



UNIVERSITÀ DEGLI STUDI DI TRENTO

Dipartimento di Matematica

Laboratory of Computer Science Education

GPS
elaborare e rappresentare
dati cartografici

Professore:

Prof. Alberto Montresor

Studenti:

Vittorio Altafini

Giada Tormen

Luca Zanini

Giugno 2020



Quest'opera è stata rilasciata con licenza Creative Commons Attribuzione - Non commerciale - Condividi allo stesso modo 3.0 Italia. Per leggere una copia della licenza visita il sito web <http://creativecommons.org/licenses/by-nc-sa/3.0/it/>

INDICE

1	Introduzione	1
1.1	Tempistiche	1
1.2	Valutazione	2
2	Strategie didattiche	3
2.1	Architettura recettiva (trasmissiva)	4
2.2	Architettura collaborativa	4
2.2.1	Apprendimento cooperativo	4
2.3	Architettura esplorativa	5
2.3.1	Problem Based Learning	5
2.3.2	Metodo dei progetti	6
2.4	Architettura metacognitivo-autoregolativa	6
2.4.1	Supporto alla metacognizione e all'autoregolamento	6
2.5	Percorso a spirale	7
3	Prerequisiti e obiettivi	8
4	Organizzazione	9
5	Teoria e raccolta dati	11
5.1	Lezione in aula	11
5.2	Raccolta dati	11
6	Elaborazione dati con Python	13
6.1	Estrapolare liste da file .gpx	13
6.2	Passare a un sistema di coordinate cartesiane	15
6.3	Calcolo della distanza, tempo e velocità media	17
6.4	Plot dei dati	18
7	Applicazioni sociali	22
7.1	Svolgimento	22
8	Conclusioni e possibili sviluppi	25
8.1	L'uso del software QGis	25
8.2	Disegnare sulle mappe con la bicicletta	26
8.3	La vita segreta dei gatti	27

Capitolo 1

Introduzione

Il presente lavoro raccoglie una serie di istruzioni per spiegare esaurientemente tutte le procedure presenti nella parte riguardante la scuola secondaria di secondo grado del progetto riguardante il GPS da noi sviluppato per il corso di Laboratory of Computer Science Education.

Il progetto è stato sviluppato insieme ad un altro gruppo, il cui compito consisteva nel curare la parte riguardante la scuola secondaria di primo grado. Il filo conduttore che abbiamo utilizzato per unire le due parti è rappresentato dall'argomento GPS. La particolarità del progetto risiede nel diverso sviluppo delle attività legate a tale argomento in ambienti differenti. Il GPS rappresenta un'occasione per gli insegnanti di trattare un argomento in maniera trasversale, coinvolgendo diverse materie ed argomenti. Inoltre, con temi di questo tipo, più immersi nella realtà di tutti i giorni, sarà più semplice coinvolgere gli studenti. Il documento si aprirà con una parte teorica riguardante le metodologie didattiche per esplicitare le strategie utilizzate per costruire il progetto. Successivamente passeremo ad elencare i prerequisiti e gli obiettivi che ci siamo prefissati. Elencheremo, quindi, le procedure da seguire, spiegate in dettaglio. Termineremo sottolineando quali possano essere eventuali sviluppi futuri.

Il progetto descritto dal documento si pone un ulteriore obiettivo: si vogliono sviluppare COMPETENZE TRASVERSALI negli studenti, ovvero una serie di abilità riutilizzabili in diversi ambiti, che mirano a renderli più consapevoli delle loro capacità e li aiutano a gestirle con efficienza. Nel capitolo seguente, tra le diverse strategie, ne vedremo una focalizzata su tali competenze.

1.1 Tempistiche

L'organizzazione delle tempistiche è un aspetto delicato che richiede un'attenta riflessione. In questa sede siamo consapevoli che la nostra è una proposta indicativa che va adattata al tipo di classe che si ha a disposizione, nonché ripensata dopo essere applicato la prima volta.

Prevediamo 3 blocchi distinti di ore da fare in classe (non comprendono il lavoro da fare a casa) che comprendono:

1. Teoria e raccolta dati (2h + uscita scolastica di più giorni o in giornata)
2. Elaborazione dati con Python (6h)
3. Sviluppo App (2h per la teoria di MIT App Inventor + 8h di sviluppo vero e proprio)

1.2 Valutazione

“Non si apprende per essere valutati, ma si valuta per apprendere” Langè, G. & Cinganotto [2013]. Questa affermazione di buon senso è da tenere bene a mente soprattutto quando si vuole valutare un percorso didattico fortemente laboratoriale. Nell’ottica della didattica per competenze in linea anche con il Piano Nazionale Scuola Digitale (2015) è fondamentale spostare l’attenzione dalla valutazione dei risultati alla valutazione del **processo** che porta a quei risultati. È fondamentale avere un’idea chiara di quali competenze si vuol sviluppare con un compito o un progetto come in questo caso. In base a tali competenze si determinano i criteri di valutazione che devono essere esplicitati con gli studenti. In questo modo si instaura un circolo virtuoso in cui lo studente diventa più consapevole e sa dove migliorare e allo stesso tempo l’insegnante ha un ritorno più preciso riguardo l’efficacia del suo lavoro (Langè, G. & Cinganotto [2013]). A tal proposito la valutazione del progetto che svilupperanno gli studenti è fortemente messa a disposizione dell’apprendimento dello studente come si constaterà meglio nel capitolo 4 del presente lavoro in cui si costruiranno dei test di autovalutazione. Concludiamo questa parte riportando la tabella 1.1 presa da Langè, G. & Cinganotto [2013] che chiarifica meglio cosa si intende con valutazione *per* l’apprendimento differenziandola dalla valutazione *dell*’apprendimento.

Valutazione dell’apprendimento	Valutazione per l’apprendimento
È separata dal processo di apprendimento	È integrata nel processo di apprendimento
Si concentra sulla preparazione alle prove	Si concentra sul capire
Prende in considerazione i risultati	Prende in considerazione i processi
È implicita negli scopi, nei criteri e nei traguardi	È esplicita negli scopi, nei criteri e nei traguardi
Si comunica con un punteggio, un giudizio, un voto	Si realizza come dialogo tra insegnante e studenti
Non ricade sull’apprendimento	Ricade sull’apprendimento
Segue un’ottica di deficit (pone l’accento sugli errori)	Segue un’ottica di potenziamento (tiene conto di ciò che lo studente sa/sa fare e delle potenzialità)

Tabella 1.1: Valutazione *dell*’apprendimento *vs* valutazione *per* l’apprendimento da Langè, G. & Cinganotto [2013]

Capitolo 2

Strategie didattiche

Nel costruire il percorso didattico abbiamo prestato particolare attenzione alla scelta della strategia più adatta a questo contesto di apprendimento. Alla luce delle riflessioni fatte durante il corso, la teoria dell'apprendimento che accompagna la nostra proposta è costituita dal Costruzionismo teorizzato da Seymour Papert attorno agli anni 80 del '900. Papert, che in precedenza aveva lavorato con Jean Piaget, il padre del Costruttivismo, sviluppò una teoria che assunse una propria autonomia e indipendenza. Piaget, infatti, si concentrò maggiormente su una teoria della conoscenza che vede quest'ultima svilupparsi da situazioni pratiche e via via viene "accomodata" dallo studente assumendo un livello di astrazione più elevato, mentre Papert diede maggior rilievo all'atto pratico della costruzione della conoscenza derivante da situazioni didattiche significative come scrive Kafai Y. B. nel capitolo 3 tratto da Sawyer [2005].

Il concetto viene ribadito da Ackermann [2001] il quale osserva che da una parte Piaget preferisce descrivere l'accomodamento della conoscenza come il raggiungimento di una conoscenza stabile, dall'altra Papert si interessa alla dinamica del cambiamento in cui si mettono in discussione gli schemi precostituiti e si costruisce una nuova concezione del mondo. In questo senso il famoso linguaggio di programmazione Logo, sviluppato da Papert, è paradigmatico in quanto esso concretizza "l'oggetto tramite il quale lo studente pensa".

L'informatica come disciplina di insegnamento, e più in generale il pensiero computazionale, si configura quindi come un ambiente ideale per cogliere occasioni di costruzione della conoscenza per mezzo di un linguaggio di programmazione che permette di interpretare un fenomeno spezzettandolo in algoritmi semplici che stimolano l'astrazione e la eseguono in modo meccanico (Lodi et al. [2017]).

Consapevoli quindi che l'informatica si presta bene ad una didattica laboratoriale, abbiamo individuato quali possano essere le strategie didattiche che maggiormente rispondono alle nostre esigenze.

Architettura	Strategia didattica
Recettiva (trasmissiva)	Esposizione classica
	Esposizione multimodale
Comportamentale (direttivo-interattiva)	Istruzione sequenziale interattiva
	Modellamento (apprendistato)
	Supporto al comportamento positivo
Simulativa	Studio del caso
	Simulazione simbolica
	Game Based Learning
	Role playing/drammatizzazione
Collaborativa	Mutuo insegnamento
	Apprendimento cooperativo
	Discussione
Esplorativa	Problem Based Learning
	Metodo dei progetti
Metacognitivo-autoregolativa (strategie per autoapprendere)	Metacognizione e autoregolazione

Tabella 2.1: Architetture dell'istruzione e strategie didattiche tratto da Bonaiuti [2014]

Aiutandoci con lo schema proposto da Bonaiuti [2014] e riportato nella tabella 2.1 si può notare come dall'alto verso il basso lo studente diventi sempre di più il protagonista dell'apprendimento, e la conoscenza venga meno prestrutturata. Si passa quindi da un'architettura trasmissiva, in cui è l'insegnante ad organizzare il sapere esponendolo nel modo migliore possibile, fino ad arrivare ad un'architettura esplorativa, dove lo studente, in un contesto laboratoriale ben definito dall'insegnante, impara e costruisce la propria conoscenza in autonomia o all'interno del gruppo. L'ultima architettura si pone ad un livello più alto in cui si persegue l'obiettivo più generale dell'imparare ad imparare.

In questa situazione storica, in Italia, in linea con le indicazioni del Piano Nazionale Scuola Digitale sviluppato dalla Legge 107/2015 (Buona scuola), l'insegnamento dell'informatica si adatta bene a questo tipo di didattica in cui l'uso degli strumenti digitali facilitano lo studente a ragionare sul proprio processo di apprendimento.

2.1 Architettura recettiva (trasmissiva)

Riteniamo fondamentale spendere alcune righe parlando dell'esposizione classica in cui l'insegnante spiega di fronte a una classe di studenti che prendono appunti e ascoltano. Questa modalità anche se è stata messa in discussione e criticata per la scarsa efficacia rispetto ad altre strategie è ancora la strategia più diffusa in Italia. In questo periodo di pandemia è risultata particolarmente utile per raggiungere gli studenti nelle proprie case tramite videolezioni, e in particolare mantenere l'attenzione dello studente è diventata una sfida ancora più grande.

Anche nel nostro percorso, viene usata per spiegare la parte teorica relativa al GPS e come introduzione alla parte laboratoriale. Per noi aspiranti insegnanti è utile sapere i punti di forza e debolezza di tale pratica per essere più consapevoli e migliorarci continuamente. È importante innanzitutto attuare un'attenta pianificazione a partire dall'individuazione degli elementi fondamentali da far apprendere nel tempo a disposizione, dei passaggi logici e narrativi da seguire. Particolare valenza ha la fase di avvio in cui è utile richiamare l'interesse sull'obiettivo e sul percorso che verrà seguito per raggiungerlo. Nella costruzione del discorso sono da evitare: frasi molto lunghe e complesse; le proposizioni subordinate; gli incisi e le parentesi; le divagazioni; le digressioni; le doppie negazioni; l'uso eccessivo di slide e di apporti multimediali. I concetti principali devono essere costantemente evidenziati e richiamati. Gli insegnanti efficaci sono capaci di aiutare gli studenti a strutturare le informazioni in modelli e schemi Bonaiuti [2014].

2.2 Architettura collaborativa

L'architettura collaborativa mette in pratica i principi del socio-costruttivismo di Vygotskij, nonché le idee di apprendimento situato di Papert [2002]. Diversi studiosi hanno incominciato a sostenere che la conoscenza è il frutto di un processo sociale, ovvero si forma quando gli individui partecipano alle pratiche culturali. Tale architettura è da considerarsi trasversale a tutte le altre, perché non è possibile pensare di costruire una strategia pedagogica che possa fare a meno di una qualche forma di interazione con gli altri.

Tra le strategie individuate in questa architettura abbiamo scelto di adottare per il nostro progetto l'apprendimento cooperativo.

2.2.1 Apprendimento cooperativo

L'apprendimento cooperativo è una strategia collaborativa che mira a sviluppare l'apprendimento individuale come conseguenza di attività svolte in gruppi. Le modalità di lavoro e le dimensioni dei gruppi possono essere diverse: i gruppi possono essere fissi o rimescolati ciclicamente, i membri possono avere o meno dei ruoli, la valutazione può essere di gruppo e/o individuale, i gruppi possono competere tra loro oppure no. Diversi sono gli elementi che caratterizzano questa strategia, alcuni esempi sono la dipendenza di ciascuno studente dal lavoro degli altri, la responsabilità individuale dello studente per la parte di lavoro assegnatagli o la parità di partecipazione. Il professore, dal canto suo, deve porre attenzione a diversi aspetti. È importante fissare degli obiettivi di apprendimento ben definiti in maniera tale da evitare incomprensioni. La formazione dei gruppi è un aspetto da non sottovalutare, dunque si deve porre attenzione all'eterogeneità nella composizione del gruppo e mantenere un ridotto numero di partecipanti

per ogni gruppo poiché in gruppi troppo grandi alcuni studenti potrebbero rischiare di non lavorare al meglio. È inoltre necessario un controllo e una valutazione dell'attività svolta.

Ciò che differenzia questa strategia da altre tipologie di lavoro di gruppo sta nella dipendenza dei componenti del gruppo l'uno dall'altro. Tale tipo di dipendenza viene chiamata interdipendenza positiva. Può essere raggiunta assegnando a ogni soggetto una funzione o un ruolo stabili oppure prevedendo una rotazione dei compiti all'interno del gruppo nelle varie fasi di lavoro. Comunque è importante che i compiti non siano distribuiti in maniera casuale così da evitare che solo alcuni membri si facciano carico dei compiti ritenuti più pesanti. Tale strategia, come tutte le altre, presenta alcune insidie che il docente può trovarsi ad affrontare. Come abbiamo già sostenuto, il docente deve fare attenzione alla formazione dei gruppi, che devono essere equilibrati e non troppo grandi. Il docente non deve presupporre che sia sufficiente formare i gruppi ed assegnare i diversi compiti: il lavoro va infatti attentamente organizzato e monitorato dato che non si può presupporre che venga sempre svolto in maniera adeguata. Infine, uno degli aspetti più critici è il fattore tempo. L'apprendimento cooperativo, infatti, non richiede un impegno solo sul lavoro che il professore assegna ma consuma energie anche sul versante relazionale e organizzativo con il rischio di una dilatazione dei tempi di lavoro. Al netto di questi rischi, rimane indubbia la potenza di questa strategia, tanto che quelle che andremo ad elencare successivamente non ne possono prescindere. Sono evidenti le diverse caratteristiche costruzioniste come il lavoro in gruppi o la condivisione delle proprie conoscenze con gli altri.

2.3 Architettura esplorativa

Questo tipo di architettura la possiamo trovare nella 2.1 sotto a quella collaborativa, dunque presenterà minor pre-strutturazione della conoscenza e maggior libertà per lo studente. Essa si basa sull'idea che l'apprendimento si sviluppi efficacemente se agli studenti vengono posti problemi da risolvere sul quale essi possono riflettere liberamente. Lo stimolo derivante dall'esigenza di dare risposta ad un problema aperto rappresenta, in effetti, un potente stimolo all'azione. In questa prospettiva il ruolo e insegnante è soprattutto quello del catalizzatore dell'interesse a conoscere, ad esempio ideando occasioni capaci di incitare l'impegno diretto.

Abbiamo individuato, in linea con Bonaiuti [2014], due principali strategie attorno alle quali si è focalizzata la nostra attenzione: il Problem Based Learning (PBL) e il metodo dei progetti.

2.3.1 Problem Based Learning

Questa strategia, la cui base è la visione dello studente come protagonista delle lezioni, mira a sviluppare l'apprendimento mediante l'investigazione e la riflessione, stimolati dal dover risolvere problemi posti dal docente. I problemi possono essere di diversa natura e possono presentarsi nelle forme più diverse: storie, problemi logici, algoritmi, ecc. . .

Gli studenti possono lavorare singolarmente ma è più funzionale una divisione in gruppi, in maniera tale da avere collaborazione e per stimolare gli studenti più in difficoltà al ragionamento. I gruppi avranno una grande libertà, ma non si potrà prescindere dal far attenzione a rispettare delle tempistiche stabilite dal docente. Uno dei compiti di quest'ultimo sarà fornire ai propri allievi gli strumenti necessari per poter affrontare un'esperienza di questo tipo. Le attività di apprendimento, per coinvolgere maggiormente gli studenti, saranno pensate per riprodurre situazioni realistiche, in modo tale che, benché rimangano comunque legate alle discipline scolastiche, risultino essere più vicine alla sensibilità dei ragazzi. In questa maniera sarà possibile uscire dalle mura scolastiche e svolgere questo tipo di lezione in luoghi quali i musei o nell'ambito di gite scolastiche. Attraverso queste attività di apprendimento si presuppone che gli studenti possano sia apprendere il funzionamento di certi processi, così da evitare lo sviluppo di una conoscenza nozionistica, sia sviluppare una certa capacità di problem solving. In questa prospettiva il docente è chiamato a rendere possibile l'apprendimento strutturando l'esperienza e facilitando il lavoro attraverso il sostegno, la guida e il monitoraggio del processo di apprendimento stesso.

Anche in questo caso non mancano difficoltà a cui si può andare incontro. I problemi posti, e le modalità di lavoro per la loro risoluzione, devono però essere commisurati con le capacità degli studenti. È necessaria una graduale progressione nella proposta dei problemi da risolvere e nei gradi di libertà lasciati agli allievi. Si deve, quindi, passare da problemi facili mostrando le procedure di risoluzione e guidando attentamente la pratica, all'aumentare gradualmente la difficoltà dei problemi diminuendo sempre più il supporto col

crescere delle abilità degli allievi. Il nostro progetto si basa molto su questa strategia. Gli studenti saranno, infatti, esposti spesso a quesiti che mireranno a stimolarne il ragionamento.

2.3.2 Metodo dei progetti

La strategia didattica del metodo dei progetti dà agli studenti la possibilità di elaborare con un certa libertà una risposta ad un assegnamento dato dal professore. La caratteristica alla base di questa strategia sta nel produrre un progetto ben strutturato partendo da una semplice idea malamente abbozzata, nata in risposta alla richiesta del docente. Agli studenti può essere data la possibilità di lavorare individualmente o a gruppi, ma nel nostro caso, coerentemente con le altre strategie adottate, si opterà per la divisione in piccoli gruppi. La natura dei progetti, invece, può essere di tipologie diverse a seconda dell'argomento, degli scopi, del contesto, nonché delle caratteristiche dei soggetti coinvolti. I risultati di un progetto possono essere documenti, come una relazione, oppure oggetti materiali come un cartellone o multimediali come un video. Come nel caso del Problem Based Learning il progetto può svolgersi in ambienti esterni alle mura scolastiche e arrivare ad assumere caratteri di concretezza, portando perfino a ricadute concrete sul territorio. In ambito scolastico la didattica per progetti ha tipicamente uno scopo conoscitivo (progetto di ricerca) e richiede lo sviluppo di un lavoro articolato in fasi e la presenza di verifiche intermedie. Il metodo dei progetti è connesso ad altre strategie pedagogiche, ponendosi però come percorso di formazione unitario, finalizzato a interconnettere organicamente in un prodotto originale le esperienze maturate e le conoscenze acquisite. Questa strategia è quella che meglio descrive la struttura dell'unità didattica da noi elaborata.

2.4 Architettura metacognitivo-autoregolativa

Questa è l'ultima architettura che possiamo ritrovare nella 2.1, dunque è quella che presenta il maggior grado di libertà dato allo studente. L'attenzione del docente si sposta dall'insegnare dei dati contenuti all'insegnare dei metodi di apprendimento dei contenuti. Lo studente diventa l'assoluto protagonista, mentre l'insegnante assume un ruolo di supporto. L'obiettivo del docente diventa l'irrobustimento di competenze nello studente quali l'autovalutazione dei propri progressi e del livello raggiunto.

2.4.1 Supporto alla metacognizione e all'autoregolamento

Tale strategia mira a rendere gli studenti sufficientemente autonomi ed efficienti nello studio e nell'apprendimento, mettendo a loro disposizione una serie di metodi di apprendimento e mirando a renderli consapevoli degli strumenti di cui dispongono. A tal proposito, risulterà fondamentale lavorare sia sul piano delle strategie cognitive sia su quello delle strategie metacognitive degli studenti, ovvero della loro consapevolezza sulle strategie cognitive migliori in una data situazione e come applicarle al meglio. Le prime si possono dividere in generali, ovvero quelle applicabili in qualsiasi situazione di apprendimento, e specifiche, ovvero quelle applicabile solo in alcune situazioni. Ciò che viene suggerito da questa strategia è di esplicitare le modalità di organizzazione e gestione delle conoscenze. È infatti vero che se alcuni studenti sviluppano autonomamente competenze cognitive, altri (se non la maggior parte) non ci riescono e necessitano di maggior supporto. Negli studenti dovrà, dunque, essere favorito lo sviluppo di conoscenze circa le proprie personali abilità cognitive come pure la capacità di regolare attività generali, come ad esempio il valutare i risultati. I maggiori beneficiari di questo tipo di strategia sono studenti con bisogni educativi speciali o difficoltà d'apprendimento. Nel nostro progetto tale strategia si concretizza con i pre e post test metacognitivi, per mezzo dei quali gli studenti hanno la possibilità di autovalutare le proprie conoscenze e di monitorare i propri progressi.

2.5 Percorso a spirale

In questa strategia il docente affronta un argomento più volte in un certo arco di tempo aumentando progressivamente la profondità della spiegazione. L'idea alla base di questa strategia è evitare di affrontare di petto un concetto che solitamente è di difficile comprensione per gli studenti, partendo con l'introdurlo semplicemente per creare dimestichezza con i termini utilizzati. L'insegnante approfitta di certe lezioni per discutere dell'argomento al centro del percorso a spirale, a cui quello delle diverse lezioni è strettamente legato. La strategia prende il nome dalla maniera in cui il percorso è strutturato: la spirale rappresenta il modo di riaffrontare lo stesso concetto ma con profondità sempre maggiori, fino a raggiungere il centro rappresentato dalla lezione riguardante l'argomento su cui si è costruito il percorso didattico.

Capitolo 3

Prerequisiti e obiettivi

Riferendoci alle Indicazioni Nazionali per i licei (MIUR [2010]) abbiamo deciso di ambientare il nostro progetto in una classe del secondo biennio, terzo o quarto anno, di un liceo scientifico con indirizzo scienze applicate. La nostra scelta è legata ad una serie di argomenti che riteniamo essere prerequisiti necessari per lo svolgimento del nostro progetto. Ciò, però, non esclude che tali argomenti possano essere stati trattati in altre scuole in anni differenti da quelli da noi indicati. Facciamo di seguito un elenco di tali argomenti e della materie a cui sono legati:

- **INFORMATICA**
I costrutti if, for e while
I dati strutturati – liste e dizionari
- **FISICA**
Legge Oraria
Grafico Velocità-Tempo
Sistemi di Riferimento
- **MATEMATICA**
Derivate
Piano Cartesiano
Trigonometria

Abbiamo utilizzato le Indicazioni Nazionali (MIUR [2010]) anche per scegliere l'obiettivo che il nostro progetto mira a realizzare. Tra le diverse sezioni presenti nelle indicazioni, infatti, abbiamo trovato anche gli obiettivi fissati per le materie di una data scuola. Oltre ad obiettivi più classici, come l'acquisizione delle basi teoriche, ne abbiamo trovato uno in particolare che ha attirato la nostra attenzione: sviluppare le competenze di acquisizione, elaborazione e visualizzazione dati. Dato che tale obiettivo viene posto al termine del percorso didattico del secondo biennio, abbiamo deciso di fissare come obiettivo di questo progetto il valutare se e come tali competenze si stanno sviluppando nei nostri studenti.

Capitolo 4

Organizzazione

Descriviamo di seguito come abbiamo organizzato il progetto. Abbiamo deciso di suddividere l'esperienza in tre fasi:

- **LEZIONE IN AULA**
Questa fase è dedicata alla lezione teorica focalizzata sul GPS e si svolgerà in aula;
- **RACCOLTA DATI**
Tale fase è ambientata all'esterno delle mura scolastiche e pone gli studenti di fronte a problemi più vicini alla realtà quotidiana;
- **ELABORAZIONE DATI**
L'ultima fase consiste nell'utilizzare i dati raccolti nella parte precedente ed elaborarli per mezzo dell'utilizzo di strumenti informatici;

Oltre a queste tre fasi ne abbiamo aggiunta una quarta che abbiamo chiamato **APPLICAZIONI SOCIALI**. Tale fase è pensata a parte rispetto alle precedenti e ha come scopo quello di far produrre agli allievi un prodotto concreto e tangibile, così da render loro evidenti i progressi fatti e il livello raggiunto.

A tal proposito, come strumento metacognitivo, abbiamo previsto nel nostro progetto un pre-test successivo alla lezione in aula ed un post-test da sottoporre agli studenti alla fine dell'esperienza. Tali strumenti vengono utilizzati anche per comprendere quali sono le misconcezioni più comuni e se, alla fine dell'esperienza, gli argomenti trattati sono stati pienamente compresi. Dunque, gli obiettivi che ci poniamo di realizzare con l'aiuto di tali test sono:

- Individuare all'inizio dell'esperienza le misconcezioni
- Valutare l'eventuale persistenza delle misconoscenze al termine
- Autovalutazione delle proprie conoscenze da parte dello studente
- Controllo dello studente dell'avanzamento del proprio apprendimento

Una caratteristica imprescindibile di un test di questo tipo è l'anonimato: lo studente non deve aver paura di rispondere ad una domanda e tale caratteristica gli permette di dare risposte sincere. I test possono essere di diverso tipo, a seconda degli scopi che il docente si pone. Nel caso del nostro progetto abbiamo optato per test a risposta multipla. Tale tipologia presenta diverse insidie (ad esempio lo studente può dare risposte casuali senza ragionare), ma è funzionale per realizzare un obiettivo che ci siamo posti: individuare le misconoscenze. Tra le risposte, infatti, oltre a quella giusta, il docente può inserire le misconoscenze che ritiene essere più comuni per quanto riguarda quel dato argomento. Inoltre, ripetendo quest'esperienza per diversi anni, può aggiornare il test con nuove misconoscenze rilevate negli anni precedenti. Per quanto riguarda la metacognizione, saranno presenti due parti ad essa dedicate. All'inizio del test vi sarà uno spazio dedicato alla data sia nel pre che nel post-test, in questa maniera si potrà rendere conto del lasso temporale in cui le sue conoscenze sono progredite. Nella parte finale, invece, lo studente, in entrambi i test, avrà la possibilità di autovalutarsi e gli verrà chiesto di confrontare i risultati del pre e del post-test, così da rendergli evidente i suoi progressi.

I test possono essere creati su siti appositi, ma vi è anche la possibilità di farli cartacei. Noi abbiamo scelto di utilizzare siti come Socrative o Google Moduli, ma la scelta è molto ampia.



Figura 4.1: I due siti utilizzati per la creazione dei test

I test di questo tipo sono materiale versatile, applicabile con diverse materie e con problemi differenti. Sono utili sia al professore per valutare lo stato delle conoscenze dei suoi allievi, sia agli studenti per rendersi conto del proprio livello.

Capitolo 5

Teoria e raccolta dati

5.1 Lezione in aula

Questa fase del progetto riguarda la lezione introduttiva teorica focalizzata sul filo conduttore tra il progetto della scuola secondaria di primo grado e la scuola secondaria di secondo grado, ovvero il GPS. Durante la lezione, che si svolgerà totalmente in aula, verrà introdotto questo argomento, spiegando di cosa si tratta e come funziona. Verranno inoltre trattati gli errori a cui questo strumento è soggetto maggiormente. I tipi di errore sono differenti e tra questi troviamo gli **ERRORI DI TIMING**, i quali ci permettono di costruire un percorso didattico particolare ovvero il percorso a spirale. Gli errori di timing sono legati, infatti, ad un argomento complesso, la relatività, che i nostri studenti non hanno ancora affrontato e non possiedono nemmeno le basi per poterlo fare. L'idea, come abbiamo già spiegato, è quella di introdurre tale argomento per creare dimestichezza con i concetti che lo compongono, non entrando troppo nel merito della matematica sottostante. Dunque, tra gli errori di timing troviamo l'errore nella misura causato dal moto del satellite rispetto alla terra e quello causato dall'effetto redshift gravitazionale. Il primo errore è legato alla relatività ristretta e dipende dal fatto che i sistemi del satellite e della terra non sono inerziali tra loro. Per mezzo della formula $\Delta t' = \frac{\Delta t}{\sqrt{1 - \frac{v^2}{c^2}}}$, della quale verranno spiegati i diversi

termini che la compongono, si arriverà a mostrare che il tempo sul satellite scorre più lentamente che sulla terra. Il secondo errore, invece, è legato alla relatività generale e dipende dal fatto che corpi di una certa massa deformano il tessuto spazio-temporale causando dilatazioni temporali. Il satellite, che si trova più lontano dalla terra, subirà di meno questo effetto di dilatazione. Tutto ciò risulterà in un acceleramento del tempo sul satellite. Per terminare questa parte, procediamo sommando il ritardo calcolato col primo errore all'anticipo visto nel secondo e concluderemo che il tempo sul satellite scorre più velocemente. Questo errore, calcolabile, una volta individuato può essere tranquillamente prevenuto.

5.2 Raccolta dati

Questa è la fase con più probabilità di coinvolgimento degli studenti, ma anche la più difficile da gestire. La raccolta dati può essere effettuata in diversi ambiti, con o senza la presenza del docente. Per il nostro progetto abbiamo ritenuto che una gita scolastica fosse il momento più adatto per attuare questa fase, in modo tale da essere presenti (lasciando una certa libertà agli studenti ma essendo presenti in caso di bisogno) e da uscire dal normale ambito scolastico. In questa occasione si procederà a dividere i ragazzi in una serie di piccoli gruppi, studiati in modo tale da non essere squilibrati. Ogni gruppo utilizzerà un'applicazione scaricata sul cellulare per raccogliere i dati necessari alla fase successiva. Esistono molte applicazioni GPS e la scelta può dipendere dall'obiettivo del progetto. Nel nostro caso l'applicazione che abbiamo selezionato si chiama Locus Map, scaricabile dal Play Store e di cui alla fine della sezione possiamo vedere il logo. La nostra scelta è dipesa dal fatto che, oltre a fornire dati sulla longitudine, latitudine e altitudine, tale applicazione dà la possibilità di variare parametri di acquisizione quali l'intervallo di tempo, la distanza percorsa e il dislivello. Una difficoltà che abbiamo riscontrato risiede nel fatto che non sia facile trovare applicazioni sia per i cellulari con sistema operativo Android sia per quelli con il sistema operativo iOS. Non escludiamo la possibilità che applicazioni simili esistano, ma non è

assicurato che possiedano le caratteristiche richieste dal progetto. La nostra soluzione è stata, dunque, selezionare un'applicazione per il sistema Android dato che i dispositivi con tale sistema operativo sono più diffusi dei dispositivi Apple. Pertanto, i docenti, nel formare i gruppi, dovranno tener conto anche di ciò, mettendo in ogni gruppo almeno un ragazzo con dispositivo con sistema operativo Android.

Arrivati nel luogo della gita verranno stabilite delle tappe nelle quali, per mezzo dell'applicazione, verranno raccolti i dati, ovvero, a seconda del luogo e dell'attività, la lunghezza di un percorso compiuto e la velocità con cui si è proceduti, il perimetro di un monumento/edificio visitato e la relativa altezza, ecc. . .

La presenza di più docenti è consigliata, così da avere la possibilità di dividersi in più gruppi che raccolgono dati differenti. La meta della gita è indifferente, benché certi luoghi possiedano più spunti per un progetto rispetto ad altri. In questa fase i ragazzi avranno la possibilità di operare concretamente, avendo a che fare con la realtà al di fuori della scuola e vivendo l'esperienza con maggior leggerezza rispetto allo stare in aula.



Figura 5.1: L'immagine rappresenta il logo dell'applicazione selezionata per la raccolta dati

Nell'esempio che mostriamo in questo lavoro andremo alla scoperta degli alberi monumentali del Veneto. Al link <https://www.regione.veneto.it/web/economia-e-sviluppo-montano/alberi-monumentali> è possibile accedere alla mappa degli alberi monumentali del Veneto (si veda figura 5.2). Tale mappa è anche scaricabile in formato KML e apribile sul cellulare tramite Locus Map, che provvede a importarlo come insieme di punti geolocalizzati. Gli studenti divisi in gruppi registreranno i tracciati alla scoperta degli alberi, ricordando di salvare le posizioni dei punti di inizio e fine percorso (in questo caso tra un albero e l'altro).

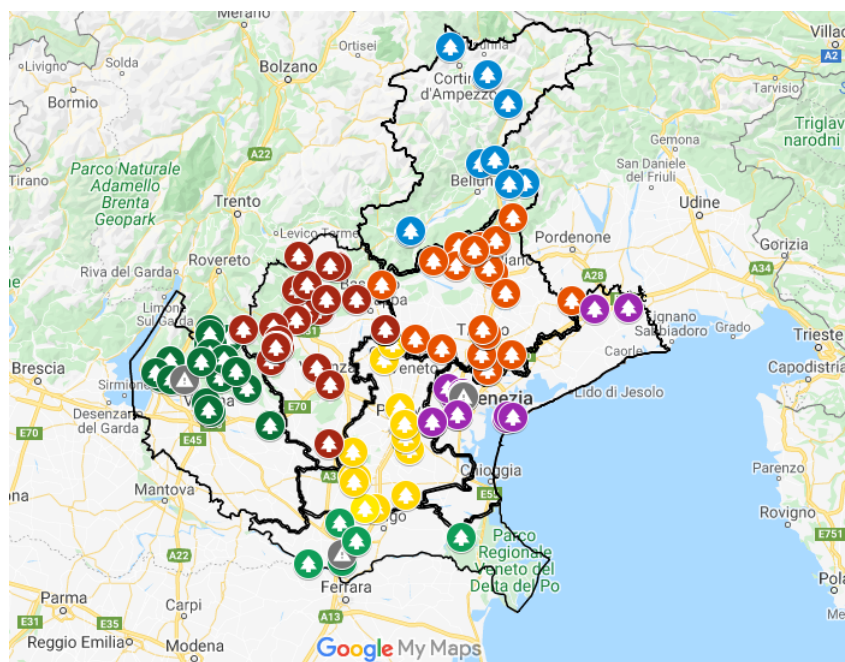


Figura 5.2: Mappa degli alberi monumentali del Veneto

Capitolo 6

Elaborazione dati con Python

In questa sezione ci occuperemo della fase in cui lo studente viene messo nella condizione di sviluppare competenze inerenti la disciplina dell'informatica. Innanzitutto inquadreremo quali competenze vogliamo che lo studente acquisisca:

Competenze attese al termine del progetto:

- Utilizzare le strategie e gli algoritmi affrontati in precedenza in classe per risolvere situazioni problematiche, elaborando opportune soluzioni. In particolare:
 - Saper scegliere il dato strutturato più idoneo per elaborare i dati acquisiti
 - Saper utilizzare in modo opportuno i costrutti di iterazione per operare matematicamente con i dati a disposizione
- Saper lavorare in gruppo
- Saper gestire il proprio tempo

Gli studenti già divisi in gruppi, al rientro dalla campagna di acquisizione dati, avranno a disposizione una serie di file con estensione **.gpx**. Per prima cosa è necessario estrapolare i dati di posizione (latitudine, longitudine e altitudine) e tempo e copiarli in una struttura idonea per poter essere manipolati. Una volta organizzati i dati, i ragazzi dovranno calcolare lunghezza percorsa e velocità media per poter poi ottenere i grafici di legge oraria, velocità-tempo e dislivello. Entriamo più nello specifico delle fasi di elaborazione dati.

6.1 Estrapolare liste da file **.gpx**

Innanzitutto premettiamo che in questo progetto abbiamo utilizzato l'editor Jupyter disponibile tramite il software Anaconda.

Per estrapolare liste da file **.gpx** ci faremo aiutare dalla libreria **gpxpy**. Siamo consapevoli che questa libreria non permette agli studenti di avere un controllo effettivo del codice, sarà di conseguenza presentata come “scatola nera” che dà la possibilità di creare le liste di dati da un file **.gpx**. Alternativamente potrebbe essere utilizzata la libreria **minidom** con la quale si possono estrapolare dati dal più generale schema **.xml**, di cui il **.gpx** è un caso particolare. Nella figura 6.1 è riportata la parte di codice con l'uso di **gpxpy**.


```

nome_file = '2020-05-31_08-50-43_giro'
gpx_file = open('{0}.gpx'.format(nome_file), 'r')

gpx = gpxpy.parse(gpx_file)

#crea una lista per ogni dato
time = []
for track in gpx.tracks:
    for segment in track.segments:
        for point in segment.points:
            time.append(point.time)

latitude = []
for track in gpx.tracks:
    for segment in track.segments:
        for point in segment.points:
            latitude.append(point.latitude)

longitude = []
for track in gpx.tracks:
    for segment in track.segments:
        for point in segment.points:
            longitude.append(point.longitude)

elevation = []
for track in gpx.tracks:
    for segment in track.segments:
        for point in segment.points:
            elevation.append(point.elevation)

```

Figura 6.1: Uso della libreria gpxpy per estrapolare liste da file **.gpx**

6.2 Passare a un sistema di coordinate cartesiane

A questo punto abbiamo a disposizione 4 liste che comprendono i dati di: tempo, latitudine, longitudine e altitudine dei punti acquisiti durante il percorso. Per poter maneggiare un unico dato strutturato, una strada può essere quella di raggruppare tutte le liste in un unico dizionario, che chiamiamo *track_points*. Costruiamo in questo modo un dizionario in cui le chiavi sono rispettivamente *time*, *latitude*, *longitude* e *elevation* e i cui valori associati sono le liste precedentemente definite. La definizione del dizionario viene riportata in figura 6.2.

```
track_points = dict()
track_points['time'] = time
track_points['latitude'] = latitude
track_points['longitude'] = longitude
track_points['elevation'] = elevation
```

Figura 6.2: Definizione del dizionario di liste

Siamo pronti per creare nuove liste da aggiungere al dizionario con le coordinate dei punti nel sistema di assi cartesiani.

Riferendoci alla figura 6.3 è possibile passare dalle coordinate sferiche a quelle cartesiane proiettando la lunghezza dei vettori lungo i nuovi assi per mezzo della trigonometria.

Per convenzione abbiamo scelto di aggiungere al parametro di altitudine restituito dal GPS la misura del raggio medio della terra pari a 6.371×10^6 m, in questo modo ricaviamo la lunghezza ρ (si veda figura 6.3) con centro del sistema di coordinate corrispondente al centro della terra.

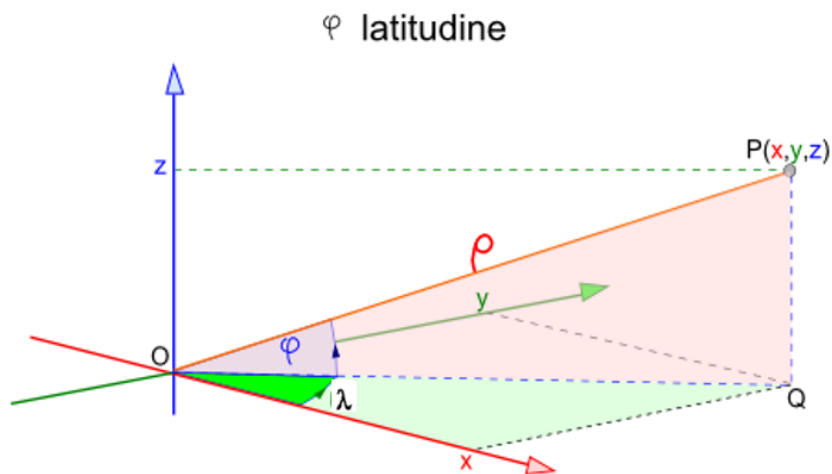


Figura 6.3: Passaggio da latitudine ϕ , longitudine λ e altitudine a x , y e z fonte (<https://www.youmath.it/lezioni/analisi-due/varie/2278-coordinate-sferiche.html>)

Fatto ciò, importando la libreria **math**, che viene richiamata con *mt* per brevità, possiamo definire le nuove liste con le coordinate x , y e z per tutti i punti acquisiti. Per prima cosa quindi aggiungiamo il raggio della terra a tutte le altitudini che abbiamo nella lista *elevation*. Dopo aver definito una lista vuota chiamata *r* (`track_points['r'] = []`) utilizziamo il costrutto *while* come indicato in figura 6.4 e aggiungiamo ad ogni ciclo un elemento alla lista *r* tramite il metodo *append()*.

A questo punto avendo la lunghezza del vettore posizione r possiamo proiettarlo lungo le tre direzioni conoscendo gli angoli di latitudine e longitudine come indicato in figura 6.5.

```

while i < len(track_points['time']):

    track_points['r'].append(track_points['elevation'][i] + 6371000)

    i = i + 1

i = 0

```

Figura 6.4: Creazione della lista r che corrisponde alla ρ della figura 6.4

```

i = 0
track_points['x'] = []
track_points['y'] = []
track_points['z'] = []

while i < len(track_points['time']):

    track_points['x'].append(track_points['r'][i] * mt.
↪cos(track_points['longitude'][i]*mt.pi/180) * mt.
↪cos(track_points['latitude'][i]*mt.pi/180))

    track_points['y'].append(track_points['r'][i] * mt.
↪sin(track_points['longitude'][i]*mt.pi/180) * mt.
↪cos(track_points['latitude'][i]*mt.pi/180))

    track_points['z'].append(track_points['r'][i] * mt.
↪sin(track_points['latitude'][i]*mt.pi/180))

    i = i + 1

```

Figura 6.5: Proiezione del vettore posizione lungo gli assi cartesiani

6.3 Calcolo della distanza, tempo e velocità media

Ora che abbiamo la posizione di tutti i punti nel sistema cartesiano possiamo calcolare la distanza fra un punto e l'altro utilizzando il teorema di Pitagora. Le distanze parziali vengono salvate nella nuova lista *dist_parz* in cui il primo valore nullo viene aggiunto prima di cominciare il ciclo *while*. La stessa cosa viene fatta per calcolare l'intervallo di tempo trascorso tra due acquisizioni consecutive come viene mostrato in figura 6.6.

```
track_points['dist_parz'] = []
track_points['dist_parz'].append(0)
track_points['delta_t'] = []
track_points['delta_t'].append(0)

i = 0

while i < ( len(track_points['time'])-1 ): #la lunghezza della lista
↳track_points['dist_parz'] conta un elemento in meno rispetto a quelle viste
↳fino a qui
    track_points['dist_parz'].append(mt.sqrt(mt.pow( (track_points['x'][i+1] -
↳track_points['x'][i] ), 2) + mt.pow( (track_points['y'][i+1] -
↳track_points['y'][i] ), 2)+mt.pow( (track_points['z'][i+1] -
↳track_points['z'][i] ), 2)))
    track_points['delta_t'].append(track_points['time'][i+1] -
↳track_points['time'][i])
    i = i + 1
```

Figura 6.6: Creazione delle liste di distanza parziale e intervalli di tempo

Infine aggiungiamo le liste di distanza progressiva e tempo progressivo, che ci serviranno per creare i grafici nella sezione seguente. Mostriamo in figura 6.7 anche la parte di codice in cui viene calcolata la velocità media per ogni coppia di punti consecutivi, anche in questo caso per convenzione il primo termine della lista viene posto uguale a zero.

```
track_points['progressive_t'] = []
track_points['progressive_dist'] = []
track_points['average_vel'] = []
track_points['average_vel'].append(0)
progressive_dist = 0

while i < len(track_points['time']):

    track_points['progressive_t'].append( (track_points['time'][i] -
↳track_points['time'][0]).total_seconds() )

    progressive_dist = progressive_dist + track_points['dist_parz'][i]
    track_points['progressive_dist'].append(progressive_dist)

    i = i + 1

i = 1

while i < len(track_points['time']):
    track_points['average_vel'].append( track_points['dist_parz'][i] / (
↳track_points['delta_t'][i].total_seconds() ) * 3.6 )
    i = i + 1
```

Figura 6.7: Distanza e tempo progressivi + velocità media

6.4 Plot dei dati

Siamo nella fase di rappresentazione grafica dei dati. Come esempi mostreremo i grafici di legge oraria, velocità-tempo e dislivello di un tratto del percorso alla scoperta degli alberi monumentali.

Utilizziamo la libreria **pygal** per ottenere grafici personalizzabili con diversi stili accattivanti (in alternativa si può utilizzare la più classica libreria **matplotlib**).

Per prendere parti di tracciato si chiede ai ragazzi di acquisire le posizioni di determinati punti che costituiranno l'inizio e la fine di quel particolare percorso; in questo caso dal tempio buddista al primo albero monumentale trovato. Per fare ciò utilizziamo il metodo *index()* come viene mostrato in figura 6.8.

```
approx_longitude = []
i = 0
while i < ( len(longitude) ):

    approx_longitude.append(round(longitude[i],5))

    i = i+1

index_start = approx_longitude.index(11.05435)
index_end = approx_longitude.index(11.06479)
```

Figura 6.8: Trovare gli indici di inizio e fine percorso

È da notare che per trovare l'indice di inizio e fine percorso abbiamo approssimato la longitudine al quinto decimale utilizzando *round*, questo per essere più sicuri di trovare il punto considerato. Fatto ciò è possibile costruirsi un nuovo dizionario che chiamiamo *track_points_1* utilizzando lo **slice** come viene mostrato in figura 6.9.

```
time_1 = time[index_start:index_end+1]
longitude_1 = longitude[index_start:index_end+1]
latitude_1 = latitude[index_start:index_end+1]
elevation_1 = elevation[index_start:index_end+1]

track_points_1 = dict()
track_points_1['time'] = time_1
track_points_1['latitude'] = latitude_1
track_points_1['longitude'] = longitude_1
track_points_1['elevation'] = elevation_1
```

Figura 6.9: Lo *slicing* delle liste

Prima di produrre i grafici conviene “sfolciare” le liste che abbiamo a disposizione, per non avere i grafici con i punti troppo ravvicinati. Come lo studente può usare efficacemente il ciclo *while*? Basta iterare sulla lista dei dati di partenza e creare una nuova lista saltando con l'indice di un numero predefinito di iterazioni. In questo caso ci è sembrato ragionevole saltare di 50 in 50 come mostrato in figura 6.10.

Si può notare che alla fine è stata sovrascritta la lista con la distanza progressiva arrotondando alla seconda cifra decimale, questo per avere sull'asse delle *x* numeri più leggibili.

Finalmente abbiamo la possibilità di produrre immagini con i grafici usando la libreria **pygal**. Questa libreria produrrebbe di norma immagini vettoriali con estensione **.svg** ma in questo ambito, importando la libreria **cairosvg**, possiamo esportare immagini in **.png**.

```

i = 0
discretizza_progressive_dist_1 = []
discretizza_elevation_1 = []
discretizza_average_vel_1 = []
discretizza_progressive_t_1 = []
while i < ( len(track_points_1['time'])-1 ):
    discretizza_progressive_dist_1.append(track_points_1['progressive_dist'][i])
    discretizza_elevation_1.append(track_points_1['elevation'][i])
    discretizza_average_vel_1.append(track_points_1['average_vel'][i])
    discretizza_progressive_t_1.append(track_points_1['progressive_t'][i])
    i = i+50

i = 1
while i < ( len(discretizza_progressive_dist_1) ):

    discretizza_progressive_dist_1[i] = round(discretizza_progressive_dist_1[i],2)

    i = i+1

```

Figura 6.10: Creare nuove liste con i dati meno fitti

Agli studenti si possono mostrare le righe di codice di un grafico di esempio e, in base a quelle, utilizzando i vari stili predefiniti gli studenti possono personalizzare i loro grafici come vogliono. In figura 6.11 è mostrata la parte di codice che produce il grafico del dislivello.

```

from pygal.style import DarkSolarizedStyle
line_chart = pygal.StackedLine(fill=True,
                                show_legend=False,
                                style=DarkSolarizedStyle,
                                x_label_rotation=90,
                                x_title= 'distanza progressiva [m]',
                                y_title = 'altitudine [m]')

line_chart.title = 'dislivello'
line_chart.x_labels = discretizza_progressive_dist_1
line_chart.add('Elevation', discretizza_elevation_1)
line_chart.fill = True
line_chart.human_readable = True
line_chart.render_in_browser()
line_chart.render_to_png('budda-faggio_dislivello.png')

```

Figura 6.11: Uso della libreria **pygal**

Esempi dei risultati che si possono ottenere sono mostrati nelle figure 6.12, 6.13 e 6.14.

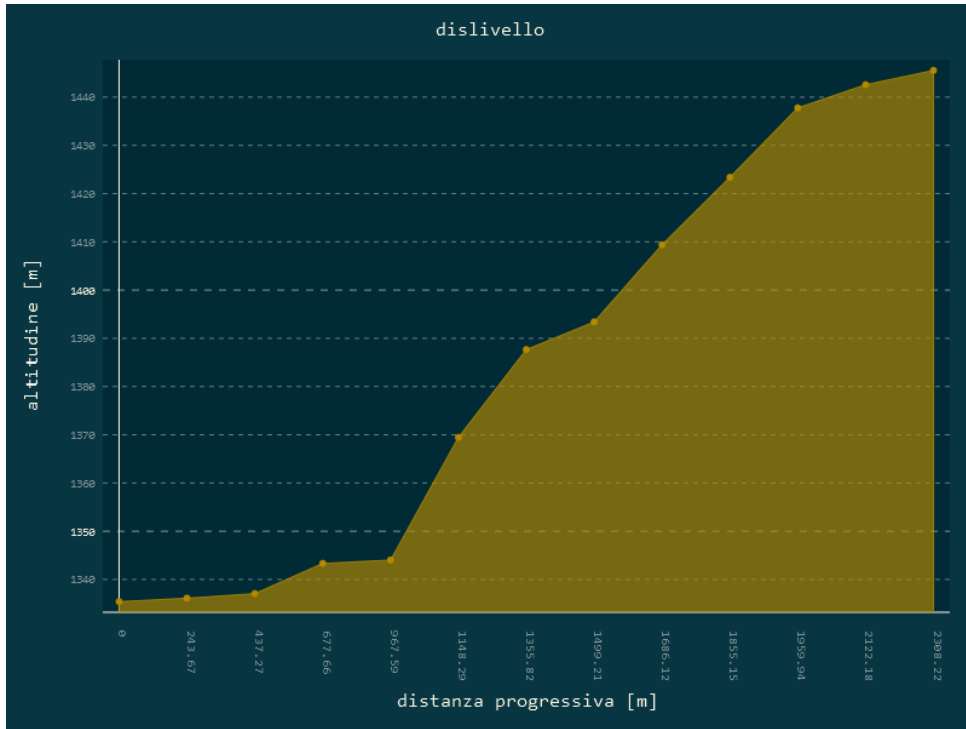


Figura 6.12: Grafico dislivello

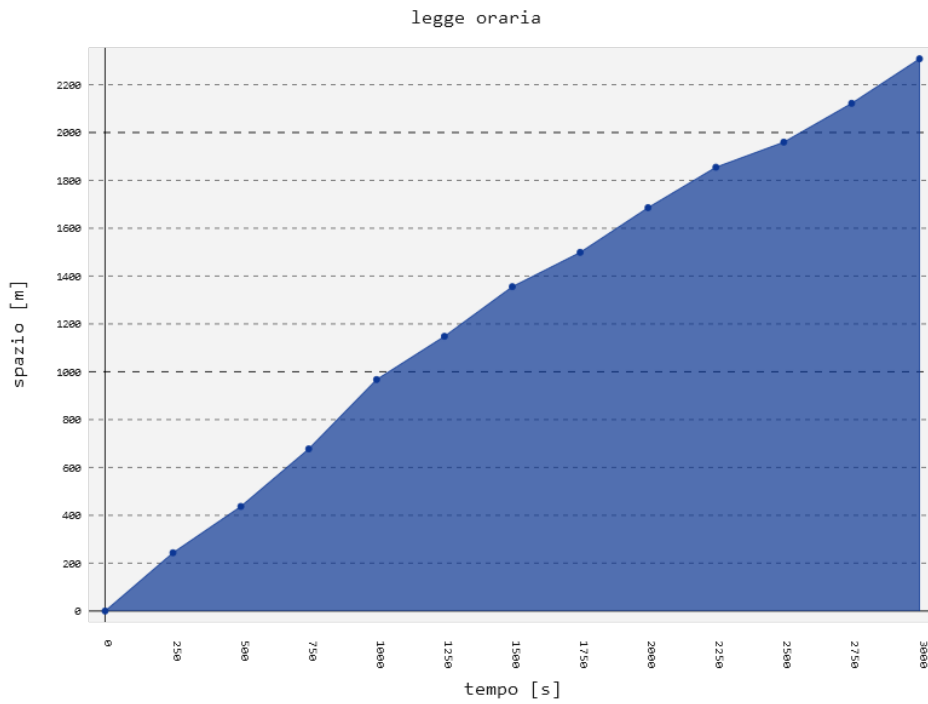


Figura 6.13: Legge oraria

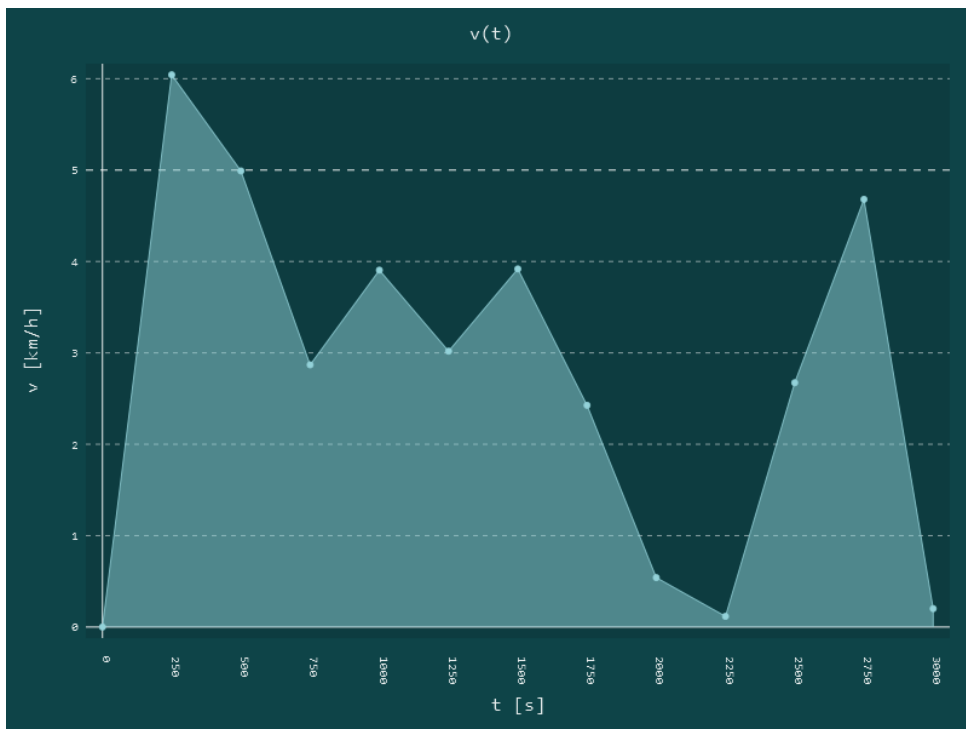


Figura 6.14: Grafico velocità-tempo

Capitolo 7

Applicazioni sociali

Una volta elaborati i dati raccolti nelle attività precedenti puntiamo a far vedere ai ragazzi come l'informatica possa rivelarsi utile anche a livello sociale.

In riferimento alle strategie dell'apprendimento cooperativo e del metodo dei progetti, i ragazzi divisi in gruppi svilupperanno un'applicazione turistica in cui organizzeranno i grafici e i dati elaborati, aggiungendo informazioni e fotografie dei luoghi visitati. Questa fase dà ulteriori possibilità di interdisciplinarietà in quanto si possono coinvolgere materie come storia e storia dell'arte.

Lo strumento che abbiamo scelto per questo scopo è MIT App Inventor (<https://appinventor.mit.edu/>).

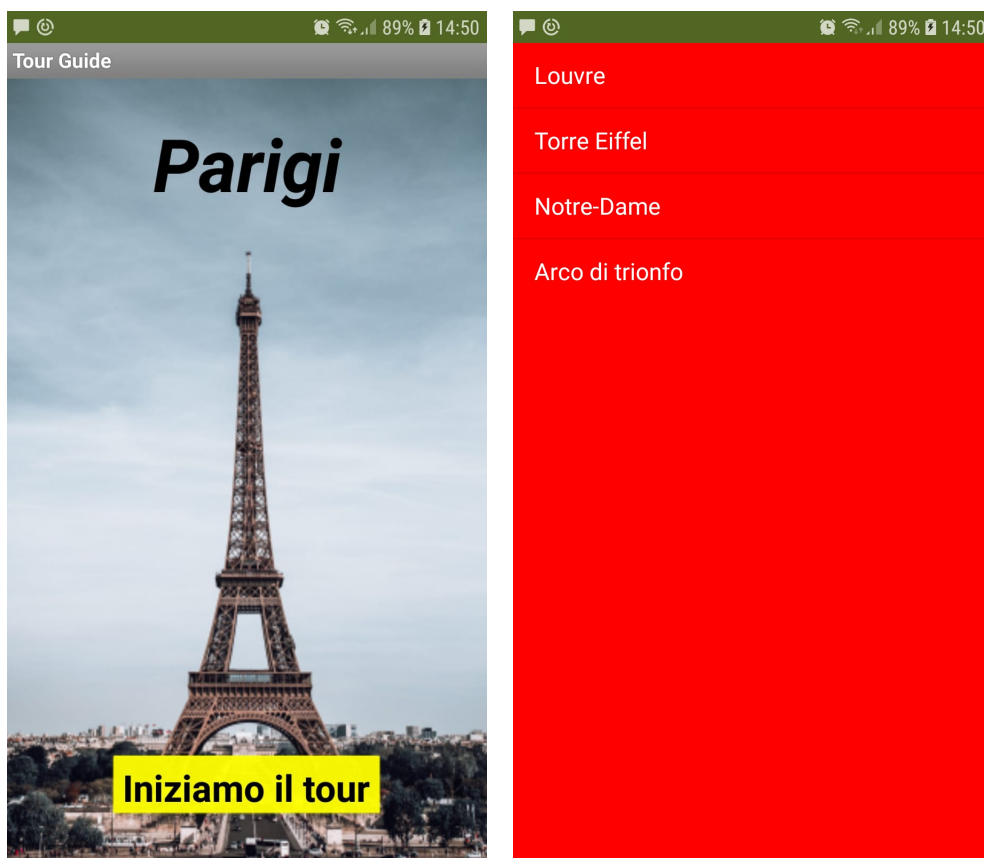
7.1 Svolgimento

- Lezioni preliminari per conoscere MIT App Inventor: Far vedere agli studenti come funziona Mit App Inventor, mostrando degli esempi di cose che già sanno fare con altri programmi. Bisognerà soprattutto soffermarsi sul tool "Maps" per mostrare come sfruttare MIT App per creare applicazioni legate al nostro argomento, cioè il GPS.
- Divisione in gruppi: L'insegnante formerà dei gruppi equilibrati da 3 o 4 persone seguendo la strategia dell'apprendimento collaborativo di cui abbiamo parlato nei capitoli precedenti. I gruppi potranno essere gli stessi delle fasi precedenti o potranno essere rimescolati a scelta del docente.
- Linee guida da seguire per il risultato finale: Gli studenti saranno per lo più lasciati liberi di creare il design della propria applicazione ma per far sì che siano "obbligati" a programmare utilizzando tutte le conoscenze acquisite negli anni precedenti, sarà opportuno dare delle linee guida da seguire. Esempio: Creare una lista con ListPicker in modo che debbano creare una variabile e poi la debbano utilizzare per definire gli elementi della lista.
- Lezioni dedicate alla condivisione delle idee dei progetti: Prima dell'esposizione finale i vari gruppi condivideranno le proprie idee sulla progettazione dell'app con gli altri compagni in modo che possano ricavare feedback e suggerimenti per migliorare il proprio progetto.
- Esposizione finale del progetto

Vediamo ora delle immagini rappresentative di un'applicazione turistica creabile con MIT App Inventor.



Figura 7.1: Nella figura vediamo il ListPicker di cui abbiamo parlato nelle linee guida da seguire.



Prima tappa: Louvre




Il museo del Louvre di Parigi, in Francia, è uno dei più celebri musei del mondo e il primo per numero di visitatori: 10 milioni nel 2018. Si trova sulla rive droite, nel 1° arrondissement, tra la Senna e Rue de Rivoli.

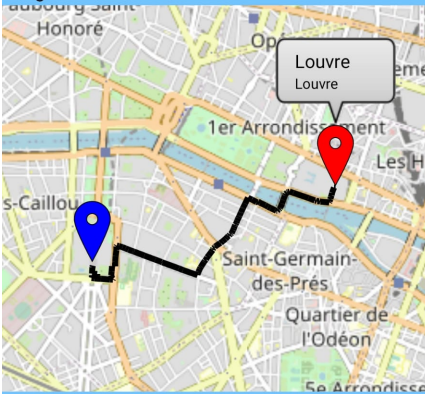
Percorsi e tempo percorrenza

Torna indietro

Prima tappa: Louvre

Dove mi trovo

Indirizzo: Hôtel des Invalides
 Latitudine: 48.854904
 Longitudine: 2.312461



Distanza: 2,6 km
 Tempo di percorrenza: 33 minuti

Grafici

Torna indietro

Prima tappa: Louvre

Grafico velocità-tempo

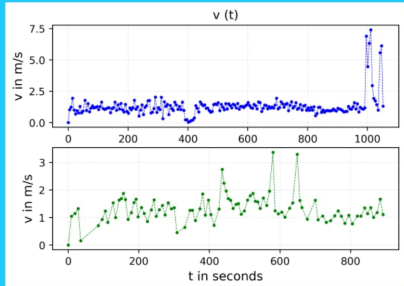
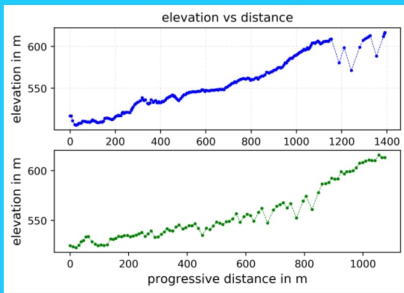


Grafico dislivello-spazio



Torna indietro

Torna alla pagina iniziale

Capitolo 8

Conclusioni e possibili sviluppi

Il percorso pensato in questa proposta didattica vuole essere un punto di partenza, uno spunto per essere poi arricchito dagli insegnanti e dagli studenti che sperabilmente lo contamineranno con le loro idee e sperimentazioni.

Siamo aspiranti insegnanti consapevoli dell'importanza di avere un'istruzione di qualità e desiderosi di creare le condizioni perché lo studente si stupisca e colga la bellezza anche dietro righe di codice.

Concludiamo questo lavoro dando ulteriori spunti che possono dare svolte inedite a questa unità didattica, in particolare con l'uso di QGis, le Mapathon di Humanitarian OpenStreetMap Team, il ciclista che disegna sulle mappe con la sua bicicletta e i tracciati segreti dei gatti.

8.1 L'uso del software QGis

Uno strumento dalle elevate potenzialità è costituito da **QGis**, un software open source che permette di lavorare sui dati georiferenziati con interfaccia utente.

Ad esempio la prima cosa che potremmo fare è importare il nostro file **.gpx** e mostrarlo sulla mappa di Open Street Map. È possibile creare un'immagine con la nostra traccia indicata in figura 8.1.

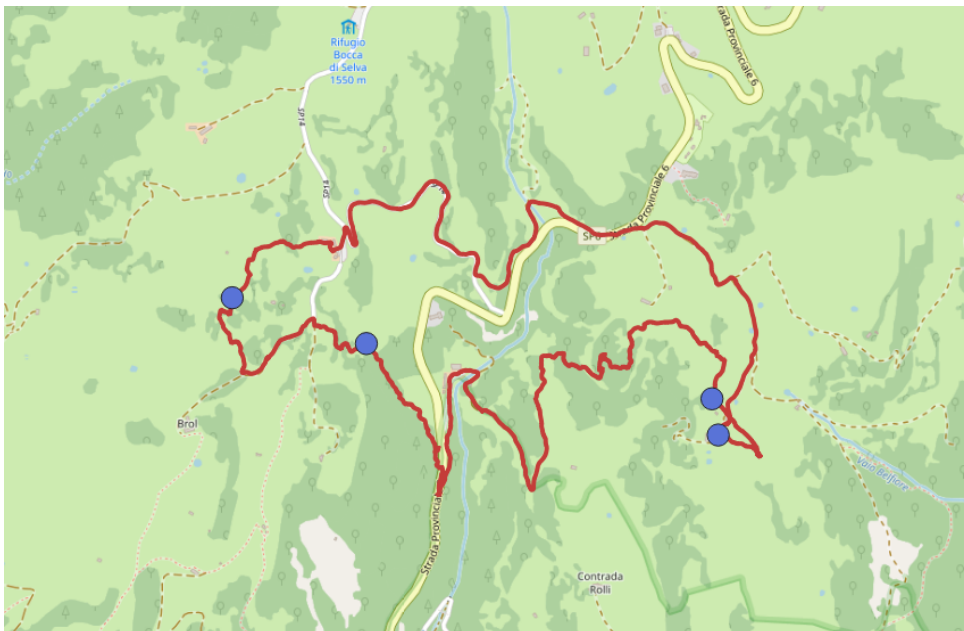


Figura 8.1: Traccia su Open Street Map

Un'altra applicazione che ci teniamo a segnalare è l'uso di Web Gis nel mappare i luoghi colpiti da catastrofi naturali e privi di riferimenti geografici aggiornati. In tali contesti le informazioni geografiche

risultano ancor più necessarie per affrontare emergenze nel recapitare aiuti umanitari alla popolazione in difficoltà.

L'organizzazione di volontariato **Humanitarian OpenStreetMap Team** (si veda figura 8.2) organizza le cosiddette Mapathon, che consistono in eventi nei quali chiunque può aiutare a georeferenziare infrastrutture e abitazioni tramite riconoscimento delle stesse nelle ortofoto a disposizione.



Figura 8.2: Humanitarian Open Street Map

8.2 Disegnare sulle mappe con la bicicletta

Stephen, un artista del GPS si è inventato un'applicazione divertente con la sua bicicletta. La sua attività consiste nel percorrere le strade di diverse città creando figure davvero incredibili. Nel suo blog (<https://gpsdoodles.com/>) si possono trovare disegni anche bizzarri come il supercane riportato in figura 8.3.

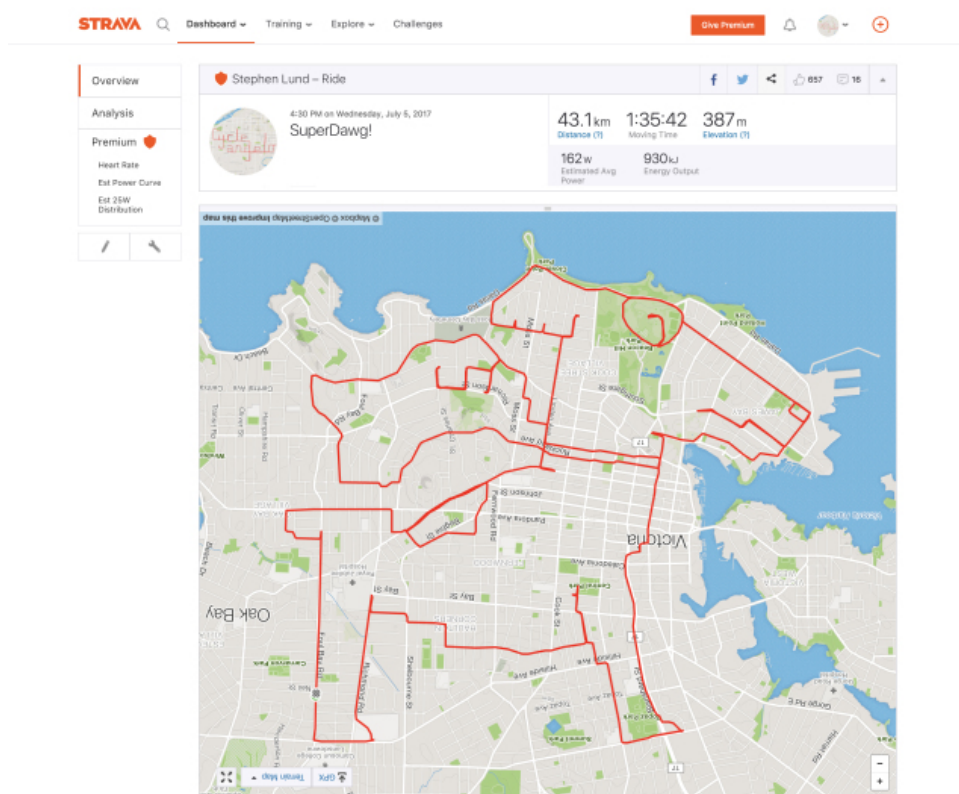


Figura 8.3: Uno dei disegni di Stephen con la bicicletta

8.3 La vita segreta dei gatti

Un altro esempio che può stimolare la curiosità degli studenti è quello di mostrare cosa hanno fatto qualche anno fa in Inghilterra per studiare dove vanno i gatti domestici quando non sono nelle abitazioni (<https://www.bbc.com/news/science-environment-22567526>), il risultato è mostrato in figura 8.4.

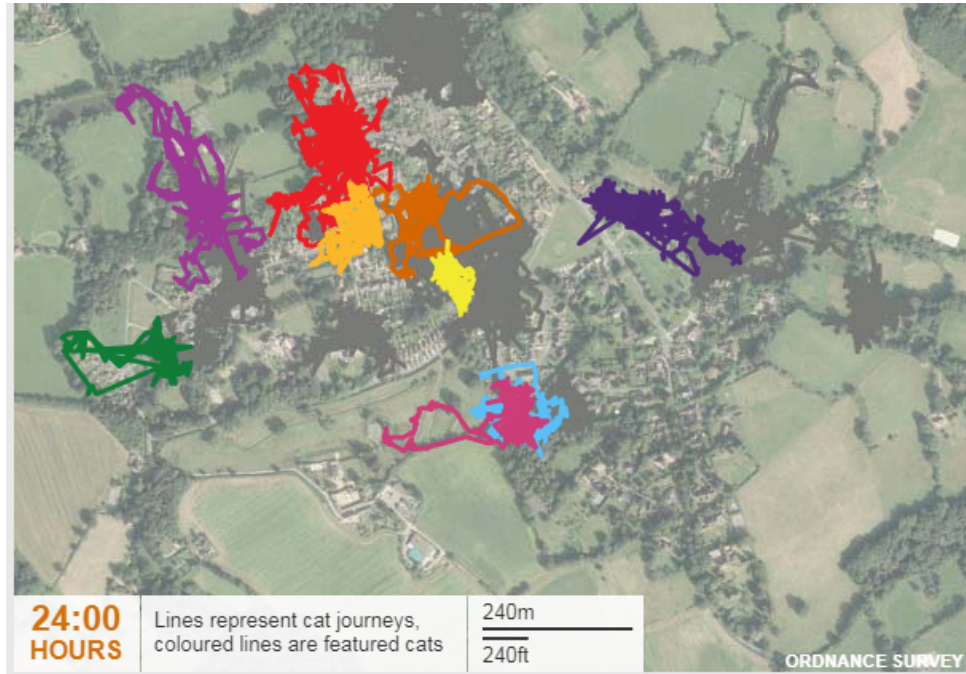


Figura 8.4: La vita segreta dei gatti

Bibliografia

- Edith Ackermann. Piaget's Constructivism , Papert's Constructionism: What's the difference? *Future of learning group publication*, 5:1–11, 2001. ISSN 1074-8121. doi: 10.1.1.132.4253. URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.132.4253{&}rep=rep1{&}type=pdf>.
- Giovanni Bonaiuti. *Le strategie didattiche*. Carocci Editore, 2014.
- L. Langè, G. & Cinganotto. *I Quaderni della Ricerca - Imparare per competenze*. 2013. ISBN 9788858310502.
- Michael Lodi, Simone Martini, and Enrico Nardelli. Abbiamo davvero bisogno del pensiero computazionale? *Mondo Digitale*, 16(72), 2017. ISSN 1720898X.
- MIUR. Indicazioni nazionali degli obiettivi specifici di apprendimento per i licei. 2010. URL http://www.indire.it/lucabas/lkmw_file/licei2010/indicazioni_nuovo_impaginato/_decreto_indicazioni_nazionali.pdf.
- Seymour Papert. Hard fun, 2002. URL <http://www.papert.org/articles/HardFun.html>.
- R. Keith Sawyer. *The Cambridge Handbook of the Learning Sciences*. Cambridge University Press, 2005. doi: 10.1017/CBO9780511816833.