro di qualcun altro. Programmare non dovrebbe essere un'attività solitaria.

**Appassionarsi:** permettere ai bambini di lavorare su progetti in linea con i loro interessi. Il loro impegno sarà maggiore e più duraturo e il loro apprendimento sarà costante.

**Giocare:** incoraggiare i bambini a sperimentare attraverso il gioco. Provare nuove cose, prendersi dei rischi, testare i limiti, imparare dagli errori.

Scratch è completamente open source e permette dunque la condivisione e il riutilizzo di codice con altri e di altri. Lo slogan della comunità online di Scratch recita "Immagina, Programma, Condividi"<sup>3</sup>, sottolineando l'importanza della condivisione e degli aspetti sociali della creatività nella filosofia alla base di Scratch.

Il Team Scratch del MIT assicura che questa comunità mantenga un ambiente amichevole e rispettoso per le persone di ogni età, razza, etnia, religione, orientamento sessuale e identità di genere. A tutti i membri viene richiesto di fornire feedback in modo costruttivo e di segnalare qualsiasi contenuto che non segua le linee guida della community. Per garantire ulteriormente questa comunità, il team di Scratch gestisce l'attività del sito e risponde ai report su base giornaliera. Esiste anche una comunità online per educatori chiamata ScratchEd. ScratchEd è stato sviluppato e supportato dalla Harvard Graduate School of Education. In questa comunità, gli educatori di Scratch condividono storie, scambiano risorse, fanno domande e trovano persone.

Un "Scratch Day" viene dichiarato ogni anno a maggio. I membri della comunità sono incoraggiati a ospitare un evento in questa giornata, grande o piccolo, che celebra Scratch. Questi eventi si svolgono in tutto il mondo e un elenco è disponibile sul sito Web di Scratch Day.

Un giorno Resnick, rendendosi conto che il giorno successivo sarebbe stata la festa della mamma, non avendole comprato un regalo cominciò a pensare e gli venne l'idea di creare un biglietto di auguri interattivo utilizzando Scratch, il software che stava sviluppando con il suo gruppo di ricerca al MIT. Prima di creare il suo biglietto, però, decise di dare un'occhiata al sito web di Scratch e, scrivendo nella barra di ricerca "MOTHER'S DAY", con estrema sorpresa, trovò una lista di decine e decine di biglietti di auguri per la festa della mamma.

Dando un'occhiata a questi biglietti, ne trovò con musica, immagini, racconti "mouse effect" e, rimanendone estremamente stupito decise di regalare a sua mamma il link a decine di questi progetti.

Questa "scoperta" rese Resnick veramente felice in quanto i ragazzi stavano usando Scratch proprio con lo scopo per cui era stato creato. Questi ragazzi erano diventati fluenti, ovvero erano riusciti ad esprimere loro stessi e le loro idee.

## 1.4 Coding

*Coding* è un termine inglese al quale corrisponde in italiano la parola *programmazione*. Parliamo di programmazione informatica ovviamente ma, non nel senso più tradizionale dell'espressione.

---

<sup>3</sup><https://scratch.mit.edu/studios/2050636/>

”Coding” va ben oltre la sua traduzione letterale in ”codifica” o ”programmazione”, indica «l’uso di strumenti e metodi intuitivi di programmazione per favorire lo sviluppo del pensiero computazionale». Il coding a scuola è una scoperta, se così può essere definita, recente.

Parliamo di un metodo che mette la programmazione al centro di un percorso dove l’apprendimento, già a partire dai primi anni di vita, percorre strade nuove ed è al centro di un progetto più ampio che abbatte le barriere dell’informatica e stimola un approccio mirato alla risoluzione dei problemi. Bambini e ragazzi con il coding sviluppano il pensiero computazionale e l’attitudine a risolvere problemi più o meno complessi. Non imparano solo a programmare ma programmano per apprendere.

La programmazione visuale a blocchi, principale strumento di coding permette di esprimere un procedimento come concatenazione di blocchi colorati che rappresentano istruzioni elementari da eseguire in sequenza. La descrizione è talmente rigorosa da poterne affidare l’esecuzione a un esecutore ”ideale”, privo di intelligenza come robot o computer. La programmazione è un atto creativo assoluto, perché sono solo automatismi quelli separano un programma dalla sua esecuzione.

Il coding si presta a eccezionali applicazioni pedagogiche in quanto presuppone una comprensione profonda del procedimento che si intende affidare a una macchina. Che siano bimbi o studenti, che frequentino ancora la scuola dell’infanzia o l’università, i campi di applicazione sono innumerevoli.

L’efficacia del coding nello sviluppo dei ragazzi è così rilevante che la Commissione Europea dal 2013 ha lanciato una campagna di sensibilizzazione e alfabetizzazione funzionale denominata ”Europe Code Week” punto di riferimento in Italia per docenti, studenti e genitori attraverso la piattaforma code-mooc.org. Un impegno e delle esperienze che stanno dando grandi risultati se si considera che le scuole italiane sono state protagoniste del 45 per cento delle attività organizzate durante l’ultima edizione di Europe Code Week.

Il coding utilizzato nella pratica didattica dagli insegnanti non va confuso con l’uso di tecnologie digitali, con l’educazione civica digitale o con l’insegnamento dell’informatica. «A scuola non esiste l’ora di coding», precisa Bogliolo, docente di Sistemi di elaborazione delle Informazioni presso il Dipartimento di Scienze pure e applicate all’Università di Urbino e coordinatore a livello comunitario della Europe Code Week, perché il coding non è una disciplina a sé. Il coding è un metodo, uno strumento nelle mani degli insegnanti che possono decidere di applicarlo alla propria didattica per innescare nuove dinamiche all’interno della classe, favorire il lavoro in gruppo, lanciare sfide, fare squadra, coinvolgere tutti.

Il coding praticato ora nella scuola non è programmazione, ma l’uso didattico di strumenti di programmazione. «Il coding si pratica, mentre l’informatica,

e con essa la programmazione, andrebbero insegnate.»[3]<sup>4</sup>

Il coding diventa una palestra di logica per tutti. Il coding a scuola sta gradualmente ritagliandosi uno spazio nella didattica non solo perché educa bambini e ragazzi al pensiero creativo, ma anche per un suo risvolto pratico. Perché ha a che fare con la nostra quotidianità, con molti oggetti che utilizziamo abitualmente e che riteniamo ormai indispensabili. Attraverso una serie di giochi ed esercizi interattivi, basati su un'interfaccia visuale, il bambino può determinare le azioni di uno o più personaggi spostando blocchi o oggetti grafici su un monitor. A ciascun blocco corrisponde un'azione, una linea di codice che non ha bisogno quindi di essere digitato.

Il Regno Unito è uno di quei paesi europei che prevede l'informatica all'interno dei suoi curriculum scolastici e, sin dal 2014 ha introdotto il computing. Anche l'Italia si sta muovendo in merito alle attività di pensiero computazionale e coding all'interno della realtà scolastica.

Il 16 Luglio 2015 è entrata in vigore la legge della "Buona Scuola". Uno dei punti salienti della riforma riguarda proprio il pensiero computazionale, il coding e le cosiddette competenze digitali. Lo scopo della riforma è dare spazio all'educazione, ai corretti stili di vita, alla cittadinanza attiva, all'educazione ambientale e guardare al domani attraverso lo sviluppo delle competenze digitali degli studenti.

Il MIUR nell'ambito del programma della Buona Scuola ha avviato l'iniziativa Programma il Futuro con lo scopo di portare nelle scuole i concetti base dell'informatica utilizzando strumenti semplici e divertenti.

Programma il Futuro si occupa di fornire supporto agli insegnanti tramite percorsi strutturati, video lezioni sui concetti di base dell'informatica, forum di discussioni, momenti di formazione e di creare delle attività di coding. Non è necessaria nessuna conoscenza tecnica o scientifica per prendere parte a questa iniziativa. La partecipazione prevede due percorsi: uno base, con la partecipazione all' "Ora del Codice" e l'altro avanzato per seguire percorsi più strutturati in base alla fascia d'età. La modalità base di partecipazione, definita L'Ora del Codice, consiste nel far svolgere agli studenti un'ora di avviamento al pensiero computazionale.

Entrambe le modalità possono essere fruite sia in un contesto tecnologico, per le scuole dotate di computer e connessione a Internet, sia in modo tradizionale, per le scuole ancora non supportate tecnologicamente.

*Gli elaboratori nelle scuole devono essere semplici e amichevoli e, al tempo stesso, potenti e utili. [21]*

---

<sup>4</sup><https://www.avvenire.it/agora/pagine/con-il-coding-a-scuola-di-futuro>



# Capitolo 2

## PROGETTO MIA

Il Progetto MIA è un progetto finanziato dalla regione Emilia Romagna. L'obiettivo del progetto MIA, in collaborazione con Criad Coding e FabLab Romagna, è quello di portare il coding ed il pensiero computazionale nelle scuole.

Oggi è sempre più importante incentivare la creatività digitale soprattutto partendo dalle scuole primarie dove il computer è visto solamente come uno strumento utilizzabile dall'insegnante per supportare la didattica.

Se non si dedica abbastanza tempo allo studio del coding, il rischio è quello di avere giovani che nel mondo del lavoro si troveranno in difficoltà, poiché non avranno gli strumenti per essere competitivi e preparati o semplicemente per poter risolvere problemi nel modo migliore possibile.

Il progetto MIA porta nelle scuole il coding stimolando la creatività dei bambini e dei ragazzi, appartenenti a scuole di diverso ordine e grado, con temi di attualità.

L'obiettivo è quello di creare una stazione meteorologica compresa di sensori ed attuatori su dispositivo Raspberry Pi3, con cui gli studenti potranno effettuare misurazioni circa la temperatura, la qualità dell'aria, l'umidità ecc nell'ambiente in cui viviamo.

### 2.1 Caratteristiche principali

Il progetto MIA è un progetto finanziato dalla regione che coinvolge diversi enti.

L'obiettivo specifico del progetto è quello di sostenere i ragazzi di vari ordini scolastici ed i relativi docenti ad un lavoro di gruppo interdisciplinare, finalizzato ad approcciare il mondo tecnologico attraverso l'utilizzo di ambienti di programmazione visuale evoluta (quali Scratch/Snap!) orientati all'utilizzo

ed all'auto-costruzione di misuratori di inquinamento replicabili, in modo da organizzare future misurazioni della qualità dell'aria sul nostro territorio.

L'obiettivo generale del progetto è far maturare la consapevolezza della vicinanza di nuove possibilità tecnologiche e ambienti deputati a farle crescere; far lavorare insieme studenti e docenti di diversi ordini scolastici, promuovendo un'estensione del lavoro di gruppo fra diverse tipologie di scuole.

Responsabilizzare gli studenti di scuole superiori di primo e secondo grado alla formazione degli alunni delle scuole di livello inferiore.

Far maturare la consapevolezza della propria creatività legata allo sviluppo del pensiero computazionale, ricordandoci che Learning to Code, Coding to Learn.

Imparare a costruire codici è un grande aiuto per codificare il metodo di acquisizione di conoscenza. Il coronamento educativo e tecnologicamente significativo è poi il "fare", il making di un Fablab.

Con il presente progetto inoltre, si vogliono promuovere le pari opportunità per le persone con disabilità al fine di sostenerne il pieno inserimento al lavoro attraverso la personalizzazione dei percorsi formativi e orientativi sulle tecnologie STEAM.

STEAM è prospettiva strategica sul territorio cesenate che coniuga al meglio gli asset con le peculiarità del suo ricco tessuto scientifico e d'impresa e, con sfide competitive, rispetto alle quali esso si deve misurare, cercando di differenziarsi nella competizione internazionale.

STEAM è investimento e fonte di ricadute economiche e occupazionali, che vanno al di là dell'impatto immediato in numeri assoluti, poiché si fondano sulla capacità di anticipare e valorizzare l'innovazione e il cambiamento. Questa piattaforma poggia su alcune necessarie premesse di governance, di territorio e reti, di competenze e d'inclusione.

STEAM prende spinta da tre leve strategiche e trasversali: l'innovazione, l'imprenditorialità e l'attrattività.

Quattro le traiettorie prioritarie di sviluppo:

- life science, cluster ad elevato potenziale e di frontiera;
- la manifattura 4.0, futuro del nostro sistema produttivo;
- l'arte, cultura e creatività;
- l'economia sostenibile, per uno sviluppo duraturo e più equilibrato nell'uso delle risorse.



## 2.2 Le fasi di progetto

Il progetto investe nella formazione di base orientata alla creatività che rappresenta la chiave di svolta per il contenimento delle perdite occupazionali. La creatività attraverso l'evoluzione tecnologica rappresenta un investimento per il futuro nella formazione dei ragazzi, garantendo buone prospettive per la loro futura professione.

Lo sviluppo del progetto MIA si suddivide in tre fasi durante le quali i ragazzi potranno conoscere nuovi luoghi, effettuare nuove esperienze ed interagire con i nuovi strumenti messi a disposizione.

Le fasi del progetto prevedono in particolare:

1. Laboratori e incontri di preparazione all'esperienza,
2. Visite guidate e laboratori sul campo,
3. Laboratori e incontri di rielaborazione all'esperienza.

Accompagnare i ragazzi individuati nominativamente dagli Istituti Scolastici del territorio nella fruizione di percorsi di orientamento, formativi e/o laboratoriali, anche in alternanza con il sistema delle imprese, sostiene il processo di transizione dai percorsi scolastici verso il lavoro degli studenti frequentanti il primo ed il secondo ciclo di istruzione.

In altre parole, attivare un lavoro comune ai diversi ordini scolastici è possibile e potenzialmente molto costruttivo se viene fatta una chiara divisione delle competenze e responsabilità fra i vari ordini scolastici.

Molto significativo per la crescita personale è l'attivazione di formazione mirata a formare gli allievi dei livelli scolastici più alti alla stessa formazione dei bimbi di scuola elementare.

Questa attività può inserirsi e perfezionare alcuni percorsi ASL già in atto nelle scuole considerate.

Il progetto prevede come obiettivo finale la progettazione, costruzione, collaudo e messa in attività di alcune periferiche. Periferiche come ad esempio un misuratore di inquinamento atmosferico il quale può essere prodotto a basso prezzo (con microcontrollori).

La produzione, inscatolamento e primo collaudo sarà affidata a FLR. Tali periferiche dovranno essere programmabili utilizzando ambienti Scratch/Snap!.

Azioni:

1. Criad Coding, referenti Tecnopoli, docenti e studenti delle scuole superiori (in particolare ITT) definiscono il progetto base.

2. Criad Coding cura la formazione per gli allievi delle scuole superiori, medie e elementari su ambienti Scratch/Snap: si punta ad attivare il pensiero computazionale e come obiettivo terminale del progetto si arriverà a far funzionare e inserire nel proprio ambiente scolastico e/o casalingo i rilevatori. Gli allievi, seguiti dai docenti interni, raccoglieranno dati dai sensori per poi effettuare delle operazioni opportune. In realtà ciò che è stato chiamato "obiettivo finale" è solo apparente, tangibile perché il vero obiettivo finale è la formazione di allievi e docenti.
3. Laboratori esperienziali a vari livelli: lavoro di gruppo interno al proprio contesto scolastico e in contesti scolastici di diverso ordine e grado, affiancamento dei docenti nei percorsi progettati, verso ragazzi più piccoli avere come riferimento ragazzi più grandi nei percorsi progettati, pianificare percorsi didattici con ragazzi di grado scolastico diverso dal proprio, educare alla responsabilità, rispetto verso l'ambiente e l'auto formazione (self directed learning).

Con il presente progetto inoltre, si vogliono promuovere le pari opportunità per le persone con disabilità, di conseguenza, la strategia sostiene l'accessibilità dei sistemi educativi generali, le misure di accompagnamento individuale e la formazione delle figure professionali del sistema educativo.

L'integrazione delle competenze degli allievi con le nuove tecnologie è uno degli aspetti che contraddistinguono le scelte compiute dall'Italia nel sistema educativo di istruzione e formazione che può vantare, a differenza di altri paesi europei, un'esperienza di lunga data.

Una scuola che "*include le nuove tecnologie*" è una scuola che **pensa** e che **progetta**, a parere dei formatori e degli esperti è una scuola che non si deve muovere nella condizione di emergenza, ma che si deve muovere sul binario del miglioramento organizzativo e tecnologico.

I punti cardine del lavoro con Scratch, secondo il fondatore Mitchel Resnick, sono le 4 P (Project, Passion, Peer, Play). Il fulcro del progetto MIA è Peer, Condivisione.

Elemento fondamentale dello sviluppo tecnologico informatico ma, altrettanto importante come elemento educativo. Un progetto che coinvolge ambienti così differenti, già orientato in partenza alla valorizzazione della creatività che ha concreta e fantastica espressione nelle Casine Enaip, ha senza dubbio l'elemento centrale della condivisione.

Per favorire le pari opportunità e per garantire libero accesso all'esperienza di apprendimento, la rete si rende disponibile ad attivare azioni di sostegno a cura dei tutor dedicati in stretta collaborazione con il coordinatore, con lo scopo di accompagnare l'allievo durante il processo di acquisizione dell'esperienza.

Al fine di valorizzare al massimo le motivazioni e le competenze individuali viene sviluppato un percorso metodologico organizzato per azioni di intervento:

- **AZIONI DI ACCOGLIENZA:** prima accoglienza in ingresso, supporto durante il percorso nelle fasi critiche.
- **AZIONI DI SOSTEGNO ALLIEVI:** attività di affiancamento prevalentemente nei laboratori tecnico professionali, per sviluppare abilità e competenze pratico-manuali e tecnico-professionali.
- **AZIONI DI POTENZIAMENTO CONOSCENZE STEAM:** per rafforzare le abilità di base legate all'area cognitiva, sensoriale-percettiva, emozionale e socio-affettiva.
- **AZIONI DI PREPARAZIONE E ACCOMPAGNAMENTO** all'esperienza per sostenere i ragazzi nel delicato passaggio dalla scuola al contesto organizzativo e produttivo.

Risorse professionali interne ed esterne:

- equipe con figure provenienti dalla rete di progetto: coordinatore del corso, tutor/consulenti con esperienza pluriennale, psicologo, docenti e direzione.

Risorse strumentali:

- mezzi audiovisivi e informatici che facilitano il processo di apprendimento. Gli ambienti sono attrezzati per ospitare allievi diversamente abili.

La rete di partner collabora con docenti di chiara fama dell'Università di Bologna nell'approfondimento delle metodologie di pedagogia speciale.

## 2.3 Strategia di Intervento

Il Progetto MIA, coinvolge diversi enti e, per garantirne la piena riuscita è stata attuata una strategia d'intervento.

Quest'ultima si compone di 4 fasi ognuna con un obiettivo specifico.

- **Fase 1** è la fase più delicata in quanto viene fornita agli studenti una preparazione inerente all'ambiente di programmazione Scratch/Snap!. Il percorso formativo è sviluppato con metodologie didattiche operative e interattive per aumentare la motivazione degli alunni nelle attività proposte, predisponendo un contesto aperto al confronto ed alla condivisione

delle competenze, conoscenze ed esperienze pregresse, propedeutiche all'acquisizione di nuove nozioni, concetti ed abilità attraverso un approccio partecipativo e operativo in linea con la didattica attiva delle discipline STEAM.

L'attivazione di laboratori sull'utilizzo di Scratch/Snap vuole offrire agli allievi delle conoscenze di programmazione base che consentano di imparare importanti concetti di calcolo matematico, con sistematicità e rigore scientifico senza tralasciare lo sviluppo della creatività e dell'acquisizione di competenze trasversali utili al lavoro in gruppo.

Il linguaggio semplice ed intuitivo favorisce lo sviluppo di competenze di programmazione e di costruzione in blocchi anche per gli allievi con meno conoscenze informatiche e matematiche come la scuola primaria.

La metodologia didattica di apprendimento adottata per questo modulo riprenderà i principi di "apprendistato esperienziale" in cui l'esperienza vissuta viene ricondotta a concetti chiave funzionali alla creazione di schemi operativi per la messa in pratica delle conoscenze acquisite in contesti reali e differenti a quello di ideazione, programmazione e realizzazione.

L'acquisizione delle conoscenze base del sistema di programmazione sarà agevolato dagli esperti che predisporranno il contesto all'interazione di gruppo favorendo eterogeneità e interdipendenza positiva tra i pari per l'apprendimento e l'approfondimento teorico e pratico di una nuova conoscenza.

Il laboratorio prevede 15 ore per ciascun gruppo di 20 allievi per un totale di 60 ore.

Questa fase si suddivide a sua volta in tre sotto fasi:

- **Fase 1.A** Gli studenti del "Pascal-Comandini" vengono selezionati in relazione ad attitudine ed interesse dai docenti interni e referenti FLR. Vengono formati con un percorso laboratoriale di preparazione dell'esperienza inerente all'ambiente di programmazione Scratch/Snap!. Inoltre aiuteranno ad assemblare dei misuratori di inquinamento atmosferico, microcontrollori e periferiche ad hoc. Questo tipo di formazione viene tenuta da FLR e Criad Coding.
- **Fase 1.B** I partecipanti della scuola di Viale della Resistenza di Cesena attiveranno percorsi laboratoriali di preparazione delle basi di Scratch/Snap!. Tali corsi avranno scopo introduttivo per imparare a conoscere l'ambiente di programmazione.

Corsi coordinati da Criad Coding e FLR con la collaborazione dei ragazzi formati al "Pascal-Comandini"

- **Fase 1.C** Come per la fase 1.B anche i partecipanti della scuola del terzo Circolo Didattico Cesena verranno coinvolti in un corso sulle basi di Scratch/Snap! per imparare a conoscere l'ambiente di programmazione.
- **Fase 2** visite ai Tecnopoli di Rimini e Forlì- Cesena con una analisi delle possibilità legate alle competenze chimiche e informatiche e una prima visione delle potenzialità di un microcircuito con cui verificare l'inquinamento atmosferico.

Il percorso formativo è sviluppato con metodologie didattiche operative e interattive per aumentare la motivazione degli alunni e collegare le conoscenze acquisite durante gli incontri di preparazione con l'esperienza dei laboratori sul campo.

Le uscite didattiche costituiscono un arricchimento dell'attività formativa e un'occasione di stimolo per lo sviluppo e la formazione della personalità degli allievi che hanno modo di sperimentare in maniera diretta i luoghi della ricerca e dell'innovazione, ampliando le conoscenze della ricerca scientifica e tecnologica.

Approcciare alle discipline STEAM con un approccio pratico sul campo, incoraggia l'allievo a pensare, progettare e realizzare con rigore scientifico uno strumento e validarne l'efficacia, approcciando ad una didattica improntata all'osservazione, screening, sperimentazione e contestualizzazione dell'esperienza sul campo.

L'applicazione pratica dei sensori Mia progettati e realizzati con programmazione Scratch/Snap, localizzati dentro a box costruiti con Laser Cutter e/o completati dalle Casine prodotte dai ragazzi dell'Enaip di Cesena, è sicuramente un'occasione di verifica e valutazione dello strumento precedentemente ideato e realizzato.

La visita guidata presso i tecnopoli sarà occasione di confronto sulle difficoltà riscontrate in fase di ideazione dello strumento.

I docenti nei momenti di ricerca applicativa avranno modo di osservare e verificare le conoscenze pregresse acquisite dagli allievi, motivando al confronto e all'ascolto tra i pari ed esperti per il raggiungimento del risultato.

Il modulo prevede 6 ore per le classi interessate dei diversi ordini e grado per un totale di 24 ore.

- **Fase 3** La terza fase comprende l'attivazione e l'utilizzo dei sensori correlati a momenti di raccolta e di verifica dati. La metodologia didattica di apprendimento adottata per questo modulo riprenderà i principi di "apprendistato esperienziale" in cui l'esperienza vissuta sarà oggetto di riflessione e esplicitazione narrativa, per ricondurla a concetti chiave funzionali, all'acquisizione di competenze informatiche e tecnologiche applicabili ad altre programmazioni di sistema.

Lo schema che verrà riproposto sarà quello che riprende le seguenti fasi:  
esperienza vissuta,

esplicitazione e narrazione,

concettualizzazione e formalizzazione degli schemi e impianti operativi per un successivo ritorno alla messa in pratica.

In questo modulo saranno attivati momenti di raccolta e verifica dati nei vari ambienti di sperimentazione con successiva sistematicità di essi per la rappresentatività e significatività dei dati dell'intera operazione. Questa operazione di rielaborazione sarà oggetto di rimando al pubblico. Il modulo prevede 10 ore per le classi interessate dei diversi ordini e grado per un totale di 40 ore. Come la prima fase anche quest'ultima viene suddivisa in sotto fasi:

- **Fase 3.A** Attivazione con ambiente di programmazione Scratch/Snap! dei sensori MIA. Essi potranno essere assemblati dentro a box costruiti con Laser Cutter e/o completati dalle bellissime Casine prodotte dai ragazzi dell'Enaip di Cesena, coinvolti nel progetto.
  - **Fase 3.B** Vengono attivate momenti di raccolta e verifica dati nei vari ambienti coinvolti e sotto la sovrintendenza dei ragazzi del "Pascal-Comandini" verrà data una sistematicità all'operazione, in modo da aumentare la significatività dei dati raccolti.
- **Fase 4** I ragazzi e docenti partecipano a un momento di rielaborazione dell'esperienza e confronto con la cittadinanza.

## 2.4 Enti coinvolti

Il progetto MIA è un progetto finanziato dalla regione, che coinvolge diversi enti:

- Fondazione EnAIP Forlì-Cesena,
- Fablab Romagna Romagna (FLR) APS,

- Tecnopolo Forlì - Cesena - gestito da Ser.In.Ar. Forlì-Cesena,
- Criad Coding, sezione del CRIAD – Centro di Ricerche e Studi per l’Informatica Applicata alla Didattica gestito da Ser.In.Ar. Forlì-Cesena
- 3 scuole di diverso ordine e grado:
  - Scuola Primaria 3 Circolo Didattico Cesena,
  - Scuola Secondaria di Primo grado Comprensivo SMS Viale della Resistenza - Cesena ,
  - Scuola Secondaria di Secondo grado IIS ”Pascal-Comandini” di Cesena FC.

Ogni ente coinvolto lavora per realizzare una parte del progetto.

### 2.4.1 Fondazione ENAIP Forlì-Cesena

L’Ente Acli per l’istruzione professionale è nato nel 1948, per ordinare le iniziative di formazione professionale che nascevano in maniera coordinata e spontanea in diverse realtà territoriali. Il 5 aprile del 1961 un Decreto del Presidente della Repubblica, sancì il riconoscimento giuridico dell’En.A.I.P nazionale e, già nel 1962 si arrivò alla definizione di un Consiglio di Amministrazione e di una Giunta esecutiva nazionale.

A Forlì, nel giugno del 1965, si inaugurarono i locali della nuova sede in via Campo di Marte, e dagli anni ’70 in poi, sotto la guida di Roberto Ragazzini, l’attività formativa del centro conobbe profonde trasformazioni.

Con l’entrata in campo del Fondo Sociale Europeo l’attività dell’ente ha potuto evolvere, diventando più solida e qualificata.

Il territorio di Cesena è stato il campo di sperimentazione per iniziative rivolte a ragazzi con handicap e immigrati.

Nel 1982 l’En.A.I.P cominciò a promuovere corsi integrativi nel settore informatico per giovani della scuola media e superiore, e diede avvio a tutte le attività formative post-diploma, in collaborazione con numerose aziende del territorio.

Nel 2015 il programma di sviluppo dell’organizzazione ha portato alla trasformazione dell’Ente in Fondazione, un passaggio necessario per la gestione di una realtà sempre più complessa a livello organizzativo, per una più efficace attività di formazione volta allo sviluppo di competenze personali e professionali, oltre che per garantire la tutela di un patrimonio maturato e consolidatosi negli anni.

Ad oggi la Fondazione En.A.I.P ha ampliato la propria offerta a tutti gli ambiti formativi: dalla formazione per minori, ai corsi di avvio al lavoro per giovani e adulti, fino a corsi di alta formazione e attività di inclusione sociale.

La Fondazione non ha scopo di lucro e svolge attività nel settore dell'istruzione, della formazione professionale e dell'assistenza sociale ai fini della promozione morale, culturale e civile di giovani, adulti e di persone socialmente svantaggiate nel quadro di un programma di educazione permanente.

Fondazione EnAIP Forlì-Cesena ha 4 sedi accreditate per la formazione professionale nella provincia, attrezzate per l'erogazione di servizi di orientamento individuale o in piccolo gruppo, azioni formative e didattica laboratoriale.

Sono dotate di aule didattiche attrezzate, laboratori di informatica con PC, collegamento a Internet, stampanti e videoproiettore, aule per lo svolgimento di seminari, incontri con ausili di video proiezione e TV. Aule laboratorio per l'orientamento e il bilancio di competenze degli utenti che si rivolgono ai centri di Formazione.

In ciascuna sede operano le seguenti risorse umane:

- Direttore di sede: con funzione di monitoraggio e controllo delle attività, amministrative, finanziarie, esecutive.

- Un team di funzione, Responsabile della progettazione, Responsabile del Sistema qualità, Responsabile informatico;

- Coordinatore delle operazioni: formatore con funzioni di gestione delle attività che cura i rapporti con la rete territoriale e soggetti finanziatori, i partner, le Scuole, i referenti regionali e dell'organismo intermedio, monitora le risorse economiche assegnate;

- Formatori e tutor: collaborano con il coordinatore responsabile e si occupano di gestire le operazioni di propria pertinenza predisponendo le risorse logistiche e strumentali a disposizione;

- orientatori: impegnati nella erogazione delle misure orientative previste nelle operazioni di pertinenza. Gli orientatori sono anche formatori esperti del mercato del lavoro e nell'erogazione di interventi orientativi individuali e di gruppo;

- risorse professionali specifiche: Fondazione EnAIP Forlì-Cesena ha al suo interno esperti in materia psico educativa, psicologi, che coadiuvano le azioni formative educative erogate agli utenti, giovani e adulti.

Gli ambienti educativo formativi sono aggiornati, ammodernati e adeguati alle nuove esigenze tecnico didattiche, anche nei termini previsti dal DM 139/07 per le competenze di base. Gli spazi formativi comprendono in specifico:

- 3 aule formative didattiche da 22 posti
- 1 aula multimediale da 22 postazioni per la consultazione online
- 1 aula insegnanti per i consigli di classe con biblioteca.
- 1 laboratorio di Fisica/Matematica attrezzato con 22 postazioni individuali.



Nel presente progetto si impegna, in accordo con gli altri partner dell'ambito territoriale interessati, a:

1. condividere con il collegio formatori le linee di sviluppo del progetto MIA,
2. impostare un approccio alle classi coinvolte che parta dalla condivisione delle famiglie e degli allievi,
3. predisporre quanto necessario per la preparazione e realizzazione dell'esperienza laboratoriale.

Quest'ultima avrà la seguente struttura: il referente del Fablab Romagna incontra gli allievi e il consiglio dei formatori e comunica in accordo con i docenti di laboratorio le fasi del progetto che coinvolgono il centro.

Condivide, con il docente prioritariamente di laboratorio che parteciperà con i ragazzi all'esperienza e sarà accompagnato nella fase di preparazione del laboratorio, le fasi delle attività a cui, nella parte laboratoriale, partecipano eventualmente anche i ragazzi dell'iti Pascal Comandini come formatori tra pari.

Successivamente verrà avviato il laboratorio a cui parteciperà il gruppo di allievi in numero di 20 designato in accordo le famiglie. Saranno programmati degli incontri con l'esperto Fablab Romagna per informare le classi. In questo caso per la piena riuscita del progetto è individuato un numero di 20 utenti nella prospettiva di raccordare il mondo della ricerca con il mondo produttivo e della formazione professionale.

Questa strategia sostiene l'accessibilità degli allievi ai sistemi produttivi, le misure di accompagnamento individuale e la formazione delle figure professionali nel sistema della formazione.

Le competenze possono essere spese all'interno degli stages formativi organizzati dal CFP nell'abito del percorso formativo IEFP. In accordo con i partner di progetto nei laboratori di Enaip inoltre saranno realizzati i case-container dello strumento di rilevazione.

Fondazione EnAIP Forlì-Cesena gestisce la governance complessiva del progetto e promuove i risultati attesi. Organizza e gestisce l'intero processo attuativo dei progetti. La governance è svolta in coordinamento con i promotori con i quali condivide in sede di progettazione l'analisi di contesto, le strategie, la pianificazione del progetto e dei relativi laboratori.

In sede di attuazione partecipa e assicura la regia complessiva dell'operazione, individua gli allievi con le competenze e le capacità per poter fruire pienamente del percorso allo scopo di integrare le competenze sulle tecnologie STEAM.

Coordina il monitoraggio e la valutazione dei risultati, gestisce ed individua gli snodi critici, la loro possibile rimozione e la ridefinizione dell'attività.

Le scuole dell'ambito territoriale di Cesena e Fondazione EnAIP Forlì-Cesena condividono il progetto all'interno della loro comunità scolastica, informano gli allievi e le loro famiglie, preparano assieme all'esperto referente del Fablab Romagna il laboratorio esperienziale per l'acquisizione delle competenze, integrano a livello curricolare le conoscenze che possono rappresentare un arricchimento per la preparazione degli allievi declinandolo nei diversi ordini, gradi di istruzione e ambito di formazione.

Partecipano con la rete di partner territoriali attivamente al laboratorio di rielaborazione dell'esperienza, garantiscono le modalità di accesso dei partecipanti ai tecnopoli, curano attraverso docenti e coordinatori la permanenza e la qualificazione degli allievi coinvolti nel progetto, sostenendo le attività programmate, con lo scopo di promuovere l'occupazione attraverso le STEAM.

Fablab Romagna (FLR) APS individua un esperto che coadiuverà i docenti nella preparazione del laboratorio esperienziale, attraverso una capillare informazione nelle scuole coinvolte. Sostiene le azioni valorizzando la scuola, il lavoro e la formazione professionale.

In particolare cura la progettazione, costruzione, collaudo e messa in attività di alcune periferiche di misurazione dell'inquinamento atmosferico.

Sostiene la scuola pubblica nella realizzazione delle attività e nella ricerca delle nuove tecnologia.

Tecnopoli e Criad Coding definiscono il progetto base curano la formazione per gli allievi delle scuole superiori, medie e elementari su ambienti Scratch/Snap, puntando ad attivare il pensiero computazionale come obiettivo terminale del progetto.

Il percorso avrà ricadute sulla formazione spendibile anche nelle esperienze di transizione, tipo alternanza scuola lavoro.

### **2.4.2 FabLab Romagna**

Fablab Romagna Romagna (FLR) è un'Associazione di Promozione Sociale con lo scopo di trovare punti di contatto tra studenti, makers ed imprese per promuovere la creatività e il bisogno di fare rete.

È il primo Fablab Romagna nato dentro la scuola pubblica italiana frutto di una collaborazione tra insegnanti dell'ITT "Blaise Pascal" di Cesena e dell'ITIG "Belluzzi – da Vinci" di Rimini che volevano mettere a frutto le competenze maturate in una carriera d'insegnamento in qualcosa di concreto e utile alla crescita professionale dei propri studenti.

Ora fanno parte di FLR anche il makerspace FLR all'interno di ITIS "G. Marconi" – Forlì e il gruppo di lavoro Bagno di Romagna (che sta attivando l'apertura presso il Comune di Bagno di Romagna).

Fablab Romagna Romagna è formato da laboratori chiamati makerspace, nati dalle convenzioni stipulate con i rispettivi istituti scolastici o Comuni.

I tre attuali laboratori sono dislocati a Cesena, Rimini e Forlì e hanno attrezzature di produzione digitale come stampanti 3D, laser cutter, macchine CNC, computers, schede a microcontrollore, ma anche torni e apparecchi elettronici per la creazione di circuiti stampati.

Può essere messa a frutto la possibilità di utilizzo delle molte strumentazioni in dotazione agli istituti suddetti, fra i più significativi istituti tecnici del territorio romagnolo.

Tali risorse sono a disposizione sia degli studenti sia di esterni per la realizzazione dei loro progetti scolastici e non.

Le principali attività sono concentrate nella produzione di software e nella realizzazione di circuiti per i progetti di ognuno, nell'uso e nella progettazione di stampanti 3D, nel design e nella stampa di oggetti in 3D utilizzando sia PLA sia ABS.

Alcuni esempi di progetti portati a termine nel Fablab Romagna Romagna sono: Robot, Sistema Domotico, Cubo led ma, anche diverse applicazioni per l'Internet of Things (IoT) e tanto altro.

Tutto questo in un'ottica Open Source, Creative Commons di condivisione di idee, competenze e aiuto reciproco.

Fablab Romagna Romagna ribalta i ruoli tra insegnante e studente creando nuove modalità per lo scambio reciproco di competenze per il raggiungimento di un obiettivo comune in un'ottica più simile ad un team di sviluppo adottando "The Fab Charter" del M.I.T. (Massachusetts Institute of Technology).

FLR si muove dentro e fuori dalla scuola, attivando contatti sinergici fondamentali per la crescita sia della scuola che della società.

Nel progetto MIA il FabLab ha il compito di individuare un esperto che effettuerà la formazione presso le scuole e nei centri interessati promuovendo la connessione con gli Istituti scolastici, il sistema produttivo ed i tecnopoli.

FabLab inoltre ha il compito di sostenere la scuola pubblica nella realizzazione delle attività e nella ricerca delle nuove tecnologie.

### 2.4.3 Tecnopolo Forlì - Cesena

Il Tecnopolo di Forlì-Cesena ospita le sedi dei laboratori dei 3 Centri Interdipartimentali di Ricerca Industriale dell'Università di Bologna, che fanno parte della Rete Alta Tecnologia della Regione Emilia-Romagna, specializzati rispettivamente in tecnologie ICT (intelligenza artificiale e Big Data), tecnologie agroalimentari (con particolare riferimento a innovazioni nel recupero di scarti e sottoprodotti, nello studio del packaging e nella sostenibilità delle produzioni alimentari) e tecnologie aerospaziali (studi fluidodinamica, monito-

raggio satellitare, utilizzo di droni per analisi avanzate del terreno e agricoltura di precisione).

Il Tecnopolo di Forlì-Cesena comprende il laboratorio internazionale Ciclope, presso le ex Gallerie Caproni di Predappio (FC), in cui vengono effettuate sperimentazioni in merito agli effetti delle turbolenze.

Tecnopolo dispone, inoltre della prestigiosa sede di Rocca delle Caminate, location riservata a percorsi di accompagnamento delle imprese (workshops, counselling sessions, design thinking seminars ecc...) e incubazione di start up.

Il Tecnopolo rappresenta il punto di accesso all'ecosistema regionale dell'innovazione e, attraverso di esso, le imprese possono usufruire di un'ampia gamma di servizi specializzati che vanno dal primo orientamento all'analisi dei bisogni tecnologici, all'affiancamento in merito a strumenti finanziari finalizzati all'innovazione, fino a progetti di open innovation e all'organizzazione di eventi di approfondimento.

Il Tecnopolo di Forlì-Cesena fa parte della Rete dei 10 Tecnopoli dell'Emilia Romagna, organismi creati dalla Regione in sinergia con le Università, i Centri di Ricerca e gli Enti locali, con l'obiettivo di organizzare attività e servizi per la ricerca industriale, lo sviluppo sperimentale ed il trasferimento tecnologico.

Fanno parte del sistema regionale dell'innovazione, coordinato da ASTER, società consortile della stessa Regione e ospitano di spazi AREA S3, gestiti dalla stessa ASTER, con l'obiettivo di avvicinare i giovani al mercato del lavoro e all'innovazione tecnologica e produttiva.

Il Tecnopolo si impegna, nel progetto MIA, a sostenere la scuola pubblica nella realizzazione delle attività e nella ricerca delle nuove tecnologia, individuando referenti del criad e criad coding che collaborino con l'esperto di Fablab Romagna nella definizione delle linee di progetto del dispositivo di rilevazione.

In particolare, mette a disposizione gli spazi per la visita formativa e la consulenza tecnologica sia per la progettazione del circuito sia per l'approfondimento scientifico.

All'interno del progetto permette alla rete di scuole di poter effettuare le visite esperienziali presso il Tecnopolo per la definizione dello strumento e la sua applicabilità ai contesti di vita delle persone e delle organizzazioni. All'interno del Tecnopolo sarà avviato un laboratorio esperienziale per l'acquisizione delle conoscenze e per la definizione del dispositivo.

#### **2.4.4 Criad Coding**

Criad Coding, nata ufficialmente a Gennaio del 2018, è una delle sezioni del CRIAD Centro di Ricerca e Studi per l'Informatica Applicata alla Didattica dedicata alle attività inerenti il Pensiero Computazionale ed il Coding.

Ad oggi, la sezione opera attivamente sul territorio delle province di Forlì-Cesena, Ravenna e in alcuni comuni limitrofi.

La missione di Criad è di svolgere attività e progetti sul territorio con le scuole al fine di poter contribuire a costruire competenze in merito al pensiero computazionale e coding promuovendole come strumento utile per potenziare ed esercitare creatività, capacità di problem solving e ragionamento, di progettazione e costruzione, di studio ed apprendimento.

La visione di Criad-Coding include la costruzione di percorsi ed attività basate sul pensiero computazionale utili in progetti e in azioni per contrastare la dispersione scolastica.

Criad svolge attività e progetti sul territorio anche al di fuori del contesto scolastico, sempre con gli obiettivi già definiti, aperti in generale non solo ai bimbi, ragazzi e insegnanti, ma a tutti coloro che vogliono maturare competenze in merito al pensiero computazionale e coding.

L'obiettivo finale è quello di costruire nel tempo una comunità, ovvero una rete di attori sul territorio nazionale e internazionale, a partire dai principi e obiettivi che caratterizzano l'azione e la riflessione di Criad-Coding.

Il Centro di Ricerche e studi per l'Informatica Applicata alla Didattica si pone i seguenti obiettivi:

1. creare una cultura informatica, diffondendo la conoscenza e l'utilizzo delle tecnologie dell'informazione;
2. contribuire allo sviluppo del pensiero computazionale nelle scuole di qualsiasi grado;
3. promuovere e organizzare attività formative rivolte a studenti, lavoratori, professionisti o comunque a tutti coloro che necessitano di competenze informatiche nell'ambito della propria attività di studio, di lavoro o nel tempo libero;
4. organizzare convegni e seminari riguardanti l'Information and Communication Technology;
5. svolgere attività di ricerca, valutazione e sperimentazione di software didattico e utilizzo dell'informatica nella didattica;
6. realizzare prodotti multimediali in collaborazione con enti pubblici e privati collegati a progetti ed attività realizzati nel territorio romagnolo.

Il CRIAD si propone come punto di riferimento per il sistema scolastico locale relativamente alla diffusione del pensiero computazionale e all'utilizzo delle tecnologie informatiche a scopo didattico. La collaborazione con le Istituzioni

scolastiche del territorio si esplica attraverso l'affiancamento, da parte dello staff CRIAD, di tecnici e insegnanti, per la sperimentazione di nuovi software e l'utilizzo di strumenti informatici ad integrazione delle metodologie didattiche classiche.

In particolare il CRIAD-Coding si rivolge a bambini e ragazzi delle scuole primarie e secondarie di primo grado. Tali corsi possono essere svolti, sia in orario scolastico, attraverso percorsi pensati e strutturati in collaborazione con i docenti della classe, sia in orario extra scolastico, con percorsi strutturati e pensati in base alle diverse fasce di età e finalità.

Criad-Coding si è occupato, nel seguente progetto, della formazione su Scratch/Snap! fornendo agli studenti le informazioni necessarie per poter programmare, utilizzando questo tipo di linguaggio, la loro stazione meteorologica. Durante la formazione, durata circa 15 ore, sono stati spiegati i principali concetti del pensiero computazionale e coding tramite la costruzione di un semplice videogioco. Questi concetti verranno poi utilizzati nella fase di programmazione della stazione meteorologica.

### **2.4.5 Scuola Primaria 3 Circolo Didattico Cesena**

Nel presente progetto il terzo Circolo Didattico Cesena si impegna, in accordo con le altre scuole dell'ambito territoriale coinvolte nel presente progetto e gli enti di formazione della Fondazione EnAIP Forlì-Cesena, a:

- condividere con il collegio docenti e i consigli di classe le linee di sviluppo del progetto MIA,
- impostare un approccio alle classi della primaria coinvolte che parta dalla condivisione delle famiglie e degli allievi.

Per poter preparare e realizzare l'esperienza laboratoriale verrà predisposto tutto il necessario; il referente del Fablab Romagna incontrerà la comunità scolastica e il consiglio di classe e comunicando ai docenti interessati le fasi che coinvolgono la scuola condividendo con il docente l'esperienza.

Il docente designato parteciperà con i ragazzi all'esperienza laboratoriale e sarà accompagnato nella fase di preparazione del laboratorio, a cui parteciperanno eventualmente anche i ragazzi dell'ITI e delle medie come formatori tra pari.

Successivamente verrà avviato il laboratorio a cui parteciperà la classe designata in accordo con la comunità scolastica e le famiglie. Saranno programmati degli incontri con l'esperto Fablab Romagna per informare le classi. In questo caso per la piena riuscita del progetto viene individuata una classe possibilmente tra quelle della quinta annualità. L'individuazione dei beneficiari del

progetto tra gli allievi iscritti si muove nella prospettiva di raccordare il mondo della ricerca didattica con le nuove tecnologie STEAM.

Questa strategia sostiene l'accessibilità dei sistemi educativi generali, le misure di accompagnamento individuale e la formazione delle figure didattiche nel sistema educativo.

#### **2.4.6 Scuola Secondaria di Primo grado Comprensivo SMS Viale della Resistenza - Cesena**

Nel presente progetto il Comprensivo SMS Viale della Resistenza - Cesena si impegna in accordo con le altre scuole dell'ambito territoriale coinvolte nel presente progetto e gli enti di formazione della Fondazione EnAIP Forlì-Cesena a

- condividere con il collegio docenti e i consigli di classe le linee di sviluppo del progetto MIA,
- impostare un approccio con le classi coinvolte che parta dalla condivisione con le famiglie degli allievi.

Il referente del Fablab Romagna incontrerà, per garantire la realizzazione dell'esperienza laboratoriale, la comunità scolastica e il consiglio di classe comunicando ai docenti interessati le fasi che coinvolgeranno la scuola.

Il docente sarà accompagnato nella fase di preparazione del laboratorio, a cui parteciperanno eventualmente anche i ragazzi dell'ITI Pascal Comandini come formatori tra pari.

Tra le classi ne verrà designata una, in accordo con la comunità scolastica e le famiglie, con la quale verrà avviato il laboratorio; inoltre verranno programmati degli incontri con l'esperto FabLab per garantire la piena riuscita del progetto.

L'individuazione dei beneficiari del progetto tra gli allievi iscritti, in accordo con gli esperti di Fablab Romagna che collabora alla definizione del progetto MIA si muove nella prospettiva di raccordare il mondo della ricerca didattica con le nuove tecnologie STEAM, la quale sostiene l'accessibilità dei sistemi educativi generali, le misure di accompagnamento individuale e la formazione delle figure professionali nel sistema educativo.

#### **2.4.7 Scuola Secondaria di Secondo grado IIS "Pascal-Comandini" di Cesena FC.**

IS Pascal – Comandini da molti anni collabora con i principali organismi di ricerca del territorio Cesenate, in particolare Fablab Romagna e il dipartimento

CRIAD all'interno dei tecnopoli Universitari, per le attività di integrazione scientifica delle aree professionali presenti nelle qualifiche professionali regionali e con Fondazione EnAIP Forlì-Cesena a progetti inerenti ai nuovi settori di sviluppo professionale a partire dai sistemi tecnologici STEAM.

IS Pascal-Comandini si impegna, nel progetto MIA, in accordo con le altre scuole dell'ambito territoriale coinvolte a programmare un disegno di ricerca per la rilevazione dell'inquinamento a partire dai curricula scolastici, condividere con il collegio docenti e i consigli di classe le linee di sviluppo del progetto MIA, impostare un approccio con le classi coinvolte che parta dalla condivisione tra gli allievi e i docenti coinvolti.

A seguito di questa prima fase preparatoria all'interno del percorso scolastico, l'istituto si occupa di individuare con l'esperto Fablab Romagna e i docenti coinvolti i 20 studenti che parteciperanno all'esperienza, verificando che ciascun allievo possa fruire il percorso con ricadute sulla propria formazione tecnologica spendibile anche nelle esperienze di transizione, tipo alternanza scuola lavoro.

Saranno programmati degli incontri con l'esperto Fablab Romagna per informare le classi e dare la possibilità ai ragazzi di scegliere volontariamente l'adesione al percorso in accordo con il docente e in funzione dei loro progetti di sviluppo occupazionale.

L'individuazione dei beneficiari del progetto tra gli allievi iscritti, in accordo con gli esperti di Fablab Romagna, che collabora alla definizione del progetto MIA, si muove nella prospettiva di raccordare il mondo della ricerca a quello dell'istruzione-formazione tecnologica sul territorio.

Questa strategia sostiene l'accessibilità dei sistemi educativi generali, le misure di accompagnamento individuale e la formazione di figure professionali nel sistema educativo.

L'integrazione delle tecnologie STEAM è uno degli aspetti che contraddistinguono le scelte compiute dalle scuole del territorio nel sistema educativo di istruzione e formazione che può vantare, a differenza di altri contesti, un'esperienza ormai pluriennale.

Una scuola che "pensa" e che "progetta" è una scuola che "non si deve muovere sempre nella condizione di emergenza ma, che si deve muovere sul binario del miglioramento organizzativo" diventando fondamentale per l'avanzamento culturale e professionale dei suoi principali clienti, le famiglie e gli allievi coinvolti.



## 2.5 Lavorare in aula

Le attività del Progetto MIA verranno svolte presso le aule delle scuole aderenti.

Con l'ingresso delle tecnologie negli ambienti scolastici, la figura dell'insegnante si deve adattare e modificare in base alle nuove esigenze, passando da un ruolo di trasmettitore di conoscenze ad una figura che vede l'insegnamento come guida e facilitazione dei processi di apprendimento.

Anche la figura dello studente deve essere rivalutata, non più passivo fruitore ma soggetto attivo che lavora in gruppo per creare conoscenza.

Per questo motivo la classe sarà organizzata, come spesso avviene, in gruppi favorendo il lavoro e lo scambio di idee tra gli studenti.

Ogni gruppo avrà a disposizione un computer per lavorare stimolando così la cooperazione ed il lavoro di gruppo.

Il computer come, dice Papert, in questo modo diventa creta nelle mani degli studenti e dei loro insegnanti con cui, in base alle proprie idee e alle proprie conoscenze, modellare la propria scultura. Gli alunni e gli insegnanti diventano modellatori di realtà artificiale, riducendo la complessità del problema in sotto problemi.

Durante il corso del progetto gli alunni ed i relativi insegnanti saranno affiancati dai docenti di Criad Coding che li aiuteranno nel processo di realizzazione e di modellazione della stazione meteorologica.

Per tutta la durata del progetto, i docenti di Criad, spiegheranno in aula alcuni concetti della programmazione in Snap! comprendendo, in base all'età degli alunni, costrutti come quelli di ripetizione e di selezione.

Durante tutto il percorso sarà necessario ed assolutamente importante, per la riuscita del progetto, integrare con le altre discipline scolastiche la costruzione incrementale del programma identificato come micromondo.

L'obiettivo è quello di far maturare nei bambini e nei ragazzi il concetto del computer come strumento estremamente flessibile con cui creare artefatti digitali di qualsiasi genere portando così lo studente a creare, progettare, sbagliare, correggere, rielaborare ed apprendere.

Questo tipo di attività e di apprendimento richiama la necessità di un cambiamento della didattica tradizionale, da trasmissiva a laboratoriale, strutturata per progetti, che incentivi la collaborazione e la discussione tra gli alunni.

La caratteristica principale del progetto MIA riguarda i diversi livelli di astrazione con cui viene approcciato il lavoro sulla stazioncina meteo basandosi sull'età degli alunni che ci si trova davanti.

La costruzione del micromondo "stazione meteorologica" risulta essere incrementale, alternando fasi di progettazione guidata a fasi di tipo esplorativo

nel quale i bambini ed i ragazzi provano testano e creano in base alla loro fantasia e alle loro conoscenze.

## 2.6 Obiettivo della tesi

Partendo dal riferimento metodologico della scuola costruzionista, dal lavoro di Papert, in cui il ruolo del computer nella scuola viene ripensato diventando uno strumento per manipolare, conoscere, scoprire e costruire, e non più soltanto di gestione delle informazioni.

Uno strumento con cui gli studenti diventano creatori del proprio processo di crescita e apprendimento, questa tesi mira allo studio e sviluppo prototipale di un framework su piattaforma Snap! per attività di apprendimento e Coding in Scuole Primarie e Secondarie utilizzato nell'ambito della creazione di micromondi sulla piattaforma omonima.

La libreria realizzata utilizza come hardware di riferimento una stazione meteorologica con i relativi sensori ed attuatori, come ad esempio i misuratori di inquinamento atmosferico, di temperatura, umidità ecc, costruita su una board Raspberry Pi3.

La realizzazione del framework cerca di rendere l'interazione dello studente con Snap! e con il relativo hardware della stazione meteorologica più semplice nascondendo alcuni elementi di programmazione o di logica che non vengono affrontati durante il percorso didattico e che sposterebbero l'attenzione dal progetto all'assimilazione di concetti puramente informatici.

In base alle diverse età degli alunni sono stati realizzati insiemi di blocchi differenti in modo tale da consentire ad ognuno di esprimersi al meglio utilizzando la piattaforma a disposizione ed allo stesso tempo il pc *come fosse creta*.

Un aspetto interessante del progetto è la comunicazione con le insegnanti ideando micromondi che favoriscano l'apprendimento di concetti e competenze interdisciplinari.

In aggiunta alla creazione dei blocchi base per l'interazione con sensori ed attuatori, si è pensato di realizzare dei blocchi configurabili in modo da consentire agli studenti di crearsi la propria raccolta di dati, attraverso una piattaforma cloud IOT, con la possibilità di visualizzare grafici e/o effettuare misurazioni di vario tipo.

L'aspetto più stimolante del progetto MIA è, a mio parere, smettere di pensare come un programmatore e provare a mettersi nei panni dei bambini o dei ragazzi cercando di vedere le cose a modo loro considerando il computer come un mezzo per esprimersi liberamente mettendo da parte costrutti informatici o matematici.

E' molto importante incentivare la creatività digitale il prima possibile, perché la nuova era del lavoro sarà dominata dalla robotica e dall'intelligenza artificiale.

Il coding sarà considerata la quarta abilità di base per gli studenti delle nuove generazioni, insieme a leggere, scrivere e contare.



## Capitolo 3

# STAZIONE METEOROLOGICA: hardware e software

Il Progetto MIA vede coinvolte tre scuole di diverso ordine e grado. Per consentire a tutti gli studenti di poter usufruire e sfruttare al meglio le proprie capacità sono state realizzate 8 stazioncine meteorologiche.

Ogni stazione verrà accolta in una casina realizzata dai ragazzi dell'associazione Enaip Forlì-Cesena. Il ruolo della casina è quello di non far incontrare direttamente il bambino con la parte prettamente hardware del progetto ma, di rappresentare il tutto come un gioco, come "malleabile" nelle sue mani con l'unico limite della fantasia.

Le casine sono il risultato di tre ingredienti: **creatività, disabilità e produzione artistica** miscelati insieme. Queste non sono solo oggetti colorati, che colpiscono l'attenzione, ma sono inspiegabilmente pieni di forza e significato.

Il Centro Socio Occupazionale della Fondazione En.A.I.P. Forlì-Cesena è un luogo che esce da regole, schemi e barriere dove il lavoro è libera espressione della creatività personale.

Seguendo questi principi associati al coding ed al pensiero computazionale in cui i bambini devono imparare divertendosi; imparare passo a passo risolvendo piccoli problemi impegnandosi per capire quale possa essere la soluzione la stazione sovrastata dalla casina mira a stimolare la fantasia dei bambini che, non conoscendo a fondo tutti i sensori, possono esplorare un mondo che non conoscono riscontrando a loro volta nuovi problemi e/o nuove idee da utilizzare per sfruttare un determinato dato raccolto.

## 3.1 HARDWARE

Le otto stazioni meteorologiche sono tutte composte in ugual modo.

La scheda che si è deciso di utilizzare è una scheda Raspberry PI3 [16]<sup>1</sup>. Raspberry è un single-board [18]<sup>2</sup> computer sviluppato nel Regno Unito. Il single-board computer è una scheda elettronica che implementa un intero computer o quasi. Il single-board si distingue dalla scheda madre di un computer perché è privo di scheda CPU ed ha un numero limitato di slot. L'implementazione dei single-board computer si è diffusa con l'avvento del microprocessore in quanto quest'ultimo ha permesso una drastica riduzione del numero di componenti elettronici necessari. Di norma un single-board computer è un microcomputer in quanto ha come CPU un microprocessore.

Il progetto non prevede né hard-disk né un'unità allo stato solido, affidandosi invece ad una scheda SD. La terza revisione del Raspberry PIh ha sulla scheda anche il WI-FI ed il Bluetooth ed è più potente rispetto ai modelli precedenti.

Le caratteristiche principali di Raspberry PI3 sono:

- 1.2 GHz di CPU
- 1 GB di RAM
- 4 porte USB 2.0
- 40 Pin GPIO.

Si è deciso di utilizzare Raspberry PI3 in quanto può avere diversi utilizzi: dall'istruzione, alla programmazione, realizzazione di device dedicati all'IOT o all'automazione della casa con l'ausilio di vari sensori.

Sul Raspberry PI3 sono stati inseriti dei sensori e degli attuatori di diverso tipo per poter effettuare varie tipologie di misurazioni sull'ambiente. I sensori sono collegati tramite un I2C [17]<sup>3</sup> bus ad eccezione del misuratore della qualità dell'aria in termini di microparticelle che risulta essere seriale. I2C è un bus/protocollo standard che permette una comunicazione veloce, robusta, a due vie, utilizzando un numero minimo di pin.

I2C è stato introdotto da Philips all'inizio degli anni '60 per poi diventare uno standard negli anni '90. I2C utilizza uno spazio di indirizzamento a 7 o 10 bit sfruttando un'architettura master-slave. Un bus I2C è controllato da

<sup>1</sup>[https://it.wikipedia.org/wiki/Raspberry\\_Pi](https://it.wikipedia.org/wiki/Raspberry_Pi)

<sup>2</sup>[https://iol.unibo.it/pluginfile.php/239319/mod\\_resource/content/0/modulo-1.1%20-%20Introduzione%20ai%20Sistemi%20Embedded.pdf](https://iol.unibo.it/pluginfile.php/239319/mod_resource/content/0/modulo-1.1%20-%20Introduzione%20ai%20Sistemi%20Embedded.pdf)

<sup>3</sup>[https://iol.unibo.it/pluginfile.php/239318/mod\\_resource/content/0/modulo-1.2%20-%20Sistemi%20embedded%20basati%20su%20micro-controllore.pdf](https://iol.unibo.it/pluginfile.php/239318/mod_resource/content/0/modulo-1.2%20-%20Sistemi%20embedded%20basati%20su%20micro-controllore.pdf)

un master device, tipicamente un microcontrollore, il quale invia informazioni a uno o più slave. Un master può gestire fino a 1024 slave.

Più device condividono le stesse linee di comunicazione la quale viene sincronizzata da un clock signal.

Tutti i dispositivi che vogliono comunicare con questo protocollo devono condividere il clock signal ed essere identificati da un id. Il master, per poter comunicare con un sensore specifico, manda sul bus l'id del sensore con cui vuole scambiare informazioni; questo viene ricevuto da tutti gli slave ma, l'unico a dover rispondere è colui a cui fa riferimento l'id. Da questo momento in poi la comunicazione tra master e slave avviene "in privato".

I sensori e gli attuatori utilizzati per il Progetto MIA sono:

1. Si7021, sensore di temperatura ed umidità,
2. BMP280, sensore per calcolare la pressione atmosferica,
3. SGP30, sensore per la concentrazione di CO2 e TVOC,
4. PMS5003 misuratore della qualità dell'aria in termini di microparticelle di diversa misura,
5. Led RGB WS2812B,
6. Servo Motore

La board con hostname "progettoMIA" è raggiungibile

1. Da WLAN: (SSID = RPi3, password = raspberry)
2. Da LAN: DHCP
3. SSH (solo pi), VNC (solo root), FTP (solo pi)

Il routing dei pacchetti è disabilitato per cui collegandosi da WLAN non è possibile collegarsi alla rete cablata.

Ovviamente la stazione meteorologica può essere ampliata in futuro; per il momento le proposte future sono l'aggiunta di un anemometro per calcolare la velocità del vento e di un sensore a raggi ultravioletti.

### 3.1.1 Si7021 Temperatura ed umidità

Il sensore Si7021 effettua misurazioni sulla temperatura e sull'umidità dell'ambiente che ci circonda. Il sensore Si7021 effettua misurazioni sull'umidità con precisione del  $\pm 3\%$  in un intervallo 0-80% di umidità relativa; inoltre lo stesso sensore permette anche di misurare la temperatura esterna con precisione di  $\pm 4$  gradi in un intervallo che può variare da -10 a +85 gradi Celsius.

Il sensore è composto da un filtro in PTFE per mantenerlo pulito consentendo così l'ottimalità delle misurazioni.

Si7021 utilizza il trasferimento dei dati fondato su I2C, ciò significa che funziona su un'ampia gamma di micro-controllori. Utilizza due cavi dati/clock I2C disponibili sulla maggior parte dei micro-controllori e può condividere tali pin con altri sensori purché non abbiano una collisione di indirizzi. In questo caso l'indirizzo predefinito per il sensore di temperatura ed umidità è 0x40 e non è più possibile cambiarlo. [10]

Il pin di alimentazione di Si7021 è Vin. Per alimentare il sensore bisogna assegnargli la stessa potenza del livello logico del microcontrollore che si sta utilizzando.

SCL è il pin logico I2C da collegare a linea di clock mentre il pin SDA è il pin dati I2C da collegare alla linea dati dei microcontrollori.

### 3.1.2 BMP280 Pressione atmosferica

Il sensore BMP280, creato da Bosch, misura la pressione atmosferica dell'ambiente circostante; a scelta dell'utilizzatore questo sensore è anche in grado di misurare la temperatura.

BMP280 è un sensore ambientale di temperatura e pressione barometrica ed è ideale per tutti i tipi di rilevamenti atmosferici; inoltre può essere utilizzato su bus I2C o SPI.

Questo sensore Bosch è la migliore soluzione di rilevamento di precisione a basso costo per misurare la pressione barometrica, con un'accuratezza assoluta di  $\pm 1$  hPa, e la temperatura con una precisione di  $\pm 1$ . Poiché la pressione cambia con l'altitudine e le misurazioni barometriche sono così meticolose, il sensore può anche essere utilizzato come altimetro con una precisione di  $\pm 1$  metro.

BMP280 è composto da tre pin di alimentazione, i pin logici SPI ed i pin logici I2C: SCK e SDI.[11]

Il pin SCK è il pin di clock I2C da connettere alla linea di clock del microcontrollore mentre SDI rappresenta il pin dati I2C da connettere alla linea dati.

La pressione barometrica viene calcolata in Pascal ed è un numero intero a 32 bit mentre la temperatura risulta essere sempre in virgola mobile, in gradi centigradi. E' possibile trasformare il BMP280 in un altimetro, se si conosce la pressione a livello del mare, è possibile calcolare la pressione barometrica in altitudine.

L'indirizzo I2C scelto per il sensore BMP280 è 0x77.



### 3.1.3 SGP30 Concentrazione di CO2 e TVOC

Questo è un sensore di qualità dell'aria molto fine con interfaccia I2C e segnali di uscita completamente calibrati con un'accuratezza tipica del 15% entro i valori misurati. SGP combina più elementi di ossido di metallo su un chip per fornire segnali relativi alla qualità dell'aria più dettagliati.

SGP30 è un sensore di gas in grado di rilevare un'ampia gamma di composti organici volatili (COV).

Se collegato al microcontrollore restituirà una lettura del composto organico volatile totale (TVOC) ed una lettura equivalente di anidride carbonica (CO2) su I2C.

L'SGP30 ha un sensore MOX a piastra calda "standard", nonché un piccolo microcontrollore che controlla l'alimentazione alla piastra, legge la tensione analogica, traccia la calibrazione di base, calcola i valori TVOC ed CO2 e fornisce un'interfaccia I2C da cui leggere.

La concentrazione di CO2 verrà calcolata in un intervallo che varia da 400 a 60.000 parti per milione (ppm) e la concentrazione TVOC (Total Volatile Organic Compound) in un intervallo da 0 a 60.000 parti per miliardo (ppb).

Si noti che questo sensore, come tutti i sensori VOC/gas, ha variabilità e per ottenere misurazioni precise è necessario calibrarlo basandosi su fonti note. Un elemento interessante di questo sensore è la possibilità di impostare la compensazione dell'umidità per una migliore precisione. È necessario un sensore di umidità esterno che scriva l'umidità su I2C, in modo da poter calcolare meglio i valori TVOC/CO2.

L'indirizzo I2C utilizzato per il sensore SGP30 è 0x58. Per sua costruzione il sensore di CO2/TVOC deve effettuare un certo numero di letture prima di dare valori corretti (necessità di "warm-up" durante il quale viene fatta una calibrazione in funzione di temperatura e umidità dell'ambiente). Queste vengono fatte prima di avviare il server in modo che una volta avviato la richiesta di lettura dei suoi parametri sia immediata.[12]

### 3.1.4 PMS5003 Qualità dell'aria

Il PMS5003 è una specie di sensore di concentrazione di particelle, digitale e universale, che può essere utilizzato per ottenere il numero di particelle sospese nell'aria, ovvero la concentrazione, e inviarle sotto forma di interfaccia digitale.

Questo sensore può essere inserito in strumenti relativi alla concentrazione di particelle sospese nell'aria o altre apparecchiature di miglioramento ambientale per fornire dati di concentrazione corretti nel tempo.

Il sensore utilizza il principio di funzionamento eLaser scattering, vale a dire produrre scattering utilizzando il laser per irradiare particelle sospese nell'aria,

quindi raccoglierne la luce posizionandosi in un certo grado di angolazione e infine ottenere la curva di scattering con il cambiamento della luce nel tempo. Alla fine, il diametro delle particelle ed il numero di particelle con diametro differente per unità di volume possono essere calcolati dal microprocessore.

In output il sensore emette per ciascuna particella, con volume unitario diverso, il volume unitario e l'unità di concentrazione di massa. Esistono due opzioni per l'uscita digitale: passiva e attiva. La modalità predefinita dopo l'accensione è attiva. In questa modalità il sensore invierebbe automaticamente i dati seriali all'host. La modalità attiva è divisa in due sotto-modalità stabile e veloce. Se il cambiamento di concentrazione è piccolo, il sensore dovrebbe funzionare in modalità stabile mentre se il cambiamento è grande, il sensore dovrebbe cambiare automaticamente in modalità veloce con un intervallo di 200-800 ms; maggiore è la concentrazione, più breve risulta essere l'intervallo.

Il sensore PMS5003 restituisce in output il numero delle particelle suddivise per diametro; in particolare restituisce tre tipologie di sigle:

- PM10
- PM1
- PM2,5

PM (Particulate Matter) è il termine generico con il quale si definisce un mix di particelle solide e liquide (particolato) che si trovano in sospensione nell'aria. Il PM può avere origine sia da fenomeni naturali sia principalmente da attività antropiche, in particolar modo dai processi di combustione e dal traffico veicolare. Esiste, inoltre, un particolato di origine secondaria che si genera in atmosfera per reazione di altri inquinanti come gli ossidi di azoto, il biossido di zolfo, l'ammoniaca ed i Composti Organici Volatili. Il PM si suddivide per grandezza delle particelle.

La sigla PM10 identifica una delle numerose frazioni in cui viene identificato il particolato, quel materiale presente nell'atmosfera in forma di particelle microscopiche, il cui diametro aerodinamico è uguale o inferiore a 10 millesimi di millimetro.

È costituito da polvere, fumo, micro gocce di sostanze liquide denominato in gergo tecnico aerosol, per la piccola dimensione delle particelle solide e liquide.

Queste particelle presenti nell'atmosfera sono indicate con molti nomi comuni: polvere, fuliggine e caligine per quelle solide e nebbia per quelle liquide.

Le principali fonti di PM10 sono

1. Sorgenti legate all'attività dell'uomo

- processi di combustione,
- usura di pneumatici,
- freni ed asfalto.

Uno studio sull'inquinamento da PM10 ha rivelato che la maggior causa della origine e dispersione di queste particelle è data dalla "combustione di biomasse legnose" (quindi stufa a pellets o legna) che contribuiscono per il 45% alle polveri sottili diffuse nell'aria, i motori diesel contribuiscono per il 14% e un altro 13% è dato da particelle che si staccano dalle pastiglie dei freni e dai pneumatici.

## 2. Sorgenti naturali

- l'erosione del suolo,
- gli incendi boschivi,
- le eruzioni vulcaniche,
- la dispersione di pollini,
- il sale marino.

La sigla PM<sub>2,5</sub> è una frazione del particolato totale interamente contenuta nella frazione di PM<sub>10</sub> presenti nell'atmosfera terrestre per cause naturali e antropiche o in luoghi di lavoro industriali.

Il PM<sub>2,5</sub> è una classificazione numerica data alle polveri sottili in base alla loro grandezza. Più il numero è minore e più sottili sono le polveri e dunque più pericolose per la salute della specie umana ed animale. Il PM<sub>2,5</sub> si forma maggiormente tramite l'industria ed i prodotti industriali. Un esempio sono i freni degli autoveicoli che consumandosi emettono PM<sub>2,5</sub> ma che non sono la fonte del maggiore inquinamento.

In uno studio effettuato dal 2004 al 2008, campionando i dati di 100 giornate, in alcune città si sono verificati livelli di PM<sub>2,5</sub> che hanno superato fino a 3 volte il valore della soglia limite di 50.[13]

L'inquinamento atmosferico rappresenta un rischio per la salute umana. L'inquinamento non agisce soltanto sull'apparato respiratorio, infatti, batteri, virus e allergeni o le sostanze contenute nel fumo possono causare disturbi a livello più generale così come le sostanze inquinanti presenti nell'aria.

Gli idrocarburi volatili e il monossido di carbonio penetrano nei polmoni e raggiungono il cervello ed altri organi tramite il sangue. Il particolato infine provoca effetti anche sul sistema cardiocircolatorio.

Gli studi epidemiologici hanno evidenziato una relazione lineare fra l'esposizione a particelle ed effetti sulla salute. Vale a dire, che quanto più è alta

la concentrazione di particelle nell'aria tanto maggiore è l'effetto sulla salute della popolazione. Allo stato attuale delle conoscenze, secondo l'Organizzazione Mondiale della Sanità non è possibile fissare una soglia di esposizione al di sotto della quale certamente non si verificano nella popolazione degli effetti avversi sulla salute. Per questo motivo, l'OMS non fornisce un valore guida di riferimento per le particelle, ma indica delle "funzioni di rischio" per i diversi effetti sulla salute.

Tali funzioni quantificano l'eccesso di effetto avverso per la salute che ci si deve aspettare per ogni incremento unitario delle concentrazioni di PM10 o di PM2,5.

### 3.1.5 Led RGB WS2812B

La tecnologia LED utilizza diodi montati su strisce in grado di emettere molta luce con un basso consumo energetico. Un LED RGB utilizza tre diodi: rosso, verde e blu. Questi vengono miscelati con intensità diverse per produrre una varietà di colori, la stessa tecnica impiegata per televisori, monitor e schermi piatti. La luce proveniente da una fonte di colore può sovrapporsi a quella di un'altra, creando una nuova e più brillante espressione di colore.

I ledi RGB posseggono 4 terminali e si possono presentare in due tipi di configurazione:

- 1 anodo e 3 catodi (RGB ad anodo comune)
- 3 anodi e 1 catodo (RGB a catodo comune)

In serie ad ogni LED sarà inserita una resistenza che consentirà di regolare la corrente circolante nel diodo.

La striscia LED indirizzabile WS2812B è disponibile in diversi modelli che differiscono per dimensioni, sigillante o densità del LED. Con WS2812B è possibile controllare la luminosità ed il colore di ciascun LED singolarmente, il che ti consente di produrre effetti sorprendenti e complessi in modo semplice.

Questa striscia LED è composta da LED WS2812B cablati in serie. Questi LED hanno un IC integrato, ciò consente una comunicazione tramite un'interfaccia a filo singolo. Ciò significa che puoi controllare molti LED usando solo un pin digitale.

La striscia LED deve essere alimentata utilizzando una fonte di alimentazione a 5V in quanto ogni LED assorbe circa 50 mA quando impostato alla massima luminosità.

I LED RGB sono controllati dal PIN 18 (PWM hardware) della RPi.[14] <sup>4</sup>

---

<sup>4</sup><https://www.maffucci.it/2014/09/27/arduino-lezione-09-uso-di-led-rgb-parte-1/>

### 3.1.6 Servo Motore

I servomotori sono piccoli motori composti da un perno che permette di effettuare una rotazione e vengono spesso utilizzati nell'ambito della robotica. Solitamente questi si presentano come piccoli contenitori di materiale plastico da cui fuoriesce un perno in grado di ruotare in un angolo compreso tra 0 e 180 gradi mantenendo stabilmente la posizione raggiunta. La rotazione del motore è effettuata tramite un circuito di controllo interno in grado di rilevare l'angolo di rotazione raggiunto dal perno tramite un potenziometro resistivo e bloccare il motore sul punto desiderato.

I Servomotori sono concepiti per essere pilotati nel modo più semplice possibile, eseguendo la movimentazione senza l'ausilio di circuiterie troppo complesse o l'uso di sistemi a microprocessore. Un servomotore dispone solitamente di soli tre fili attestati ad un connettore femmina.

La maggior parte dei servo standard viene progettata in modo da ruotare avanti e indietro da 90 gradi a 180, in funzione degli impulsi di temporizzazione, anche se gran parte è in grado di effettuare una rotazione completa a 180 gradi, o molto vicina a questo valore.

Il pin che controlla il servo motore è il pin 17.[15]<sup>5</sup>

## 3.2 REST API

Le stazioni meteorologiche sono accessibili tramite due tipologie di indirizzi:

- 1- LAN,
- 2- WAN.

Inizialmente l'url WAN per accedere alla stazione era http ma, a causa del funzionamento di Snap! che, viene eseguito come web app scaricata da una connessione HTTPS, è stato necessario modificare la chiamata come tale. Inoltre è stato necessario abilitare lato server CORS; questo poiché alla fine il micromondo Snap! viene visto come script Javascript scaricato da un server che tuttavia cerca di accedere ad un altro server.

CORS è un meccanismo che utilizza intestazioni HTTP aggiuntive per indicare ai browser di consentire ad una applicazione Web, in esecuzione su un'origine, di accedere a risorse selezionate da un'origine differente. Per motivi di sicurezza, i browser limitano le richieste HTTP di origine incrociata avviate dagli script. Ciò significa che un'applicazione Web che utilizza tali API può richiedere risorse dalla stessa origine da cui è stata caricata l'applicazione, a meno che la risposta di altre origini non includa le intestazioni CORS corrette. Lo standard di condivisione delle risorse tra origini funziona aggiungendo

---

<sup>5</sup>[https://www.adrirobot.it/servotester/il\\_servomotore.htm](https://www.adrirobot.it/servotester/il_servomotore.htm)

nuove intestazioni HTTP che consentono ai server di descrivere quali origini sono autorizzate a leggere tali informazioni da un browser Web. La specifica impone che i browser "eseguano il preflight" della richiesta, sollecitando i metodi supportati dal server, quindi, dopo "l'approvazione" dal server, avviene l'invio della richiesta effettiva. Per poter leggere i valori misurati e calcolati dai vari sensori sulla board sono state istituite delle REST API in figura 3.1 e in figura 3.2.

Route	Metodo HTTP	Esempio Input	Esempio Output
<code>/api/temp</code>	GET	/	<pre>{   "temp":   22.01585205078124 }</pre>
<code>/api/hum</code>	GET	/	<pre>{   "hum":   71.53372192382812 }</pre>
<code>/api/press</code>	GET	/	<pre>{   "press":   1010.9997206356333 }</pre>
<code>/api/co2TVOC</code>	GET	/	<pre>{   "co2": 400,   "tvoc": 125 }</pre>
<code>/api/servo</code>	POST	Header = Content-Type:application/json Body = <pre>{   "angolo": 255 }</pre> Valori angolo da 0 a 255 (rotazione di 180° in senso orario e antiorario)	Movimento del servomotore
<code>/api/rgb</code>	GET	/	Spegne LED RGB

Figura 3.1: Rest Api

Route	Metodo HTTP	Esempio Input	Esempio Output
<code>/api/qual</code>	GET	<code>/</code>	<pre>{   "quality": [     {       "pm1_std": 6.0     },     {       "pm25_std": 9.0     },     {       "pm10_std": 11.0     },     {       "pm1_atm": 6.0     },     {       "pm25_atm": 9.0     },     {       "pm10_atm": 11.0     },     {       "part_03": 1038     },     {       "part_5": 2     },     {       "part_1": 66     },     {       "part_25": 8     },     {       "part_10": 0     }   ] }</pre>
<code>/api/rgb</code>	POST	<p>Header = Content-Type:application/json</p> <p>Body =</p> <pre>{   "rgb": [     {       "r": 255,       "g": 0,       "b": 0     },     {       "r": 0,       "g": 255,       "b": 0     },     {       "r": 0,</pre>	I LED RGB sul prototipo si colorano secondo quanto indicato in input

Figura 3.2: Rest Api



Causa l'implementazione di CORS, è necessario eseguire alcuni passaggi sul browser su cui viene mandato in esecuzione Snap!, per consentire il corretto funzionamento delle chiamate https utili per la lettura dei sensori e degli attuatori.

Lato browser web è fondamentale copiare l'url corrispondente alla stazione meteorologica configurata associato ad una chiamata rest (es temperatura) ed accettare di continuare a lavorare su questo certificato.

I siti web garantiscono la propria identità attraverso i certificati e, non considerando il sito attendibile, in quanto utilizza un certificato non valido, il browser propone all'utente l'accettazione o no del rischio.

Una volta accettato il rischio apparirà una schermata contenente, in questo caso la temperatura in formato JSON. Se la configurazione è andata a buon fine lato Snap! tutte le chiamate di lettura di sensori ed attuatori restituiranno un valore. E' importante sapere che, nel caso in cui si voglia mandare in esecuzione Snap! su un browser differente da quello su cui è stata effettuata la configurazione dovranno essere replicati tutti i passaggi.



# Capitolo 4

## Snap! Build Your Own Blocks

Creato da Jens Mönig e Brian Harvey dell'Università della California a Berkeley, Snap! [19]<sup>1</sup>, formalmente BYOB-Build Your Own Blocks, è un linguaggio di programmazione open source, visuale e caratterizzato da un'interfaccia drag and drop per la definizione di codice che consente la programmazione diretta da browser web.

Snap! può essere visto come una reimplementazione estesa di Scratch al quale viene aggiunta la possibilità di costruire i propri blocchi sintattici e semantici a complessità crescente, definendone la categoria di appartenenza, il nome, il tipo, e la lista dei parametri. Questa funzionalità consente di diventare creatori di mondi strutturati ed organizzati sulla base degli obiettivi prefissati.

### 4.1 La nascita di Snap!

Snap! è un linguaggio di programmazione grafico, educativo, gratuito, basato su blocchi da far corrispondere tra loro come tasselli, rivolto agli studenti per esplorare, creare e mescolare animazioni interattive, giochi, storie e altro ancora, imparando a conoscere idee matematiche e computazionali.

Brian Harvey, nato nel 1949 è un ex docente di informatica dell'Università della California ed insieme al programmatore tedesco Jens Mönig ha progettato il linguaggio di programmazione Build Your Own Blocks (BYOB) ed il suo successore Snap!.

La prima modifica di Jens fu Chirp, un programma che presentava miglioramenti a Scratch. Successivamente iniziò a lavorare su Build Your Own Blocks, un progetto molto più grande composto da tre versioni basate principalmente su Scratch; fino al 2009 quando l'Università della California a Berkeley decise di creare un nuovo corso di informatica per non laureati.

---

<sup>1</sup><http://snap.berkeley.edu/SnapManual.pdf>

Il corso prevedeva l'utilizzo di Scratch come linguaggio di programmazione, ma voleva anche insegnare la ricorsione e le funzioni di ordine superiore. Ecco come Brian Harvey, uno degli sviluppatori del nuovo corso, si unì al progetto BYOB, lavorando con Jens per progettarne l'interfaccia utente.

Nasce così Snap!, un ambiente di sviluppo in cui realizzare micromondi animati, basato su un linguaggio a blocchi, open-source ed eseguibile direttamente da browser.

Quest'ultimo può essere visto come una reimplementazione di Scratch al quale vengono aggiunti alcuni elementi non presenti originariamente in Scratch e presi in prestito dal linguaggio Scheme. La prima versione fu rilasciata nel 2011 e venne utilizzata in un corso introduttivo di Computer Science per studenti non informatici.

Snap! importa da Scratch:

- Interfaccia drag-and-drop
- Facili strumenti di animazione
- Metafore visuali per cicli, condizioni, etc.

Snap! importa da Scheme elementi first-class:

- Procedure
- Liste
- Oggetti

Queste caratteristiche risultano essenziali per poter proiettare l'utilizzo di Snap! dalla didattica all'utilizzo accademico.

Snap! come Scratch viene utilizzato sia in ambito didattico, dai bambini, sia nel tempo libero dai non esperti di informatica grazie alla semplicità ereditata da Scratch, ma anche dagli informatici e dagli esperti del settore grazie ai costrutti avanzati messi a disposizione.

Con gli elementi visivi forniti, gli utenti possono realizzare animazioni, testi, storie, musica oppure creare blocchi nuovi per rendere ancora più customizzato il proprio micromondo. Esistono già molti programmi all'interno della community Snap! che gli studenti possono utilizzare, elaborare o modificare. L'uso di Snap! consente ai giovani di comprendere la logica della programmazione e come costruire e collaborare in modo creativo.

## 4.2 User Interface

L'ambiente "Snap!" è fruibile direttamente via Internet, ovvero utilizzabile tramite un qualunque browser. Questa è la mappa della finestra di Snap! suddivisa in tre aree principali: Palette, Scripting Area e Stage.

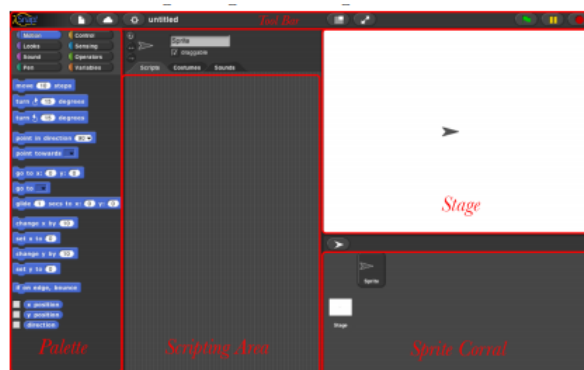



Figura 4.1: Schermata Snap!

La *Palette* è l'area nella quale sono presenti tutti i blocchi suddivisi in categorie con colori di riferimento differenti in base alle loro caratteristiche. Da quest'area l'utente può scegliere un blocco, trascinarlo nell'area di *Scripting*, dove avviene la vera e propria realizzazione del micromondo, e iniziare a realizzare il proprio script.

Lo *Stage* rappresenta il "teatro" dove lo script viene messo in esecuzione e tutte le idee dell'utente vengono rese visibili come se fosse un vero e proprio spettacolo.

Cliccando sul logo di Snap! è possibile accedere ad un menu di opzioni che mostrano informazioni su Snap! stesso compresi i numeri di versione per i moduli sorgente, gli implementatori della piattaforma e la licenza.

L'icona di File Menu  mostra un menu contenente alcune voci, tra le principali quella di salvataggio, l'importazione di sorgenti esterne, l'esportazione di blocchi o del progetto e l'importazione di librerie. L'opzione **Importa** permette di importare risorse esterne nel progetto corrente, anziché caricare un progetto completamente separato. Si possono importare costumi (qualsiasi formato di immagine supportato dal browser), suoni o sprite precedentemente esportati.

**Esporta Progetto** consente di salvare il progetto su disco o di aprire una nuova scheda del browser contenente il progetto completo in notazione XML.

**Esporta Blocchi** è l'alternativa che viene utilizzata per creare una libreria di blocchi. Questa presenta un elenco di tutti i blocchi globali del progetto e consente di selezionare quali esportare. Per memorizzarli si può scegliere di

aprire una nuova scheda di navigazione contenente i blocchi in formato XML oppure salvarli direttamente sul disco locale.

Un aspetto fondamentale per quanto riguarda la programmazione in generale è il riuso di codice, anche Snap! adotta questa strategia, infatti, tramite l'opzione **Moduli** è possibile importare delle librerie all'interno del progetto.

Le librerie che verranno utilizzate per la creazione della stazione meteorologica sono principalmente:

- **WEB SERVICES ACCESS**, contenente due importanti blocchi, il primo, un'estensione della versione del blocco http, consente di effettuare chiamate GET, POST, PUT e DELETE utilizzando il protocollo https. Il secondo *current location*, utilizzando la geo localizzazione, permette di ottenere la posizione dell'utente che sta utilizzando il programma.
- **DEAL WITH JSON DATA**, contenente due blocchi. *Litify* permette di convertire un dato JSON in una lista chiave valore e *value at key of list* consente di estrapolare i dati ottenuti definendone la chiave e la lista da cui estrarli.

### 4.3 Scrivere un programma in Snap!

Un programma in Snap! consiste in uno o più *scripts*, ognuno dei quali realizzato da *blocchi*. Questo è un tipico script:

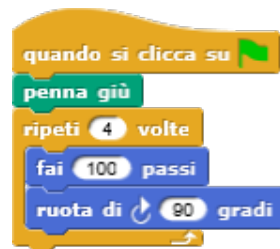


Figura 4.2: Esempio di script base in Snap!

I blocchi dello script sono di colori differenti, corrispondono a tre delle otto *palettes* in cui si possono trovare. L'area delle palette si trova sulla parte sinistra della schermata.

I blocchi gialli provengono dalla palette di Controllo, quelli verdi dalla palette Penna mentre i blu dalla palette dei Movimenti. Lo script viene assemblato trascinando i blocchi dalla palette all'area di scripting.

I blocchi si attaccano tra loro grazie alle indentazioni (da qui il nome Snap!).

*"Blocks snap together(hence the name Snap! for the language) when you drag a block so that its indentation is near the tab of the one "*

Nella parte alta dello script si trova *hat block*, tipicamente rappresentato dalla parola "when"; lo script inizia quando la bandiera verde viene cliccata. Uno script non può avere più di un hat block e questo può essere utilizzato solo nella parte superiore. Altri blocchi presenti sono i *command blocks* questi corrispondono alle azioni che lo sprite deve eseguire all'interno dello stage.


Snap mette a disposizione dell'utente la possibilità di poter modificare il *costume* dello sprite rispetto all'**Alonzo costume, simbolo di Snap!, che prende il nome dalla chiesa di Alonzo, un matematico che ideò le procedure come dati.**

Per poter modificare il costume esistono due modi, il primo, cliccando su file icon e scegliendo "Costumes", permette di scegliere da una lista di costumi provenienti da librerie pubbliche mentre il secondo, da la possibilità di caricare un'immagine direttamente dal pc purché in formato compatibile con il browser.

In aggiunta al costume lo sprite può avere anche un suono. I file musicali possono essere importati in qualunque formato supportato dal browser.

Esistono principalmente due blocchi per l'utilizzo del suono che sono  per permettere allo script di continuare l'esecuzione mentre il suono è in riproduzione e  che attende il completamento del suono prima di continuare lo script.

Prevedendo la possibilità di lavorare direttamente sul browser web Snap! dispone di due diverse modalità di salvataggio dei progetti. La ragione di questa complessità è che Javascript, con cui è implementato Snap!, limita deliberatamente la capacità dei programmi in esecuzione in un browser di "affettare" il computer; questo significa che mandando in esecuzione i progetti Snap! di altri utenti non c'è il rischio di eliminare i propri ma, rende le cose un po' complesse.

Cliccando sulla ToolBar scegliendo l'icona  appare un menù con due possibili scelte : Cloud o Browser.

Nel caso in cui l'utente selezioni browser il progetto verrà salvato in un file speciale che può essere letto solo sullo stesso computer, dallo stesso browser, collegato alla stessa pagina web. In questo caso il progetto non verrà visualizzato in un elenco di file al di fuori di Snap!, questo è come Javascript li protegge dai malware.

Questa procedura porta a due ulteriori possibilità:

- Localstore

Un'importante limitazione di questa modalità di salvataggio è la quantità totale di spazio di archiviazione disponibile la quale verrà impostata dal browser, per tutti i siti Web combinati. (Questo limite può essere modificato nelle

preferenze del browser). Inoltre, se il browser è configurato per disabilitare i cookie dai siti Web non consentirà questa circostanza di salvataggio.

- XML

Il secondo modo per salvare un progetto sul computer richiede due passaggi, ma non ha limiti. I progetti salvati con questa modalità sono normali file sul pc e possono essere condivisi con gli amici, possono essere aperti in qualsiasi browser e non hanno limiti di dimensione. Dal menu file, scegliere "Esporta progetto ...", a seconda del browser in alcuni casi il progetto verrà scaricato sul computer in un file nella directory dei download in altri casi, invece, l'intera finestra di Snap! scomparirà, sostituita da una schermata rappresentante il progetto in sintassi XML.

Un'ulteriore possibilità è salvare il progetto in "cloud". Per fare ciò è necessario avere un account Snap!.

Dopo aver salvato un progetto se si desidera ricaricarlo sulla piattaforma basterà aprilo o dal menu "File" scegliendo Browser o Cloud oppure importandolo se salvato in formato XML.

Snap! concede anche l'importazione di progetti creati in BYOB 3.0 o in Scratch 1.4 o 2.0. Quasi tutti questi progetti funzionano correttamente in Snap!, a parte un piccolo numero di blocchi incompatibili.

La prima versione di Snap!, chiamata BYOB "Build Your Own Blocks", fu la più importante funzionalità aggiunta a Scratch. La differenza tra i due linguaggi, come sopra descritto, è la possibilità fornita da Snap! di creare nuovi blocchi.

Nella palette si trova il pulsante "**Make a block**" tramite il quale si apre una finestra di dialogo nella quale è possibile scegliere la forma ed il nome del blocco. Seguendo una convenzione che dovrebbe essere familiare agli utenti di Scratch esistono tre forme di blocco.

I blocchi a forma di puzzle sono comandi e non riportano alcun valore. I blocchi ovali sono Reporter e i blocchi esagonali sono Predicati, riportanti valori booleani (true o false).

Supponendo di voler creare un blocco chiamato "quadrato" che disegna un quadrato si sceglie Movimenti, Creazione blocco digitando "square" nel campo del nome. Quando si fa clic su OK, si entra nell'Editor blocchi che funziona proprio come la creazione di una sceneggiatura nell'area di scripting dello sprite, tranne per il fatto che il blocco "hat" in alto, invece di dire qualcosa di simile a "quando mi viene richiesto", ha un'immagine del blocco che si sta costruendo ovvero il prototipo del blocco personalizzato.

Trascinando i blocchi sotto al "cappello" si programma il blocco personalizzato dopo di che il tutto viene salvato. Come per tutti i blocchi preesistenti è possibile ideare un custom block che prende in ingresso uno o più parametri.



Esistono dodici tipi di forme di input oltre a tre categorie che si escludono a vicenda. Il tipo di default è "qualsiasi", il che significa che lo slot in input deve accettare qualunque valore di qualunque tipo.

La disposizione degli input è sistematica, come mostra l'immagine ogni riga rappresenta una categoria.

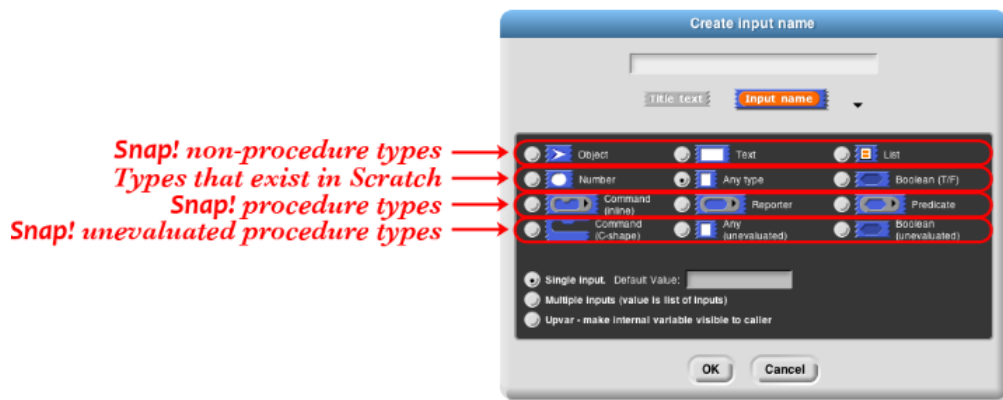


Figura 4.3: Tipologie di input per la creazione dei blocchi

La seconda riga contiene le tipologie di input esistenti in Scratch: Number, Any e Boolean.

A volte sarebbe utile poter salvare, non solo i blocchi personalizzati creati all'interno del progetto stesso ma, anche una raccolta di blocchi che si ritiene possano essere utili in più di un progetto. Per poter salvare la propria libreria personale si utilizza il meccanismo di esportazione XML sopra descritto scegliendo la funzione "Esporta blocchi". Verrà aperta una finestra la quale mostra tutti i custom blocks.



Figura 4.4: Schermata di esportazione blocchi a scelta

È possibile deselezionare alcune delle caselle di controllo per selezionare esattamente quali blocchi si desidera includere nella propria libreria; premendo ok si aprirà una nuova scheda con le definizioni in codifica XML.

## 4.4 Programmazione Object Oriented con Sprites

La programmazione orientata agli oggetti è un paradigma di programmazione, in cui un programma viene visto come un insieme di oggetti che interagiscono tra loro tramite messaggi. Un aspetto da enfatizzare di OOP molto importante per gli utenti di Snap! è la simulazione con cui ogni oggetto rappresenta in modo astratto qualcosa nel mondo.

L'approccio che Snap! ha con la programmazione ad oggetti è quindi meno restrittiva rispetto ad altri linguaggi object oriented. Snap! consente agli oggetti un facile accesso ai dati ed ai metodi di altri.

Tecnicamente, la programmazione object oriented si basa su tre pilastri:

(1) Passaggio di messaggi: esiste una notazione tramite la quale qualsiasi oggetto può inviare un messaggio a un altro oggetto.

(2) Stato locale: ogni oggetto può mantenere in memoria il risultato di una chiamata ad un metodo. (Questo significa che non è necessario ricordare tutti i messaggi che ha gestito, ma solo gli effetti duraturi di quei messaggi che influenzeranno il calcolo successivo.)

(3) Ereditarietà: Sarebbe poco pratico se ogni singolo oggetto dovesse contenere metodi simili o tal volta identici a quelli di altri oggetti, per questo motivo l'ereditarietà mi permette di specificare che un oggetto eredita i metodi di un altro preesistente con la possibilità di ampliarli.

Come Scratch anche Snap! fornisce degli oggetti: sprites. Ogni sprite può possedere variabili locali ed ha i suoi script (metodi). Un'animazione di Scratch è chiaramente una simulazione dell'interazione dei personaggi in un'opera teatrale. Ci sono due aspetti tali per cui gli sprite di Scratch sono meno versatili degli oggetti di un linguaggio OOP. In primo luogo, il passaggio dei messaggi è primitivo sotto tre aspetti:

1. i messaggi possono essere solo trasmessi e non indirizzati a un singolo sprite;
2. i messaggi non possono accettare input;
3. i messaggi non possono accettare input ed i metodi non possono restituire valori al chiamante.

In secondo luogo nel paradigma OOP gli oggetti sono dati; possono essere il valore di una variabile, un elemento di un elenco e così via, ma questo non è il caso degli sprite. Gli sprite di Snap! sono *firstclassdata* ciò significa che possono essere creati ed eliminati da uno script, archiviati in una variabile o in un elenco inoltre possono inviare messaggi singolarmente. I figli di uno sprite possono ereditare variabili *sprite-local*, metodi (procedure *sprite-local*) e altri attributi (ad es. Posizione *x* ). Il mezzo fondamentale con cui i programmi hanno accesso agli sprite è il blocco *myreporter* il quale contiene uno slot di ingresso che dà accesso a tutti gli sprite, oltre al palco.



Figura 4.5: Idea di script che sfrutta OOP con l’invio di un messaggio

Molti linguaggi orientati agli oggetti utilizzano le classi/istanze per approssimare alla creazione di oggetti. La classe è un particolare tipo di oggetto mentre un’istanza è un oggetto reale di quel tipo.

Snap! utilizza un approccio differente chiamato "prototipo", con il quale non crea distinzione tra classe ed istanza. Il "prototyping" potrebbe sembrare strano ma, in realtà è un sistema molto espressivo in quanto consente di simulare facilmente una gerarchia di classe/istanza nascondendo il prototipo *sprite*.

Snap! può eseguire diversi script contemporaneamente, all’interno di un singolo *sprite* e in molti *sprite*. Alla fine di ogni script c’è un comando implicito "end thread". Tutto questo avviene in automatico perciò, in genere, non è necessario che l’utente pensi ai thread.



## Capitolo 5

# Sviluppo prototipale di un Framework su piattaforma Snap! per attività di apprendimento e coding

Un framework, termine della lingua inglese che può essere tradotto come intelaiatura o struttura, in informatica e specificatamente nello sviluppo software, è un'architettura logica di supporto su cui un software può essere progettato e realizzato, spesso facilitandone lo sviluppo da parte del programmatore. Alla base di un framework c'è sempre una serie di librerie

Lo scopo delle librerie software è fornire una collezione di entità di base pronte per l'uso ovvero riuso di codice, evitando al programmatore di dover riscrivere ogni volta le stesse funzioni o strutture dati e facilitando così le operazioni di sviluppo e manutenzione. I vantaggi principali derivanti dall'uso di un simile approccio sono i seguenti:

- Si può separare la logica di programmazione di una certa applicazione da quella necessaria per la risoluzione di problemi specifici, quali il calcolo di funzioni matematiche o la gestione di collezioni;
- Le entità definite in una certa libreria possono essere riutilizzate da più applicazioni.

La definizione appena riportata risulta essere prettamente informatica e rigorosa. In questo caso la realizzazione della libreria sfrutterà un approccio più pragmatico.

## 5.1 Framework su piattaforma Snap!

Il framework verrà utilizzato sulla piattaforma Snap! per la creazione di micromondi. Sfrutterà come hardware sottostante una board realizzata con Raspberry Pi3 su cui sono stati montati dei sensori e degli attuatori che interagiscono con il mondo esterno.

L'idea non è quella di realizzare una libreria o un framework che segua le specifiche informatiche ma, creare un insieme di blocchi che la compongano, basandosi sulle età differenti dei bimbi e dei ragazzi che la manipoleranno, per consentire loro di realizzare micromondi che abbraccino le loro competenze e la loro fantasia.

Il progetto MIA ha come obiettivo quello di portare il coding nelle scuole di diverso ordine e grado dimostrando agli alunni e agli insegnanti la potenza del computer a scuola e nella didattica.

Utilizzare un computer a scuola non è importante per capire l'informatica, per sapere come è fatto un calcolatore o per conseguire la patente europea, ma per comprendere meglio le varie e complesse possibilità che l'uomo ha a disposizione per esplorare e conoscere la realtà che lo circonda.

Un aspetto fondamentale della libreria sottostante sarà quello di comporre blocchi creando diversi livelli di astrazione utilizzabili. Ogni alunno deve potersi approcciare con lo sviluppo dei micromondi nella maniera più semplice possibile senza doversi fermare davanti ad aspetti strettamente legati alla programmazione.

Le scuole coinvolte nel progetto MIA sono tre di diverso ordine e grado e tre dovranno anche risultare i livelli di astrazione realizzati.

Per quanto concerne la scuola primaria verrà utilizzata una strategia basilare fornendo tutti i blocchi di cui i bimbi hanno bisogno nascondendogli qualunque aspetto di programmazione spinta o di logica matematica che non abbiano svolto nel percorso didattico. Diversa sarà, invece la strategia per le scuole medie e superiori dove saranno lasciati alla realizzazione di ogni studente blocchi che includano logiche o ragionamenti matematici oppure strettamente informatici consentendo loro non solo, di interagire con il mondo esterno tramite il computer ma, di imparare a ragionare e a comporre idee complesse che racchiudano argomenti trattati a scuola.

## 5.2 Idea iniziale

Questa tesi mira allo studio e alla realizzazione di framework su piattaforma Snap! per attività di apprendimento e coding in Scuole Primarie e Secondarie utilizzato nell'ambito della creazione di micromondi sfruttando come hardware una board Raspberry.

L'idea di un micromondo così caratterizzato permette di avere uno strumento flessibile con cui i bambini o i ragazzi possano progettare esperienze che spazino da ambienti totalmente virtuali ad ambienti ibridi, che includano elementi del mondo fisico in cui vivono, sfruttando però tecnologie embedded.

L'idea della libreria sottostante consiste nel mettere a disposizione degli alunni un insieme di blocchi predefiniti con cui, limitati solo dalla loro fantasia, possano creare micromondi differenti in base anche alla loro età ed alle loro conoscenze. Inizialmente si era pensato ad un unico insieme di blocchi, principalmente per la lettura e l'interazione con sensori ed attuatori ma, successivamente ragionando sulle diverse età e sulle diverse abilità degli studenti e delle scuole coinvolte si è pensato di realizzare un insieme base di blocchi da fornire condito, sulla base della scuola in questione, da altri blocchi che nascondano o mettano in luce alcuni concetti di programmazione.

Per lo sviluppo dei micromondi sono stati pensati e sviluppati, in stretta collaborazione con gli insegnanti, dei percorsi paralleli ai contenuti che gli alunni avrebbero visto nelle materie curriculari per favorire l'apprendimento di concetti e competenze interdisciplinari.

Ad estendere la libreria, l'idea di dare la possibilità di utilizzare i dati raccolti per effettuare misurazioni e/o elaborazioni di qualunque tipo. Per i più piccoli ma, non solo, il pensiero di appoggiarsi ad un piattaforma IOT preesistente in grado di raccogliere, elaborare e graficare i dati in modo da rendere il tutto non solo didatticamente più stimolante ma visivamente più facile da capire e da incamerare per l'alunno.

Un aspetto cruciale della libreria è capire quali tipologie di blocchi implementare e quali a loro volta fornire agli studenti in base al loro livello di studi permettendo di definire il livello di astrazione con cui si vuole sviluppare l'attività di apprendimento, ovvero l'insieme di concetti da utilizzare nel ragionamento e nella progettazione, poiché risultano significativi, e, dualmente gli aspetti da cui astrarre.

Definire un livello di astrazione e l'insieme dei blocchi significa definire anche linguaggio che gli alunni useranno per rappresentare il problema e ragionare sulla soluzione o sul progetto da sviluppare. Tale insieme deve, perciò, essere progettato in relazione all'età, alle competenze, e più in generale alle specificità degli alunni.

### 5.3 Progettazione e Sviluppo

La progettazione del framework è iniziato studiando il manuale di Snap e la stazione meteorologica nel suo hardware. Sulla piattaforma Snap è stato realizzato un primo progetto sul cloud chiamato "test". Su test sono stati rea-

lizzati tutti i blocchi e poi, solo in un secondo momento esportati in tre progetti differenti per scuola chiamate rispettivamente "LIBPrimaria", "LIB1Grado" e "LIB2Grado".

Partendo dall'hardware della stazione meteorologica sono stati creati come primi i blocchi per la lettura dei dati dai sensori e dagli attuatori.

La lettura dei sensori richiede l'utilizzo di chiamate https; per poterle effettuare in Snap! è stato necessario l'utilizzo del modulo "Web services access(https)", un'estensione del blocco http il quale permette di effettuare delle chiamate POST, PUT, DELETE e GET utilizzando il protocollo https, ottenendo il controllo degli headers.

Inizialmente l'url WAN per accedere alla stazione era http ma, a causa del funzionamento di Snap! che, viene eseguito come web app scaricata da una connessione HTTPS, è stato necessario modificare la chiamata come tale. Inoltre è stato necessario abilitare lato server CORS; questo poiché alla fine il micromondo Snap! viene visto come script Javascript scaricato da un server che tuttavia cerca di accedere ad un altro server.

Snap! mette a disposizione una palette di controllo composta da 8 colori differenti che identificano le varie suddivisioni delle aree di esecuzione dei blocchi.

I primi blocchi creati sono i blocchi della lettura dei sensori e l'interazione con gli attuatori, e come tali sono stati creati all'interno dell'area "Sensori", colore azzurro della palette. Tra questi ci sono:

- leggi temperatura,
- leggi umidità,
- leggi pressione atmosferica,
- leggi co2,
- leggi composti organici volatili,
- qualità dell'aria,
- accendi led (colore predefinito),
- accendi led RGB,
- spegni tutti i led,
- leggi polveri ultra fini,
- leggi polveri fini,



- leggi polveri grossolane.

Ognuno di questi blocchi è stato realizzato sia come Comando, caratterizzato dalla forma puzzle il quale non restituisce nessun valore in uscita ma, setta direttamente all'interno dello script una variabile che poi verrà visualizzata, sia come Monitor di forma ovale il quale restituisce un valore e viene utilizzato come input in altre tipologie di blocco.

Tutti i blocchi che leggono i dati dai sensori sono stati realizzati allo stesso modo.

Un aspetto fondamentale della libreria è la configurazione delle stazioni meteorologiche.

Essendo state costruite 8 stazioni, che in futuro possono aumentare, lo studente o l'utente che sfrutterà la libreria dovrà avere la possibilità di configurare su quale stazione lavorare andandone a specificare l'indirizzo ip che, ovviamente potrà essere modificato in qualsiasi momento.

Al cospetto di questo è stato realizzato il blocco di configurazione 

Quest'ultimo è un blocco di tipo Controllo in quanto agisce direttamente sul controllo della stazioncina.



Figura 5.1: Editor di blocchi-Configura stazione meteo ip.

La sua costruzione risulta molto semplice, in realtà, sfruttando il blocco "porta a", che mi consente di settare il valore di una variabile, porto, in base all'input ottenuto dall'utente, confidando che l'input sia un indirizzo ip corretto, il valore della variabile globale ip all'indirizzo.

Questa variabile globale mi servirà poi in tutti i blocchi che interagiscono con la stazione in quanto, verificando l'ip si è in grado di ottenere il valore corretto desiderato.

La lettura dei sensori avviene come spiegato in precedenza utilizzando le rest api. Prendendo in esame il blocco "leggi temperatura" come prima cosa viene verificata la variabile ip.

Nel caso in cui l'ip risulti uguale a zero, ovvero l'utente non abbia configurato correttamente la stazione su cui operare, questo viene settato all'indirizzo ip base definito nelle specifiche.

Nel caso in cui l'ip sia stato configurato, alla variabile locale ipFinal, definita all'interno dello script, viene associata la chiamata rest da effettuare scegliendo anche il tipo di chiamata tra GET, POST, PUT e DELETE.

Ogni sensore restituisce come risultato un JSON contenente il valore richiesto; per poter sfruttare e, visualizzare il risultato è stato necessario l'utilizzo di un modulo predefinito chiamato "Deal with JSON data" contenente i blocchi per poter lavorare ed estrarre dati dai Json specificando chiave valore.

In base alla richiesta che si sta effettuando all'interno dello script, nel blocco "value at key of" viene specificato in input la chiave JSON corrispondente al dato da ottenere come risultato. In ultimo viene settata la variabile, ad esempio temperatura, la quale indica la temperatura rilevata dal sensore sulla board.

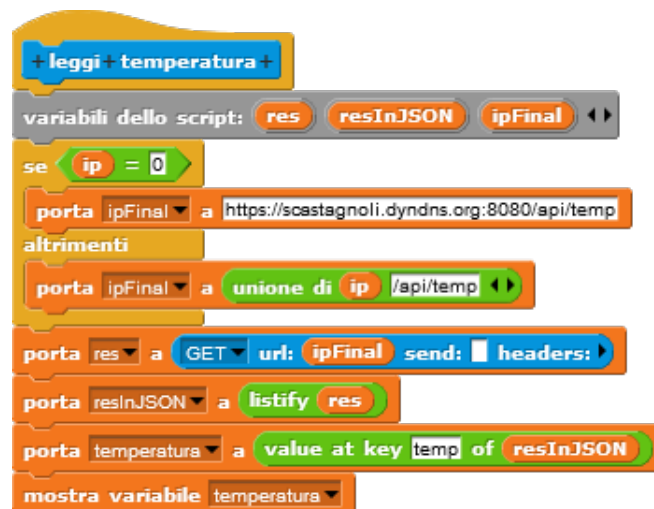


Figura 5.2: Editor di blocchi-Leggi temperatura.

Tutti i blocchi che leggono dati dai sensori sono stati implementati seguendo la stessa logica del `leggi temperatura`, andando a modificare la rest api da associare all'ip, in base al valore richiesto.

Allo stesso modo sono stati implementati i blocchi di tipo Comando che leggono valori dai sensori; l'unica variante di questa tipologia di blocchi è che non viene settata una variabile alla fine del blocco ma, lo script usa il blocco di base "Result" per mantenersi il risultato ottenuto e, all'occorrenza restituirlo come output.

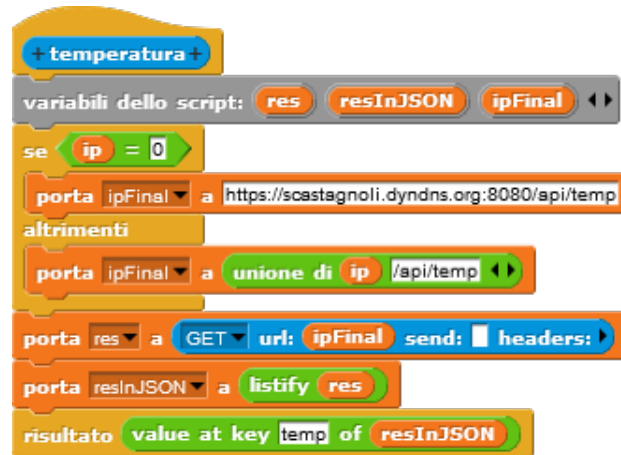


Figura 5.3: Editor di blocchi-Leggi temperatura(Comando).

**Accendi led (colore predefinito)** e **Spegni tutti i led** sono blocchi molto semplici. Agiscono direttamente sul led andando rispettivamente ad accenderlo o a spegnerlo effettuando una semplice chiamata https di tipo POST senza ottenere nessun valore di ritorno.

Accendi led di colore predefinito permette di accendere il led RGB andando a specificare come valore di default 255 per le tre componenti. In contrasto a quest'ultimo, è stato realizzato il blocco `accendi led R G B`.

Come il precedente è un blocco della categoria Sensori ma, in questo caso viene permesso all'utente di specificare come parametri in input dei valori da 0 a 255 per tutti e tre le tipologie di colore RGB. La sua implementazione è leggermente più complessa rispetto alla precedente in quanto, dopo aver verificato che i tre input siano valori effettivamente compresi tra 0 e 255, viene riformulato il JSON da utilizzare per effettuare la chiamata https.

La riformulazione del JSON avviene sfruttando il blocco unisci di tipo "Operatori" che unisce due o più parametri specificatigli in input.

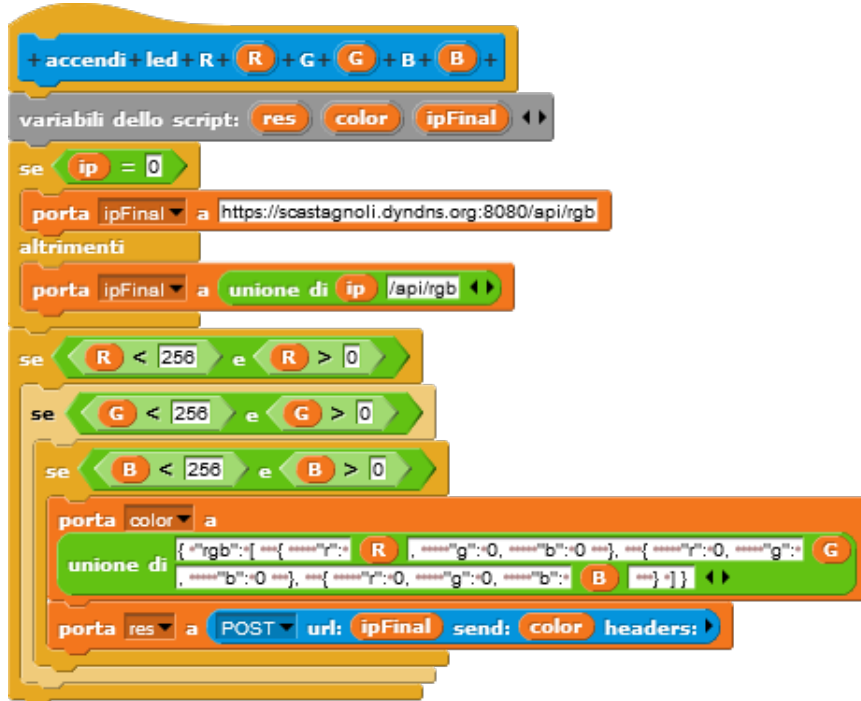


Figura 5.4: Editor di blocchi-Accendi led RGB.

All'interno della libreria sono stati realizzati tre blocchi nella sezione "Movimento":

- Accendi ventola,
- Spegni ventola,
- Gioco luci.

Accendi e spegna ventola sono due blocchi che permettono all'utente di accendere il servo motore specificandone un angolo che va da 0 a 255 gradi o spegnerlo.

Il movimento del servo potrebbe simulare il movimento di una ventola che viene accesa, quando la temperatura risulta troppo alta e spenta quando questa si abbassa sotto una certa soglia un po' come fanno i condizionatori.

L'implementazione dei blocchi appena descritti è corrispettiva all'implementazione dell'accensione dei led effettuando una POST request inserendo come parametro l'angolo preso in ingresso dopo aver verificato che questo sia compreso tra 0 e 255.

**Gioco di luci** è un blocco molto particolare e, a mio parere molto divertente. Permette di creare un gioco di luci sui led RGB montati nella board modificandone il valore dei tre componenti. Il gioco di luce che viene a crearsi

non è un vero e proprio aspetto rilevante all'interno del progetto ma, permette all'alunno di poter non solo misurare, comporre il proprio micromondo e sfruttare i dati ottenuti per effettuare operazioni ma anche creare una visione ibrida tra le misurazioni e ciò che sta avvenendo nel computer con ciò che avviene sulla board.



Figura 5.5: Editor di blocchi-Gioco di colori.

Un aspetto fondamentale della libreria realizzata per il Progetto MIA sono i livelli di astrazione che verranno creati e di conseguenza utilizzati per le scuole coinvolte. Ogni alunno deve poter essere in grado di sfruttare la libreria secondo le proprie conoscenze ed abilità senza doversi soffermare su implementazioni prettamente informatiche o matematiche che lo distoglierebbero dal vero obiettivo del progetto. Il computer deve diventare non solo uno strumento per permettere allo studente di mettere in pratica le proprie conoscenze creando micromondi, ragionando, sbagliando e rielaborando ma, anche uno strumento creativo da poter modellare a piacimento secondo la propria creatività.

A questo proposito sono stati realizzati due blocchi di monitoraggio dei valori, entrambe con lo stesso aspetto:



I due blocchi di monitoraggio sono stati creati con l'idea di inserirli in due libreria differenti ovvero il primo nella libreria della scuola primaria ed il secondo nella libreria per gli alunni della scuola secondaria di primo grado.

Il primo blocco, come anche il secondo, prevede cinque parametri in ingresso:


1. indica il numero di minuti/ore ogni quanto si vogliono monitorare i valori,
2. questo parametro permette due possibili scelte ore o minuti; nel caso in cui il monitoraggio debba essere fatto ogni x ore verrà scelta la prima in caso contrario la seconda,
3. il terzo ed il quarto periodo rispecchiano i primi due parametri ma, in questo caso indicano per quanti minuti o ore deve essere effettuata l'azione di monitoraggio del dato,
4. l'ultimo parametro specifica il valore da monitorare scegliendo da un menù a tendina contenente le varie opzioni. Lo script per prima cosa, prima di entrare nel cuore dell'implementazione utilizza il costrutto `for` ed il costrutto `if` concatenati tra loro. In questo modo lo studente può concentrarsi sull'utilizzo che può fare del monitoraggio dei dati senza dover ragionare su aspetti di programmazione di cui non ha le basi.

Si è deciso di fornire agli alunni il monitoraggio già implementato in quanto, dopo aver consultato il piano didattico, si è ritenuto troppo complesso e non costruttivamente utile per un bimbo delle elementari o un ragazzo delle scuole medie imparare come si sfruttano il costrutto `for` ed il costrutto `if` concatenati tra loro. In questo modo lo studente può concentrarsi sull'utilizzo che può fare del monitoraggio dei dati senza dover ragionare su aspetti di programmazione di cui non ha le basi.

La differenza sostanziale tra i due blocchi di monitoraggio sta nell'output ottenuto. Il primo blocco setta, basandosi sul valore da monitorare, una variabile globale realizzando una lista di dati ottenuti nel tempo. Il secondo blocco di monitoraggio non si limita a restituire una lista di valori ma, setta la variabile globale somma con il numero di dati contenenti in essa.

Questo avviene in quanto, il blocco di monitoraggio è stato pensato per raccogliere un insieme di valori nel tempo su cui poi effettuare delle misurazioni e/o delle operazioni. Un'operazione plausibile che si può applicare su una raccolta di dati è il calcolo della media.

La media, per quanto banale, necessita di una base matematica solida; per quanto concerne le scuole elementari, i bimbi nel loro percorso didattico matematico non affrontano il concetto di media mentre i ragazzi delle medie hanno sicuramente le basi per poter implementarla in autonomia. Questo è il motivo del secondo blocco di monitoraggio il quale restituisce la somma di valori con cui calcolare la media.

Onde evitare di rendere il tutto troppo complesso per la scuola primaria, è stato realizzato, sotto la palette "Operatori", il blocco .

Questo, in base al parametro specificatogli in input scorre tutta la lista di dati calcolandone la media e restituendola in output.

Ad estendere la libreria, l'idea di dare la possibilità di utilizzare i dati raccolti per effettuare misurazioni e/o elaborazioni di qualunque tipo. Per i più piccoli ma, non solo, il pensiero di appoggiarsi ad un piattaforma IOT preesistente in grado di raccogliere, elaborare e graficare i dati in modo da rendere il tutto non solo didatticamente più stimolante ma visivamente più facile da capire e da incamerare per l'alunno.

La piattaforma scelta per raccogliere e graficare i dati è DWEET <sup>1</sup>.

Dweet è una piattaforma iot con la quale è possibile interagire, inviando dei dati, attraverso l'url che si trova sulla loro piattaforma. Ogni studente può iscriversi a dweet in modo gratuito creandosi la propria dashboard.

Dweet permette di realizzare dei grafici lineari che indichino l'andamento dei dati inviati.

I dati inviati verranno mantenuti per 24 ore dopo di che verranno automaticamente eliminati ma, la dashboard continuerà ad essere utilizzabile. Un aspetto divertente di dweet è la customizzazione che ogni utente può fare rispetto al nome dei dati che egli invia.

Questa piattaforma, come le stazioni meteo ha bisogno di essere configurata prima di poter inviare i dati, per questo è stato realizzato, lato Snap!, il blocco "Configura dweet". Quest'ultimo prende in ingresso l'url indicato nella piattaforma web dweet, differente per ogni utente, e, come per la configura-

---

<sup>1</sup><https://dweet.io/>

zione della stazione meteo, viene settata la variabile globale urlDweet con l'url corrispondente.

Una volta configurata la piattaforma cloud iot per la raccolta dei dati, tramite i blocchi `invia dato` e `invia lista di dati` è possibile inviare rispettivamente una lista di dati oppure un dato singolo specificando a piacimento la denominazione da assegnargli.

Questi due blocchi non fanno altro che, verificare inizialmente se dweet è stato configurato e, solo in caso positivo, tramite una POST request all'url indicato inviare i relativi dati.



Figura 5.6: Editor di blocchi-Invia lista di dati.



Un ultimo blocco realizzato è **Grafico a torta dati**; questo ha una struttura molto complessa, sfrutta la rotazione degli angoli, la realizzazione di cerchi e di rettangoli per poi monitorare i valori andando a creare degli angoli colorati all'interno della torta per delineare l'andamento della qualità dell'aria nell'ambiente in cui viviamo.

## 5.4 Diversi livelli di astrazione

Per ogni scuola coinvolta verrà utilizzato un metodo di lavoro differente "nascondendo" o "mostrando" alcune implementazioni in modo da consentire a tutti gli studenti di potersi esprimere e poter lavorare nel modo migliore e più conforme alle loro capacità ed alla loro età.

Per portare il coding nelle scuole è fondamentale creare dei progetti che portino gli studenti ad interfacciarsi con la creazione e l'ideazione di micromondi senza doversi soffermare su concetti di programmazione o di logica che non hanno già affrontato all'interno del percorso didattico.

Per il Progetto MIA che vede coinvolte tre scuole di diverso ordine e grado sono stati ideati e realizzati tre livelli di astrazione differenti.

Ovviamente non è possibile pensare che il programma didattico della scuola primaria sia uguale a quello della scuola secondaria di primo grado tanto meno della scuola secondaria di secondo grado, allo stesso modo le tre librerie devono avere delle differenze sostanziali.

L'ideazione e l'implementazione è stata proprio realizzata tenendo conto dei diversi livelli di preparazione degli alunni coinvolti. Un esempio lampante di come sono stati pensati i blocchi, è il monitoraggio dei valori ed il calcolo della relativa media.

Gli alunni della scuola primaria nel percorso didattico matematico-scientifico non si trovano ad affrontare il calcolo della media e tanto meno i costrutti di programmazione spinti come ad esempio il costrutto `for` ed il costrutto `if`, parte fondamentale dell'implementazione del blocco.

Per i bimbi è stato deciso di fornire, già implementati, sia il blocco di monitoraggio composto da cinque parametri in ingresso con cui giocare per creare monitoraggi differenti tra loro, sia il blocco per il calcolo della media.

In questo modo i ragazzi possono sfruttare tutto ciò che gli viene messo a disposizione per realizzare dei micromondi che abbraccino le loro competenze didattiche e creando dei percorsi interdisciplinari anche perché l'obiettivo del progetto MIA è quello di portare il coding nelle scuole e non di insegnare aspetti di programmazione o logica.

Tenendo sempre conto del percorso didattico e formativo delle scuole coinvolte, esaminando l'aspetto matematico relativo alla scuola secondaria di primo

grado si è notato che gli alunni vengono a conoscenza del concetto di somma, media e lista andando anche ad esplorarne alcuni aspetti più in profondità.

Fondamentale risulta essere, però, la non formazione rispetto ad elementi di logica o di programmazione che possano aiutarli a capire i cicli o degli aspetti logici relativi all'informatica vera e propria.

Come per la primaria, anche in questo caso è stato ritenuto necessario fornire il blocco monitora già funzionante lasciando invece nelle mani degli alunni l'implementazione del calcolo della media, in questo caso, o comunque di opportune operazioni sui dati.

Portare il pensiero computazionale nelle scuole non è semplice, bisogna ideare e comporre dei progetti che portino i bambini, come poi i ragazzi ad esplorare concetti che già conoscono riuscendo, blocco dopo blocco a comporre e visualizzare l'idea che avevano in mente. Questo approccio consente all'alunno di imparare giocando, riutilizzare argomenti studiati a lezione o della vita quotidiana portandoli all'interno del computer. Il computer diventa perciò, un mezzo con cui creare, con cui realizzare micromondi di qualsiasi genere che mescolino realtà, fantasia e virtuale.

I micromondi realizzati saranno a complessità crescente, basati sulle indicazioni delle insegnanti ma, soprattutto, sulle idee dei ragazzi coinvolti. Gli alunni della scuola secondaria di secondo grado hanno una preparazione molto più elevata, completa costruita su basi logiche, matematiche, scientifiche e di ragionamento solide.

In questo caso l'idea di fornire i blocchi preimpostati per il monitoraggio ed il calcolo della media avrebbe ridotto notevolmente la complessità dei micromondi e la possibilità dei ragazzi di interagire con costrutti logici e di programmazione che possano aiutarli nel ragionamento e nell'aumento della capacità di problem solving.

La decisione presa ha visto la non fruizione del blocco per il calcolo della media tanto meno quello per il monitoraggio. Gli studenti verranno lasciati liberi di implementare questi ultimi come preferiscono sfruttando ragionamenti più o meno complessi andando a sviluppare una capacità di problem solving sempre più solida e ricca.

In figura 5.7 vengono mostrati tre diversi script che utilizzano i concetti appena descritti.

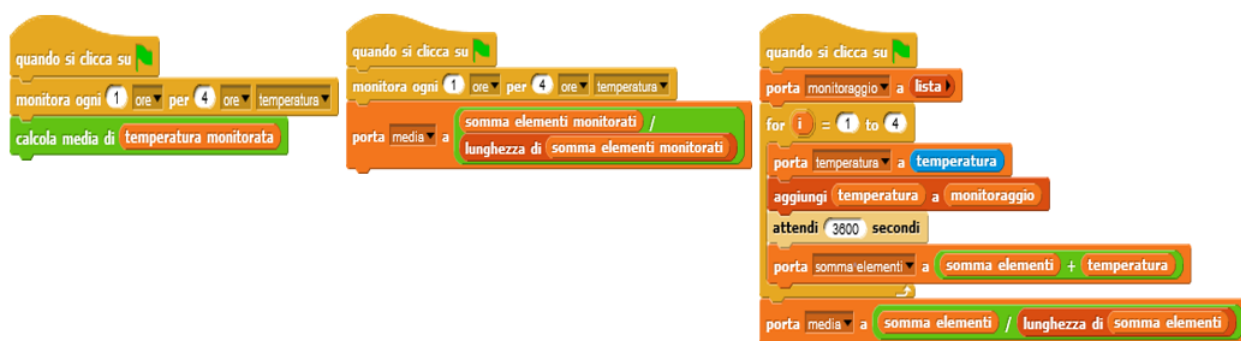


Figura 5.7: Tre diversi script per il monitoraggio.

Tutti e tre gli script in figura rappresentano il monitoraggio della temperatura ogni ora per un totale di quattro ore ed il relativo calcolo della media dei valori ottenuti.

Partendo da sinistra vediamo lo script inerente alla "LIBPrimaria". Come descritto sopra in questo caso i bambini hanno a disposizione tutti i blocchi potendo comunque ragionare sulla logica con cui costruire lo script ma, senza doversi interfacciare con argomenti da loro non studiati o aspetti che li distoglierebbero dal vero e proprio obiettivo del progetto MIA.

Il secondo script è leggermente più complesso del primo. Agli alunni delle scuole medie viene fornito il blocco *monitora* il quale può essere modificato andando a specificare parametri differenti. A differenza del blocco "monitora" per le scuole elementari, l'implementazione di quest'ultimo prevede in output la lista contenente i valori monitorati e la somma totale dei valori ottenuti.

Questo scalino crescente di complessità permette all'alunno di poter e soprattutto dover ragionare su come sfruttare gli elementi di una lista per poterne calcolare la media o effettuare qualsiasi operazione risulti opportuna. Risulta così evidente la differenza, se pur sottile, tra le librerie della scuola primaria e della scuola secondaria.

Le scuole superiori risultano essere un po' staccate dalle prime due in quanto tutto lo script viene lasciato all'implementazione dello studente.

La parte destra della figura 5.7 ci mostra come i ragazzi dovranno approcciare il monitoraggio ed il calcolo della media andando a ricercare nei loro ragionamenti dei costrutti matematici e/o informatici che già conoscono incastrandoli nella maniera corretta per realizzare una logica implementativa funzionante.

Il blocco *monitora* ed il calcolo della media sono solo un esempio, forse il più evidente, di come sono state pensate le tre librerie e di come queste siano state realizzate con tre livelli di astrazione totalmente differenti basandosi su chi le dovrà utilizzare.

### 5.4.1 Scuola Primaria

Il coding nella scuola primaria è un valore aggiunto in quanto consente agli alunni di prendere confidenza, molto gradualmente, non con la tecnologia ma, con le facoltà mentali e con le capacità umane, quindi un giorno diventarne utenti consapevoli e soprattutto di fare dell'informatica e di ciò che l'informatica rappresenta un proprio bagaglio culturale.

E' importante parlare di computer e informatica nella scuola primaria poiché sono tecnologie formidabili per l'apprendimento.

Il pensiero computazionale può essere uno strumento di crescita e di espressione personale. Saper scrivere con le nuove tecnologie permette di esprimere sé stessi e di raccontarsi. Un micromondo permette di esplorare, combinare, elaborare, sbagliare e rielaborare.

Mettere in atto il code to learn esplorando micromondi, lavorando, cooperando in gruppi e con gli insegnanti, sfruttando il coding diventa per lo studente un ottimo mezzo per ragionare, capire, esplorare problemi e contenuti curricolari, nonché immaginare e creare.

Le metafore e le chiavi narrative sono fondamentali per permettere agli alunni a digiuno di qualsiasi concetto di programmazione di avere un quadro di riferimento generale per interpretare man mano i vari concetti che vengono introdotti e concretamente utilizzati nella costruzione dei micromondi.

La metafora e chiave narrativa più generale utilizzata è stata la teatralizzazione, ovvero la visione di un programma (su Snap!) come spettacolo teatrale da mettere in scena. La descrizione di un copione include sequenze di istruzioni, che tipicamente l'attore deve eseguire a fronte di determinati eventi che avvengono durante l'esecuzione dello spettacolo.

Questa metafora permette agli alunni di diventare immediatamente operativi nella formulazione di programmi di una certa complessità.

La libreria realizzata per la Scuola Primaria, denominata "LIBPrimaria", comprende un insieme di blocchi studiati ad hoc per le esigenze dei bimbi.

La creazione dei blocchi ha richiesto la documentazione del percorso curricolare delle varie materie per la scuola in questione. Si è scelto di fornire agli alunni tutti i blocchi realizzati in modo tale da tenere un livello di astrazione che non vada a toccare la programmazione vera e propria ma, che gli permetta di giocare mescolando le proprie conoscenze ed abilità con i valori restituiti dai vari blocchi.

"Interagendo con la simulazione, lo studente capisce innanzitutto perché vede sullo schermo del computer i fenomeni mentre si svolgono e cambiano nel tempo, ma soprattutto capisce perché può agire sulle condizioni che nella simulazione fanno avvenire i fenomeni e ne determinano le caratteristiche,

variando tali condizioni, modificando variabili e parametri, e può osservare gli effetti sui fenomeni di queste azioni”.[22]

Per portare i bimbi dentro al progetto e fargli interpretare i concetti introdotti per la costruzione dei micromondi sono state sfruttate delle metafore e chiavi narrative. In cooperazione con le insegnanti della scuola, esaminando il piano didattico delle varie materie sono stati ideati dei micromondi che sfruttino concetti già appresi.

Le materie principali che sono state esaminate sono:

1. Scienze,
2. Storia,
3. Musica,
4. Geografia,
5. Italiano.

Le materie didattiche appena descritte sono state utilizzate per realizzare micromondi esempio partendo dalle idee degli insegnanti per, innanzitutto verificare la funzionalità della libreria e, in secondo luogo testare le diverse tipologie di utilizzo.

Ovviamente gli esempi realizzati potranno poi essere utilizzati come base di partenza per ideare i progetti da far realizzare ai bimbi o ai ragazzi.

#### 5.4.1.1 Scienze

**Scienze** è la materia principale su cui ci si è concentrati in quanto risulta essere la più affine al progetto. Sfolgiando il piano didattico ci è saltato subito all’occhio il capitolo sul corpo umano in cui vengono spiegati, senza andare nello specifico, gli organi i tessuti e le malattie principali che possono affliggerlo.

Il micromondo pensato vede come protagonista principale un essere umano disegnato dai bimbi che, nel momento in cui mandato in esecuzione lo script, inizia raccontando cos’è il corpo umano e come è composto; una breve introduzione che ogni alunno potrà comporre secondo le proprie abilità e secondo ciò che ha imparato a lezione.

Terminata l’introduzione, utilizzando i blocchi di lettura dei sensori vengono letti tutti i dati riguardanti temperatura, umidità, pressione atmosferica, co2, tvoc e qualità dell’aria.

Proprio quest’ultimo dato risulterà fondamentale per poter definire, in base ai valori di pm10, pm1 e pm2.5 ottenuti, i cambiamenti che questi apportano al nostro corpo anche in termini di malattie. Il processo è stato ideato in

maniera visiva, ad esempio nel caso in cui il valore di pm10 risulti alto il colore del polmone e del sangue diventa più scuro simulando l'effetto dell'inalazione del cosiddetto "fumo nero" nell'organismo.

Tutto ciò è stato pensato per rendere il più semplice, più creativo e più didatticamente utile il micromondo.

#### 5.4.1.2 Storia

Un altro micromondo ideato sfrutta come materia curricolare **storia**. Da sempre l'uomo ha cercato di misurare il tempo, prima con gli antichi gnomoni babilonesi ed egizi e poi con le meridiane greche e romane, per arrivare alle clessidre ed agli orologi portatili. La realizzazione di questo micromondo ha richiesto molta fantasia ed una documentazione dettagliata del programma didattico di storia.

L'idea principale è quella di comparare gli strumenti che oggi utilizziamo per misurare o a cui associamo le condizioni atmosferiche come ad esempio un termometro, pale eoliche, grafici, le stazioni meteo, l'anemometro, ecc con gli strumenti e le tecniche utilizzate dagli antichi.

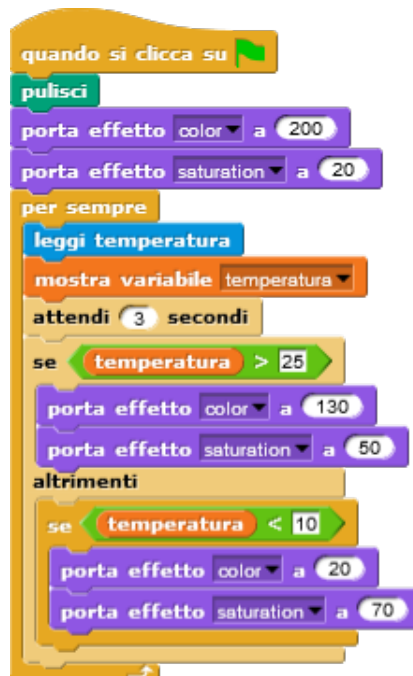


Figura 5.8: Script Storia- Sprite termometro

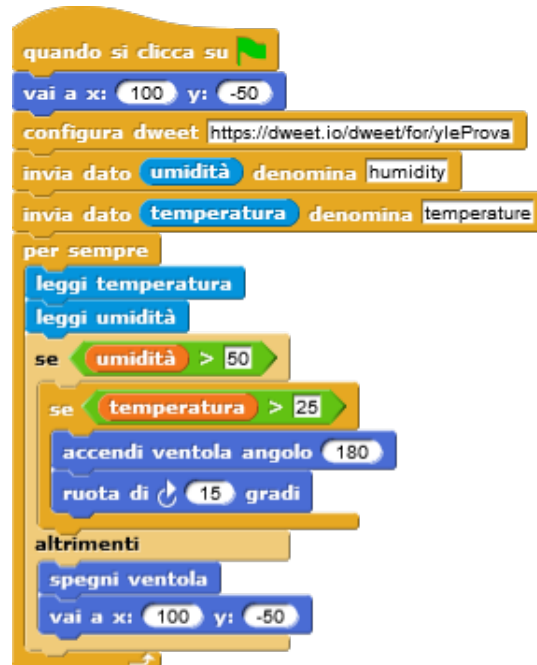


Figura 5.9: Script Storia-Sprite Girandola

Questo micromondo prevede tre principali protagonisti tra cui l'egiziano il quale racconta come gli strumenti si sono evoluti nel tempo e come venivano utilizzati nell'antichità.

Gli altri due attori dello stage sono un termometro ed una girandola o pala eolica.

Il termometro all'inizio dell'esecuzione presenta il "mercurio" colorato di verde; nel corpo dello script, leggendo continuamente il valore della temperatura varia di colore. Se la temperatura supera i 25 gradi allora il colore del mercurio diventerà rosso ad indicare un innalzamento della temperatura esterna. In caso contrario, ovvero quando la temperatura scende sotto i 10 gradi centigradi viene indicata dal colore azzurro a ricordare il ghiaccio e quindi una sensazione di freddo.

L'ultimo attore in scena è la girandola, può anche essere vista come pala eolica, sta alla fantasia di chi la osserva. La girandola, come il termometro, calcola costantemente la temperatura e l'umidità e, se queste superano determinate soglie viene accesa la ventola sulla board e lo sprite viene fatto ruotare a simularne il movimento.

A differenza del termometro, la pala invia i dati alla piattaforma dweet correttamente configurata per poterne graficare l'andamento.



Figura 5.10: Script Storia-Stage

#### 5.4.1.3 Geografia e Musica

**Geografia**, per geografia ci si è ispirati, non prettamente alla didattica ma, ad un aspetto fondamentale per qualsiasi bimbo ovvero la visione di ciò che sta accadendo. Lo script pensato prende in considerazione principalmente tre sensori: temperatura, umidità e co2.

Basandosi sui dati restituiti dai blocchi "leggi temperatura", "leggi umidità" e "leggi co2" sono stati ideati dei paesaggi che ogni alunno può modificare e/o disegnare a proprio piacimento che variano al variare dei valori.

Il paesaggio è composto da uno sfondo disegnato su carta ed importato su Snap!; questo è il primo aspetto fantasioso del micromondo in quanto ogni alunno può realizzare, disegnare e comporre il proprio paesaggio a seconda del luogo su cui vuole lavorare e di quali aspetti meteorologici vuole visualizzare il cambiamento.

In questo caso ho scelto di rappresentare un paesaggio semplice su cui lo script va ad attuare dei cambiamenti sulla base dei valori ottenuti dai sensori sulla board. Sono presenti diversi attori e varie tipologie di costumi che possono indossare. Gli attori in questione sono:

- Sole,
- Nuvole,



- Vento,
- Neve.

Il micromondo non fa altro che modificarsi mostrando o nascondendo gli attori, cambiandogli costume e posizione facendo riferimento ai valori sopra definiti.

Per coinvolgere ancora di più l'alunno nella creazione del micromondo si è pensato di inserire anche la **musica**.

Mettendosi nei panni di un bimbo e pensando al meteo il primo suono che ci viene in mente è quello del vento. Sfruttando il valore ottenuto dall'anemometro, il quale misura la velocità del vento, viene emesso un suono. Il suono è dettato dalle condizioni meteo di quel momento che ne aumentano o diminuiscono l'intensità.

Questa idea è molto stimolante per il bambino in quanto il rumore del vento è un suono che ci accompagna sin dalla nascita, al quale siamo abituati e che sappiamo riconoscere nell'immediato. In questo modo l'alunno è in grado di associare una condizione meteorologica esterna dell'ambiente in cui vive ad una che sta progettando sul pc rendendolo come diceva Papert *creta nelle sue mani*.

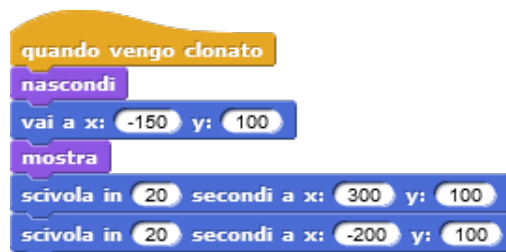


Figura 5.11: Script Geografia-Nubi Script



Figura 5.12: Script Geografia-Sole Script

Il risultato finale del micromondo di geografia è la rappresentazione di un mondo virtuale che si modifica sulla base dei valori rilevati nell'ambiente fisico. Se la temperatura risulta molto alta allora il sole diventa rosso ed inizia a ruotare su se stesso ad indicare l'irradiazione dei raggi solari.

Le nubi, se attraverso una rilevazione rilevano la temperatura bassa ma un valore di umidità molto alta iniziano a muoversi cambiando colore all'aumentare o al diminuire della temperatura.

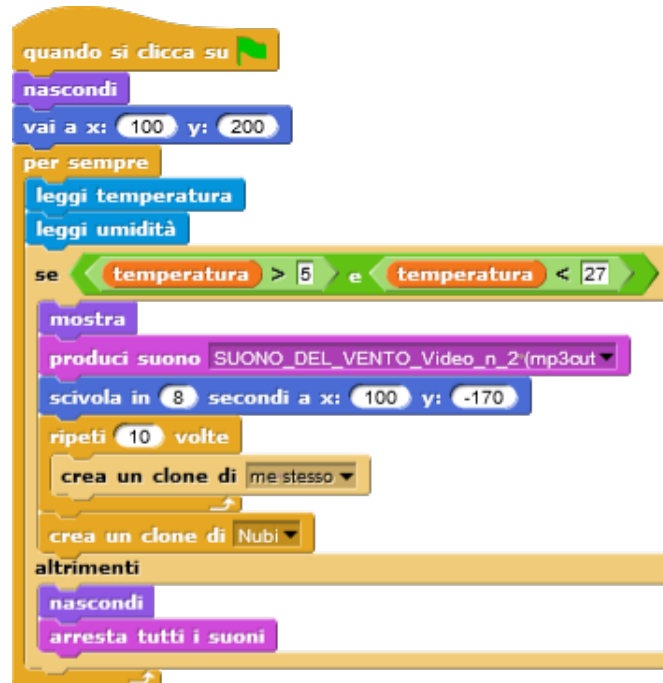


Figura 5.13: Script Geografia-Foglia Script



Figura 5.14: Script Geografia-Neve Script

In questo micromondo viene sfruttata anche la lettura del sensore di pressione atmosferica. Questo valore viene sfruttato dall'attore Neve che, in caso questa si abbassi sotto al valore 1000, ovvero risulti molto bassa, si mostra nella posizione definita e, clonandosi scivola in una posizione random andando a simulare una nevicata sulla nostra città.

Allo stesso modo le foglie simulano l'autunno e, nel momento in cui iniziano a cadere viene riprodotto in sottofondo il suono del vento autunnale per rendere il micromondo ancora più vicino alla realtà.

#### 5.4.1.4 Italiano

**Italiano**; questa non è la materia che più si addice al progetto MIA ma, si può comunque integrare realizzando un micromondo stimolante.

La poesia nella scuola primaria ha uno strano destino; spesso accompagna delle ricorrenze e festività, altre volte gli insegnanti si concentrano sulla produzione di testi da parte dei bambini "creando" piccoli poeti.

L'idea del micromondo è quello di prendere una poesia studiata durante l'anno facendola recitare da uno sprite rendendo il tutto teatrale. Ad accompagnare la poesia, sullo sfondo un paesaggio che si modifica variando i propri colori nel tempo come se anch'esso stesse recitando.

Con questa modalità il bambino sfrutta le proprie conoscenze per realizzare lo spettacolo che preferisce. Ovviamente il risultato non sarà uguale per tutti; ogni alunno racconterà la poesia che meglio ha imparato o che più gli è piaciuta delineando anche lati del proprio carattere e della propria personalità.

Il disegno del paesaggio non è la riproduzione passiva delle percezioni fornite dall'esterno, bensì è un'elaborazione originale della mente del disegnatore, una ricostruzione soggettiva. Il bambino, con linee e figure che conosce, tenta di mettere su carta quello che ha graficamente visto solo dentro di sé.

Inizialmente il piccolo rappresenta se stesso e lo spazio intorno a lui, con una scala di allontanamento/avvicinamento che dipende solo dalla sua affettività.

A seconda di come egli disegna gli oggetti intorno a sé, si possono interpretare i diversi pensieri che gli affollano la mente.

L'esempio realizzato sfrutta la poesia "San Martino" di Giosuè Carducci; questa racconta quanto è malinconico l'autunno con la sua stagione fredda e piovosa, ma nello stesso tempo nell'aria c'è un odore di vino nuovo e nelle case si sente lo scoppiettio dei camini accesi che danno una grande sensazione di pace e gioia per la serenità e felicità che solo le cose vere della vita, cioè i valori, possono dare.

Nella prima strofa Carducci descrive il paesaggio autunnale dove la nebbia copre gli alberi spogli bagnandoli di minuscole goccioline e, a causa del vento maestrale, il mare è agitato e spumeggiante ed infine il rumore delle onde forma delle urla spaventose.

Nella seconda strofa, invece, il poeta mette in risalto la differenza tra la tristezza della natura e la felicità delle persone semplici.

Per realizzare il micromondo ho scelto di utilizzare un paesaggio neutro su cui poi andare a creare lo spettacolo. Per rendere il tutto più creativo ho

deciso di inserire nel micromondo un file audio che recita la poesia con dei suoni di sottofondo che replicano la natura nelle sue sfumature. Sono stati inseriti diversi sprite partendo dal protagonista di nome Mario, le gocce, le nubi e gli stormi di uccelli neri.

Quando viene premuta la bandiera verde, la quale indica l'inizio dell'esecuzione, il protagonista Mario si posiziona sulla sinistra dello schermo ed utilizzando il blocco "dice per" recita la poesia andando a tempo con il suono in sottofondo. Gli altri sprite inizialmente risultano nascosti, ma, a seconda del verso recitato nella poesia si "mostrano" arricchendo il paesaggio di effetti grafici.



Figura 5.15: Script Italiano-Sprite Mario

In questo script prevale il blocco "Dire"; questo blocco permette allo sprite in questione di scrivere o meglio recitare a video la frase che gli viene fornita in input. Il secondo parametro permette di specificare per quanto tempo quest'ultima deve rimanere visibile.

Il blocco rosa "Riproduci suono" è colui che permette di riprodurre in sottofondo la poesia; il formato datogli in input è un mp3 che poi viene mandato in esecuzione per tutta la durata dello spettacolo.

"Invia a tutti" permette di spedire a tutti gli sprite un messaggio, in questo caso "nubi". All'arrivo di questo messaggio poi per ogni sprite verrà deciso il comportamento.

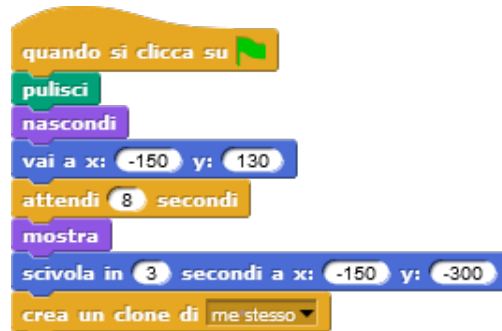


Figura 5.16: Script Italiano-Sprite Goccia



Figura 5.17: Script Italiano-Sprite Nubi



Figura 5.18: Script Italiano-Sprite Stormo

In questi script è stato utilizzato il blocco "Scivola", il quale permette di far spostare lo sprite su cui si sta lavorando in una determinata posizione, identificata da coordinate x e y, per un determinato lasso di tempo definito in input.

Il micromondo di italiano, non prevede l'utilizzo di blocchi per interagire con la board, in quanto, come descritto in precedenza, italiano non è la materia che più si addice al progetto MIA. Nonostante ciò, questo script permette ai bimbi di giocare creando il paesaggio e raccontando la poesia a loro piacimento con forme, colori, sprite e movimenti differenti.

### 5.4.2 Scuola Secondaria di Primo Grado e Scuola Secondaria di Secondo Grado

La scuola secondaria di primo grado e la scuola secondaria di secondo grado Pascal-Comandini di Cesena, rispetto agli altri istituti coinvolti permettono un utilizzo della libreria più matematico. Grazie al programma di studi, gli alunni possono, durante la costruzione dei micromondi, incontrare aspetti di logica più avanzati riuscendo, affiancati dai collaboratori di Craid Coding, a risolverli.

A differenza delle altre scuole la libreria messa a disposizione dei ragazzi delle superiori offre un numero ristretto di blocchi. I blocchi già implementati sono quelli per l'interazione con i sensori e gli attuatori, le configurazioni ed il grafico a torta.

Gli aspetti salienti di logica riguardante l'utilizzo dei dati ottenuti dal sensore vengono lasciati totalmente nelle mani degli studenti per farli ragionare sull'utilizzo dei costrutti più complessi.

Un aspetto ulteriore con cui gli alunni, sia delle medie sia delle superiori, interagiranno saranno le piattaforme di cloud IOT per il mantenimento dei dati monitorati e/o calcolati in un lasso di tempo. La piattaforma principale, come per l'istituto primario è DWEET, la quale permette di raccogliere dati e rappresentarne un grafico lineare nel tempo ma, per consentire loro di customizzare la raccolta e la visualizzazione dei dati è stata inclusa anche freeboard<sup>2</sup>.

Freeboard.io è una dashboard IOT che consente di visualizzare ed elaborare dati provenienti da nodi remoti, nel nostro caso da Snap!.

Con questo tipo di strumento ogni ragazzo può decidere come e soprattutto cosa rendere visibile utilizzando grafici, testi, onde, mappe, ecc.

Per poter graficare i dati su freeboard bisogna passare per dweet. La libreria dispone già dei blocchi "invia dato" ed "invia lista di dati" che tramite una chiamata https inviano rispettivamente un dato singolo o un insieme di dati alla piattaforma. Dweet permette di fare da tramite tra Snap! e freeboard in quanto, una volta ottenuto il dato su dweet lato freeboard basterà scegliere il tipo di widget desiderato e, in fase di creazione associargli il nome del dato presente su dweet utilizzando i datasources. Il meccanismo di utilizzo di freeboard è molto semplice ma, allo stesso tempo permette di creare spazi differenti di memorizzazione e visualizzazione dei dati.

Come dweet anche freeboard.io è configurabile; ogni alunno può creare la sua freeboard differente nei grafici, nei colori e nella visualizzazione partendo dalla configurazione di dweet ed espandendola.

---

<sup>2</sup><https://freeboard.io/>

## 5.5 Possibili ampliamenti

In futuro la libreria Snap! potrà essere ampliata con l'aggiunta di nuovi blocchi per la lettura dei sensori o degli attuatori sulla board. Oltre ai blocchi per la lettura il progetto potrà essere ampliato per consentire ai bambini ed ai ragazzi di sviluppare delle idee riguardanti l'ambiente in cui viviamo.

Un'idea molto valida potrebbe essere quella di realizzare, sfruttando i blocchi esistenti, un micromondo che, basandosi sulle rilevazioni giornaliere sia in grado di prevedere il meteo del giorno successivo o di attuare delle strategie per permettere di abbassare alcuni valori se questi risultassero troppo alti favorendo l'ottimalità della qualità dell'aria nella nostra città.

Per quanto riguarda la stazione meteorologica in futuro sarà possibile l'inserimento di nuovi sensori come il sensore a raggi ultravioletti. Oltre a questi si possono incrementare i sensori e gli attuatori preesistenti aumentando così il potenziale e la quantità di rilevazioni della stazione meteorologica.

Il progetto MIA in futuro potrà sicuramente far parte di progetti più ampi anche a livello nazionale, non sono territoriale che mostrino come la regione Emilia Romagna stia lavorando per portare sempre di più il coding ed il pensiero computazionale nelle scuole di diversi ordini e gradi formando ragazzi che siano "fluenti" con le nuove tecnologie.



# Conclusioni

In questa tesi si presenta il concetto di pensiero computazionale e coding proposti da Seymour Papert padre del costruzionismo, e di come questi stiano modificando la didattica nelle scuole italiane. Il pensiero computazionale fornisce un quadro potente per studiare l'informatica; la sua applicazione riguarda il processo di riconoscimento di aspetti della computazione nel mondo che ci circonda.

Secondo la prof.ssa Jeannette Wing il pensiero computazionale dovrebbe far parte delle competenze di tutti i nuovi studenti universitari. Jeannette Wing sostiene che bisognerebbe diffondere la gioia, i timori e la potenza dell'informatica, con l'obiettivo di rendere popolare il pensiero computazionale sfruttando una metodologia didattica attraverso la programmazione "a blocchi grafici".

Il coding viene associato al making, movimento culturale contemporaneo nato dal mondo del fai da te. Facendo convergere saperi, conoscenze ed idee vengono realizzati degli oggetti, strumenti, materiali originali e innovativi per rispondere a piccole e grandi esigenze.

Il coding associato al making favorisce la capacità di collaborare e comunicare sviluppando il pensiero critico dando vita a un progetto comune tramite la fabbricazione di qualcosa.

Il Progetto MIA, attraverso percorsi guidati in tre scuole del territorio di diverso ordine e grado vuole portare il pensiero computazionale all'interno della didattica permettendo l'utilizzo del computer, non solo come esecutore di algoritmi, ma, come creta nelle mani dei bambini o dei ragazzi con cui poter creare una scultura o più nello specifico poter programmare la propria stazione meteorologica. Attraverso percorsi costruiti, grazie anche al contributo delle insegnanti, sarà possibile incorporare a questa attività i concetti e i principi delle tradizionali discipline scolastiche.

Questa tesi ha posto l'attenzione sui diversi livelli di astrazione utilizzati consentendo a tutti gli studenti di potersi esprimere e poter lavorare nel modo migliore e più conforme alle loro capacità ed alla loro età.

Per il futuro sarà sicuramente possibile ampliare la stazione meteorologica incrementando il numero dei sensori e degli attuatori già presenti o inserendone di nuova costruzione. Un ampliamento futuro potrebbe riguardare le

conoscenze acquisite dai ragazzi o dai bambini e di come queste possano essere sfruttate a livello territoriale realizzando strumenti e/o modelli di purificazione dell'aria nell'ambiente in cui viviamo.

# Ringraziamenti

Desidero ringraziare il professor Alessandro Ricci e la Dott.ssa Laura Tar-  
sitano per essersi sempre resi disponibili e per avermi reso partecipe di un pro-  
getto a livello territoriale dandomi la possibilità di interagire con un metodo di  
lavoro inesplorato potendo crescere sia professionalmente sia personalmente.

Ringrazio inoltre, il professor Andrea Vaccari ed il professor Stefano Ca-  
stagnoli per le delucidazioni riguardanti la stazione meteorologica e per aver  
condiviso con me le idee che hanno portato il progetto MIA a migliorarsi.

Desidero ringraziare i miei compagni che, nel corso di questi tre anni, ci  
sono sempre stati sia dentro sia fuori dall'Università. A loro vorrei solo dire  
grazie per essermi stati vicini, per avermi fatto sorridere quando pensavo di  
non essere all'altezza di questo corso ma, soprattutto grazie per avermi fatto  
crescere.

In ultimo ringrazio la mia famiglia per avermi sempre sostenuto ed aver  
creduto sempre in me.



# Bibliografia

- [1] Papert Seymour, *Mindstorms: Children, computers, and powerful ideas*, Basic Books, Inc., 1980. ix, 8
- [2] Wing Jeannette M, *Computational thinking - Communications of the ACM*, Volume 49, 2006. xi, 4
- [3] Giuseppe Matarazzo, *Avvenire - Con il coding, a scuola del futuro*, <https://www.avvenire.it/agera/pagine/con-il-coding-a-scuola-di-futuro>, Avvenire, 2 Marzo 2019. 15
- [4] Stacie Slotnick, *Mit-Massachusetts Institute of Technology, In memory of Seymour Papert*, <https://www.media.mit.edu/posts/in-memory-seymour-papert/>, 20 Gennaio 2017. 10
- [5] Capponi Massimo, *Un giocattolo per la mente. L'informazione cognitiva di Seymour Papert*, Morlacchi Editore, 2009. 4, 5
- [6] Giunti Scuola, *La Vita Scolastica-I bambini adorano costruire oggetti. Ciao a Seymour Papert*, <https://www.giuntiscuola.it/lavitascolastica/magazine/news/eventi/i-bambini-adorano-costruire-oggetti-ciao-a-seymour-papert/>, Giunti Scuola, 4 Agosto 2016. 9
- [7] Mitch Resnick, *Mitch Resnick: Let's teach kids to code*, <https://www.youtube.com/watch?v=Ok6LbV6bqaE>, 29 Gennaio 2013. ix, 10
- [8] Resnick M, *Learn to code, code to learn. EdSurge*, 2013. 11
- [9] Papert S, *Bambini e adulti a scuola con il computer*, Mediamente Biblioteca Digitale, 1997. ix
- [10] Adafruit, *Adafruit Si7021 Temperature + Humidity Sensor*, <https://cdn-learn.adafruit.com/downloads/pdf/>

- adafruit-si7021-temperature-plus-humidity-sensor.pdf*,  
Adafruit, 21 Novembre 2018. 42
- [11] Adafruit, *Adafruit BMP280 Barometric Pressure + Temperature Sensor Breakout*, <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-bmp280-barometric-pressure-plus-temperature-sensor-breakout.pdf>, Adafruit, 28 Marzo 2019. 42
- [12] Adafruit, *Adafruit SGP30 TVOC/eCO2 Gas Sensor*, <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-sgp30-gas-tvoc-eco2-mox-sensor.pdf>, Adafruit, 05 Aprile 2019. 43
- [13] Adafruit, *Digital universal particle concentration sensor*, [https://cdn-shop.adafruit.com/product-files/3686/plantower-pms5003-manual\\_v2-3.pdf](https://cdn-shop.adafruit.com/product-files/3686/plantower-pms5003-manual_v2-3.pdf), Adafruit, 2016. 45
- [14] Michele Maffucci, *Arduino- lezione 09: uso di LED RGB-parte 1*, <https://www.maffucci.it/2014/09/27/arduino-lezione-09-uso-di-led-rgb-parte-1/>, Arduino, 27 Settembre 2014. 46
- [15] Adrorobot, *Il servomotore*, [https://www.adrirobot.it/servotester/il\\_servomotore.htm](https://www.adrirobot.it/servotester/il_servomotore.htm), Adrorobot, 12 Novembre 2017. 47
- [16] Wikipedia, *Raspberry Pi*, [https://it.wikipedia.org/wiki/Raspberry\\_Pi](https://it.wikipedia.org/wiki/Raspberry_Pi), Wikipedia, 18 Luglio 2019. 40
- [17] Prof. Alessandro Ricci, *Sistemi Embedded basati su Micro-ctrllore*, [https://iol.unibo.it/pluginfile.php/239318/mod\\_resource/content/0/modulo-1.2%20-%20Sistemi%20embedded%20basati%20su%20micro-ctrllore.pdf](https://iol.unibo.it/pluginfile.php/239318/mod_resource/content/0/modulo-1.2%20-%20Sistemi%20embedded%20basati%20su%20micro-ctrllore.pdf), 2018/2019. 40
- [18] Prof. Alessandro Ricci, *Introduzione ai Sistemi Embedded*, [https://iol.unibo.it/pluginfile.php/239319/mod\\_resource/content/0/modulo-1.1%20-%20Introduzione%20ai%20Sistemi%20Embedded.pdf](https://iol.unibo.it/pluginfile.php/239319/mod_resource/content/0/modulo-1.1%20-%20Introduzione%20ai%20Sistemi%20Embedded.pdf), 2018/2019. 40
- [19] Harvey Brian and Mönig Jens, *Snap! reference manual*, <http://snap.berkeley.edu/SnapManual.pdf>, Sanp!, 2017. 53
- [20] Papert Seymour, *The Children's Machine. Rethinking School in the Age of the Computer*, Basic Books, Inc., 1993. ix, 6, 8

- [21] Reggini H.C., *Alas para la mente*, Mondadori tr.it., 1984. 15
- [22] Parisi Domenico, *Le simulazioni e la storia*, TD-Tecnologie Didattiche, 2001.