

POLITECNICO DI TORINO

Corso di Laurea Magistrale in
INGEGNERIA GESTIONALE

A.a. 2021/2022

Sessione di Laurea Dicembre 2022



**Politecnico
di Torino**

**PixelCodingStudio: piattaforma per
l'apprendimento del coding unplugged**

Relatore

Prof. Alessandro Fiori

Candidato

Antonio Monteleone

Sommario

Il coding, inteso come la capacità di programmazione informatica, sta diventando sempre più un aspetto fondamentale nella formazione scolastica. La Pixel-art è una delle attività di coding unplugged più utilizzate per l'introduzione del pensiero computazionale e l'insegnamento della programmazione nella scuola dell'infanzia e primaria. Per questo motivo, questa tesi si è posta come obiettivo lo sviluppo di PixelCodingStudio, una piattaforma per aiutare gli insegnanti, in modo semplice ed intuitivo, sia a realizzare esercizi di coding unplugged attraverso la Pixel-art, sia a condividere i propri progetti con la comunità scolastica. PixelCodingStudio è un'applicazione web in cui gli utenti possono creare la propria griglia, colorarla a loro piacimento e visualizzare con un click il relativo codice-colore che gli permetterà di ricreare quanto fatto approcciando il pensiero computazionale. Le Pixel-art possono essere generate anche a partire da immagini grazie all'impiego di algoritmi di clustering per la creazione delle griglie. Infine, è possibile salvare i progetti creati, visualizzarli, modificarli e condividerli con la community. Essendo un'applicazione web, essa è stata progettata implementando un'architettura suddivisa in tre livelli: il client, il server e la base dati. Per lo sviluppo si è fatto uso di diverse tecnologie e linguaggi di programmazione con l'ausilio di framework e librerie. Lo studio effettuato si è concentrato anche sull'analisi di diversi algoritmi di clustering fondamentali per il processamento delle immagini durante la pixellizzazione.

Ringraziamenti

Prima di procedere con la trattazione, vorrei dedicare qualche riga a tutti coloro che mi sono stati vicini in questo percorso di crescita personale e professionale.

Un sentito grazie al mio relatore Fiori Alessandro per la sua infinita disponibilità e tempestività ad ogni mia richiesta. Grazie per avermi dato questa possibilità.

Un ringraziamento che non sarà mai abbastanza va ai miei genitori, a Papà e Mamma, il mio punto di riferimento. Grazie per aver permesso tutto ciò, per avermi sempre sostenuto in ogni mia decisione e per crederci sempre. Grazie per tutti i sacrifici fatti e per quelli che continuate a fare, a voi devo tutto. Grazie per non avermi mai fatto mancare nulla. Grazie per tutti i consigli, i valori che mi avete trasmesso e per avermi insegnato a vivere, se sono quello che sono diventato oggi è merito vostro. Grazie per esserci sempre, perché anche da lontani siete sempre con me, con voi mi sentirò sempre al sicuro. Vi voglio bene.

Un grazie speciale ai miei fratelli, Andrea e Daniele, perché con voi tutto è più leggero, perché voi siete i miei fans numero uno. Grazie per avermi sempre supportato e per darmi sempre la carica necessaria. Grazie per tutti i consigli, le confidenze ed i momenti di divertimento. Grazie perché so che voi ci sarete sempre per me come io ci sarò per voi. Grazie perché ognuno può contare sull'altro. Grazie perché con voi non mi sentirò mai solo.

Ringrazio la mia famiglia, i miei zii, i miei cugini e le mie cognate per tutte le giornate belle passate insieme ed i momenti di spensieratezza che mi avete concesso. Grazie perché con voi mi sento a casa. In particolare, un grazie va alla mia madrina Marianna per essere stata il mio mentore, per aver fatto parte di questo mondo da prima di me ed avermi guidato durante tutto il percorso con i suoi consigli.

Grazie ad Alessandro perché si dimostra essere un vero amico, un fratello. Grazie per essere stato una spalla sia nei momenti belli ma soprattutto in quelli brutti. Grazie per supportarmi, per i consigli, per tutte le risate, le nottate e le avventure passate insieme. Grazie perché su di te posso contare.

Grazie a Syria, per credere sempre in me e più di me. Grazie per esserci sempre ed incoraggiarmi. Grazie per tutto quello che abbiamo condiviso, gli esami, le chiamate ad ogni ora ed i concerti. Grazie perché con te posso confidarmi e parlare di ogni cosa. Grazie nonostante tutto lo stress che mi causi, perché in realtà ogni momento passato insieme sembra volare. Grazie per essere una vera amica.

Un grazie a tutti i miei amici, da chi c'è da più tempo nella mia vita a chi da meno per aver reso il tutto più bello e divertente. Un grazie particolare ai miei gestionfirends per aver condiviso tanto di questo percorso, dalle lezioni agli esami, le fatiche, i progetti ed i momenti di delusione, ma grazie soprattutto per avermi sempre strappato un sorriso.

Un ringraziamento più in generale a tutti coloro che mi hanno sempre sostenuto, a chi vede questo traguardo con orgoglio, a chi ha fatto parte della mia vita e continua a farlo o a chi magari non c'è più, ma grazie davvero a tutti perché in questo momento sono esattamente dove vorrei essere.

Ed infine, un grazie a me stesso, per essermi superato, per la costanza e la dedizione che ci ho messo in ogni singola azione e perché mai mi sarei aspettato tutto ciò, ma nonostante le difficoltà non ho smesso di provarci. Grazie perché ora posso ritenermi soddisfatto e orgoglioso di me stesso.

Indice

Elenco delle tabelle	VIII
Elenco delle figure	IX
1 Introduzione	1
1.1 Panoramica sulla tesi	2
2 Tematiche trattate	3
2.1 Coding	4
2.2 Pensiero computazionale	6
2.3 Coding e pensiero computazionale: l'importanza nelle scuole	10
2.4 Coding Unplugged	13
2.5 Pixel-art	15
3 Tecnologie	17
3.1 Docker	18
3.2 Python	21
3.3 OpenCV	21
3.4 Scikit-learn	22
3.5 xhtml2pdf	22
3.6 Django	23
3.7 Alternative a Django	25
3.7.1 Flask	25
3.7.2 Web2py	26
3.8 Django-SES	27
3.9 JavaScript	27
3.10 jQuery	28
3.11 D3.js	28
3.12 HTML	29
3.13 CSS	30
3.14 Bootstrap	30

3.15	MySQL	31
3.16	PhpMyAdmin	33
3.17	Clustering	33
3.17.1	K-Means	34
3.17.2	K-Medoids	36
3.17.3	DBSCAN	37
4	Implementazioni	39
4.1	Database	40
4.1.1	Modello ER	40
4.1.2	Modello logico	42
4.2	Funzionalità	46
4.2.1	Gestione utente	46
4.2.2	Gestione progetti	47
4.2.3	Community	50
5	Analisi algoritmi di clustering	52
6	Casi d'uso	60
6.1	Creazione Pixel-art	61
6.2	Salvataggio progetti	65
6.3	Visualizzazione codice-colore	66
6.4	Ricerca progetti	68
6.5	Registrazione	69
7	Conclusioni	71
7.1	Sviluppi futuri	72
	Bibliografia	73

Elenco delle tabelle

4.1	User	43
4.2	Project	44
4.3	Category	45
4.4	Tag	45
5.1	Caratteristiche immagine HOMER	54
5.2	Caratteristiche immagine SUPER MARIO	55
5.3	Caratteristiche immagine MARE	57

Elenco delle figure

2.1	Modello della spirale dell'apprendimento creativo	9
2.2	Pixel-art	15
3.1	Architettura Docker	20
3.2	Logica di funzionamento Django	24
3.3	K-Means	35
3.4	K-Medoids	36
3.5	DBSCAN	38
4.1	Modello ER	41
5.1	Immagini per il test	53
5.2	Test HOMER	54
5.3	Grafico HOMER	55
5.4	Grafico SUPER MARIO	56
5.5	Test SUPER MARIO	56
5.6	Grafico MARE	57
5.7	Test MARE	58
6.1	Diagramma UML dei casi d'uso del sistema	61
6.2	Pagina Crea	62
6.3	Modulo per caricare immagine	64
6.4	Pagina Crea con immagine caricata	64
6.5	Modulo per salvare progetto	65
6.6	Pdf contenente codice-colore	66
6.7	Pdf contenente legenda colori	67
6.8	Pdf contenente griglia bianca	67
6.9	Pdf contenente Pixel-art	68
6.10	Pagina Community	69
6.11	Pagina Registrati	70

Capitolo 1

Introduzione

Nel 2022 il coding è stato reso obbligatorio in tutte le scuole dell'infanzia e primaria (Mozione n. 1- 00117 del 12 marzo 2019), in conformità alle indicazioni nazionali per il curricolo. Inoltre, a partire dal 2025/26 il coding diventerà argomento di studio in tutte le scuole italiane. I primi anni di scuola sono, infatti, il momento più indicato per introdurre gli alunni alla programmazione, intesa sia come digitale, sia come un modo computazionale di pensare al mondo che ci circonda.

Il coding consente ai bambini di implementare e migliorare le capacità logiche e analitiche ed è uno strumento prezioso per stimolare la creatività. Ad oggi, la programmazione informatica, elettronica e robotica è stata considerata la quarta abilità essenziale, la base per la corretta alfabetizzazione dei linguaggi delle nuove tecnologie per una nuova generazione di nativi digitali. Sono infatti strumenti indispensabili per lo sviluppo di competenze trasversali e di processi logici e creativi, contribuendo a porre gli studenti al centro del processo di apprendimento, rendendoli attori attivi dei futuri sviluppi digitali.

Insegnare coding e robotica nelle scuole significa progettare lezioni basate su modelli pedagogici innovativi e strategici, non più erogati staticamente, ma costruiti attraverso laboratori ed esercitazioni che vedono protagonisti gli studenti: attivando il processo di scoperta della conoscenza, il bambino inizia a migliorare le capacità di problem solving da solo mettendosi in gioco. La scelta di un metodo divertente è strategica in quanto permette di catturare l'attenzione anche degli studenti più distratti e demotivati che pensano di giocare e imparare a risolvere problemi più o meno complessi in base alla loro fascia d'età.

Questa Tesi Magistrale si è concentrata sullo sviluppo di PixelCodingStudio, una piattaforma per aiutare gli insegnanti nelle scuole dell'infanzia e primaria a realizzare

esercizi di coding unplugged attraverso la Pixel-art e rendere possibile la condivisione dei progetti creati con la comunità scolastica in un modo semplice ed intuitivo.

PixelCodingStudio è un'applicazione web in cui gli utenti possono creare la propria griglia, colorarla a loro piacimento e visualizzare con un click il relativo codice-colore che gli permetterà di ricreare quanto fatto approcciando il pensiero computazionale. Inoltre, è possibile salvare i progetti creati, visualizzarli, modificarli e condividerli con la community.

1.1 Panoramica sulla tesi

In questo documento verranno presentati tutti gli argomenti fondamentali e le scelte progettuali fatte per raggiungere gli obiettivi preposti. In particolare nel Capitolo 2 si definisce un contesto teorico nel quale vengono spiegate le principali tematiche trattate per l'implementazione della piattaforma, affrontando concetti come il coding, il pensiero computazionale, il coding unplugged e la Pixel-art. Il Capitolo 3, si concentra sull'ambiente di sviluppo della piattaforma, ovvero sulle tecnologie utilizzate in questa Tesi per la progettazione dell'applicazione web. Tra le tecnologie descritte e spiegate in maniera teorica troviamo i diversi linguaggi di programmazione utilizzati insieme a framework, librerie ed algoritmi di clustering. Sono presenti anche delle alternative per alcune tecnologie utilizzate. Nel Capitolo 4 si parla di come sia stato costruito il database, andando ad analizzare la sua struttura e di come siano state implementate le funzionalità offerte dalla piattaforma. Il Capitolo 5 racconta l'analisi, il confronto ed i test effettuati sui diversi algoritmi di clustering, andando a spiegare le motivazioni che hanno portato alla scelta dell'algoritmo di clustering più adatto. Successivamente nel Capitolo 6 relativo ai casi d'uso vengono mostrate e spiegate le funzionalità principali e come l'utente dovrebbe interfacciarsi. La Tesi si conclude con il Capitolo 7 dove vengono stilate le conclusioni e presentati alcuni possibili sviluppi futuri.

Capitolo 2

Tematiche trattate

Il contributo dell'informatica alle nostre vite è ormai normalizzato dalla maggior parte di noi: ogni giorno tutti usiamo tablet, cellulari e altri strumenti senza pensare ai meccanismi che consentono loro di funzionare, una complessa rete di algoritmi scritti in sequenza, che grazie a linguaggi specifici consentono a queste macchine di eseguire compiti specifici. Questo processo di analisi e risoluzione di un problema non si applica solo ai robot, ma può essere applicato anche a comportamenti e problemi nella vita quotidiana e sul lavoro. Molti di noi fanno questo costantemente senza rendersene conto. Per esempio quando ci troviamo di fronte ad un obiettivo, per raggiungerlo analizziamo le vie più efficaci, pianificando una soluzione ed eseguendola. Questo processo si basa su due concetti fondamentali che sono alla base di molti dei meccanismi e delle tecnologie su cui è costruita la società umana: coding e pensiero computazionale.

Quando si parla di coding si fa riferimento alla capacità di programmazione informatica, ma è possibile considerarlo anche come uno strumento utile per imparare ed utilizzare la rete e la tecnologia, oltre che allo sviluppo del pensiero computazionale. Dato che il mercato del lavoro sta attraversando una grande rivoluzione digitale, comprendere e imparare a programmare in tenera età a scuola è fondamentale per sviluppare competenze e abilità trasversali utili nel mondo del lavoro e nella vita di tutti i giorni.

Il pensiero computazionale, d'altra parte, si riferisce ad un insieme di processi mentali che portano alla formulazione di un problema e all'espressione della sua risoluzione in modo che una macchina o un essere umano possano eseguirla. Questo processo è alla base della programmazione, ma svolge anche un ruolo utile in molte attività quotidiane: il pensiero computazionale in particolare è un importante strumento di apprendimento per i bambini e può essere efficacemente sviluppato attraverso il coding e la robotica educativa. Tanto che diventa necessario includere

il pensiero computazionale nella scuola dell'infanzia.

2.1 Coding

Coding [1] è un termine inglese al quale corrisponde l'italiano "programmazione", cioè la scrittura di una sequenza di istruzioni che fanno eseguire ad un calcolatore determinate operazioni. Non si pensi, però, soltanto a computer e software. Il nostro stesso cervello lavora così: individuazione di un problema, scelta di una soluzione, attivazione di una serie di istruzioni per metterla in atto. Quello che, in termini tecnici, viene definito "pensiero computazionale".

I processi propri del pensiero computazionale vengono messi in atto tutti i giorni, per esempio quando stabiliamo il percorso più breve per raggiungere una destinazione oppure dobbiamo preparare un piatto di pasta: la procedura per passare dalla situazione iniziale (ingredienti della ricetta) a quella finale (piatto di pasta pronto da mangiare) è data da un algoritmo (procedimento che risolve un determinato problema attraverso un numero definito di passi elementari) che viene eseguito dall'individuo. Proprio per questo, lo sviluppo del pensiero computazionale può essere stimolato da attività logiche svolte utilizzando comuni materiali ed oggetti, da soli o in gruppo, chiamate attività unplugged poiché utilizzano strumenti non digitali.

Un discorso a parte può essere fatto per il Coding in quanto concetto estremamente legato a quello del pensiero computazionale. Infatti, secondo la definizione che ne dà Bogliolo (2016) nel suo testo "Coding in your classroom, now!", il coding non può essere semplicemente tradotto in italiano con il termine "programmazione", ma in modo più complesso come *«l'uso di strumenti e metodi di programmazione visuale a blocchi per favorire lo sviluppo del pensiero computazionale»*. Alfieri (2016) aggiunge: *«il coding è la palestra del pensiero computazionale che va stimolato e allenato sin da piccoli»* e, poiché consente di apprendere le basi della programmazione informatica in modo pratico e divertente, è lo strumento didattico più utilizzato per educare i bambini al pensiero computazionale.

Va precisato, però, che il coding non può essere totalmente identificato con il pensiero computazionale, anche se attualmente rimane lo strumento più diffuso per favorirne l'acquisizione. Infatti, il pensiero computazionale utilizza strategie come algoritmi, astrazione e debugging (individuare e correggere bug, errori), che sono necessarie anche nel coding, il quale ha in più il merito di rendere concreti dei concetti altrimenti molto astratti chiedendo agli alunni di mettere in gioco in maniera pratica e tangibile le proprie competenze di logica e programmazione. *«Il*

pensiero computazionale è più della codifica di una soluzione eseguibile da un computer attraverso la programmazione» (Grover e Pea, 2017). Significa pensare come un informatico, inteso però come abilità generale che chiama in causa astrazione, rappresentazione e capacità di analisi e che non va ridotta alla sola capacità di programmare utilizzando uno specifico linguaggio.

L'informatico, nello sviluppo di un software, non si limita alla sola scrittura del codice, ma progetta la struttura del programma dopo aver compreso e definito il problema (es. esigenza del cliente) attraverso una sua analisi ed esplorazione, consapevole che non esiste un'unica soluzione, ma tante, e che differiscono tra loro soprattutto in termini di efficienza e adattabilità a nuove situazioni.

Per comprendere il coding è fondamentale conoscere almeno le basi del suo funzionamento. Alla base ci sono i concetti fondamentali dei linguaggi informatici che spiegano come opera qualsiasi programma informatico che ci troviamo ad utilizzare. Ad esempio, viene rispettata sempre una sequenza, ovvero un ordine in cui vengono scritte le istruzioni, nulla è casuale. C'è una condizione, ovvero il programma esegue differenti istruzioni in relazione a specifici casi. Un programma può eseguire le stesse istruzioni per un certo numero di volte, per questo vi è una ripetizione (ciclo). Nei programmi tutto nasce da un evento, alcune istruzioni vengono utilizzate solo nel momento in cui accade qualcosa (es. viene cliccato un pulsante con il mouse). Inoltre, si può verificare un parallelismo, più istruzioni possono essere eseguite contemporaneamente. Nel linguaggio si chiamano operatori le espressioni logiche e matematiche utilizzate all'interno delle istruzioni e si parla di dati quando si analizzano, memorizzano e rappresentano informazioni.

I "modi di lavorare, pensare e approcciare i problemi" nel mondo informatico e, di conseguenza, nell'ambito del coding sono incrementali e interattivi. È molto più semplice lavorare in modo incrementale su un progetto, testandolo e variandolo, piuttosto che tentare di realizzarlo perfetto al primo tentativo. Fondamentale è lavorare per testing e debugging, in quanto per determinare se un programma informatico funzioni è utile provarlo alla ricerca di eventuali errori e cercare soluzioni per correggerli. Si usa molto il riuso e il mixing poiché risulta più comodo riutilizzare del codice già scritto (anche da altri) piuttosto che scrivere ogni volta un programma da zero. Infine, si mettono in campo attenzione ad efficienza, calcolabilità e complessità. Il risultato, cioè la soluzione di un problema, deve essere il migliore ma usando meno risorse possibili. A questi si aggiunge la scomposizione, cioè giungere alla soluzione dell'intero problema suddividendolo in parti più semplici e facilmente risolvibili, l'astrazione, cioè concentrarsi sugli aspetti importanti per la soluzione ed il riconoscimento di pattern, generalizzazione, cioè riconoscere che alcune parti della soluzione possono essere utilizzate in problemi simili.

2.2 Pensiero computazionale

Il pensiero computazionale [2] è un concetto coniato nel 2006 dalla docente ed informatica dell'MIT Jeanette Wing, sebbene il termine fosse stato già introdotto negli anni '80 da Seymour Papert, nel libro *Mindstorms*.

Sebbene non vi sia una definizione unica di pensiero computazionale, ci si può riferire ad esso come “L'insieme dei processi mentali coinvolti nella formulazione di un problema, nonché l'espressione della sua soluzione in modo che una macchina o un essere umano siano in grado di eseguirlo in modo efficace. È tutto ciò che viene prima della tecnologia informatica, pensato da un essere umano conscio della potenza dell'automazione” (Wing, 2010). A partire da questo punto di vista, è possibile concepire il pensiero computazionale quale attività mentale finalizzata alla formulazione di un problema che ammette una soluzione computazionale, cui si perviene attraverso l'intervento di una macchina o dell'essere umano, o dalla combinazione di entrambi.

Quando si parla di pensiero computazionale, un'interpretazione immediata potrebbe ricondurre alle parole “problema” e “soluzione”. Il pensiero computazionale, tuttavia, non definisce solo un problema matematicamente ben definito, le cui soluzioni siano completamente analizzabili (a titolo esemplificativo, un algoritmo o un programma), ma anche problemi del mondo reale, le cui soluzioni rientrano in un sistema molto più complesso e ampio. In quest'ottica, il pensiero computazionale si sovrappone al pensiero logico, attivando dei processi e delle procedure di pensiero che attingono alla creatività, alla ragione, alla logica e alla capacità interpretativa e critica.

Difatti Wing definiva il pensiero computazionale, o *computational thinking*, come un atteggiamento, un set di capacità che non dovrebbero avere ed utilizzare solo gli informatici, ma tutti. Egli nel suo articolo inoltre diceva “Il pensiero computazionale è un processo di formulazione di problemi e di soluzioni in una forma che sia eseguibile da un agente che processi informazioni.”

Sappiamo infatti che i computer servono a risolvere i problemi, ma prima ancora di risolverli è importante capire con quali mezzi è possibile farlo. Ciò che ce lo permette è proprio il pensiero computazionale.

In breve, il pensiero computazionale è un metodo di **problem-solving** che identifica un problema complesso, lo comprende, lo riformula in modo semplificato per sviluppare possibili soluzioni e successivamente le presenta in un modo che possano essere comprese sia da un umano che da una macchina.

L'esecutore delle istruzioni non conosce il significato delle stesse, ed elabora dati di cui non conosce il significato, per questo si parla di esecuzione meccanica, che chiunque può eseguire solo applicando le istruzioni, la cui espressione ed efficacia sono della massima importanza. Quindi, per Wing, usare il pensiero computazionale significa pensare come un programmatore, riducendo, trasformando, simulando ed applicando i principi appartenenti all'informatica per la formulazione di istruzioni che portino alla risoluzione di problemi logici complessi.

Il termine computazionale in realtà deriva dall'informatica, dalla parola "computing", ovvero legato al calcolo, al computare, perché il pensiero computazionale richiede le capacità, i processi di pensiero e le pratiche di computazione di chi opera e studia nell'informatica, quindi il modo di pensare richiesto per la comprensione di un problema e la progettazione delle computazioni attraverso le quali un computer o un elaboratore di informazioni siano in grado di funzionare in modo efficiente per arrivare a una soluzione definita. Wing affermava che il pensiero computazionale integrasse e combinasse il pensiero matematico ed ingegneristico con il pensiero creativo.

Il pensiero computazionale è definito come un processo iterativo con fasi o tecniche molto specifiche che portano a formulare soluzioni a problemi in modo che le macchine o gli esseri umani possano risolverli. Secondo l'International Society for Technology in Education, il pensiero computazionale consente di:

- Rappresentare i dati del problema attraverso modelli specifici;
- Organizzare in maniera logica i dati del problema;
- Formulare e analizzare i problemi così che essi possano essere risolti mediante un esecutore (un umano o un computer);
- Segmentare le soluzioni in sequenze di step ordinati ed accuratamente descritti, automatizzandoli mediante algoritmi;
- Identificare le possibili soluzioni per implementare la più efficiente ed efficace in termine di risorse ed effort;
- Astrarre tali processi per la risoluzione di problemi simili;

Il pensiero computazionale, quindi, non è altro che un processo logico-creativo che porta al pensiero algoritmico a più livelli di astrazione, scomponendo un problema e riducendolo nelle sue parti distinte, risolvendone una per volta in modo da risolvere l'intero problema.

Generalmente, si tende a descrivere il pensiero computazionale attraverso tre fasi principali:

- **Astrazione**: si intende la formulazione del problema;
- **Automazione**: indica l'espressione della soluzione;
- **Analisi**: comprende esecuzione della soluzione e valutazione.

Come si può constatare, nessuna di queste fasi menziona esplicitamente la sfera informatica, né la robotica o le soluzioni eseguite da computer. Il pensiero computazionale può essere utilizzato nel coding, nella programmazione e nella robotica educativa, utilizzando gli strumenti di queste discipline, ma anche per normali situazioni di vita che comportano la scomposizione di problemi, l'espressione di soluzioni attraverso istruzioni e l'esecuzione di esse: individuare il percorso più breve per raggiungere un luogo, l'esecuzione di una ricetta, assemblare un oggetto, incastrare tra loro i mattoncini LEGO. Questi sono tutti processi che coinvolgono il pensiero computazionale e presuppongono una serie di istruzioni chiare, precise, ordinate e ripetibili che consentono il raggiungimento di una soluzione efficace a chiunque le stia eseguendo. Queste istruzioni, di fatto, rappresentano un algoritmo, in quanto se osservate alla lettera porteranno certamente alla soluzione e potranno essere applicate con lo stesso risultato ad un altro problema identico.

Come abbiamo visto, il pensiero computazionale consente mediante i concetti dell'informatica di formulare e risolvere problemi, definire sistemi e comprendere il comportamento umano. Ciò è dovuto allo sviluppo di una serie di strumenti e processi psicologici e intellettuali strettamente legati all'informatica, come la certezza di affrontare la complessità dei problemi e l'accettazione dell'ambiguità, conciliandola con il rigore dei processi mentali, ma che sfociano anche nell'ambito della comunicazione, della dimensione umana e della creatività.

Mitchel Resnick, direttore del "Lifelong Kindergarten" al MIT di Boston e "LEGO Papert Professor of Learning Research", propone il modello della spirale dell'apprendimento creativo (Figura 2.1), come ci spiega Passantino:

- **Imagine** (Immaginare): usare la fantasia;
- **Create** (Creare): trasformare in azione le idee;
- **Play** (Giocare): provare, sperimentare le creazioni;
- **Share** (Condividere): collaborare alle creazioni degli altri, raccontare, documentare;
- **Reflect** (Riflettere): valutare, ragionare, migliorare la creazione;

- **Imagine** (Immaginare): immaginare idee nuove e trarre spunti nuovi.

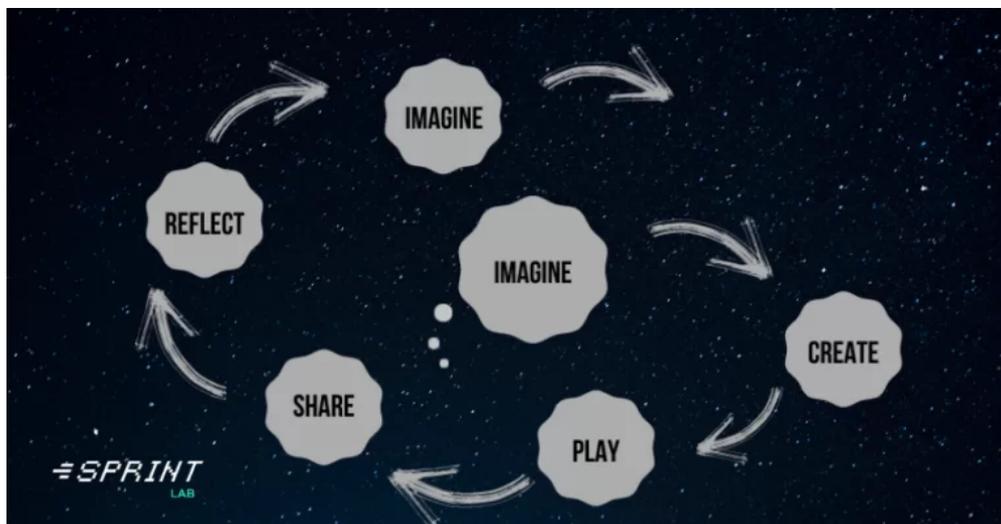


Figura 2.1: Modello della spirale dell'apprendimento creativo

Resnick, inoltre, ha individuato quattro spunti per aiutare gli studenti a pensare in modo creativo, incoraggiando chi lavora su progetti basati sulle passioni, in collaborazione tra loro e in uno spirito giocoso, ispirando il modello delle “quattro P dell'apprendimento creativo”:

- **Projects** (Progetti): creazione di progetti, passando in maniera continuativa attraverso la spirale dell'apprendimento creativo e sviluppando una più profonda comprensione del processo;
- **Passion** (Passione): quando le persone lavorano su progetti a cui tengono, sono disposte a lavorare più duramente e più a lungo. Si possono sviluppare diversi progetti (giochi, musica, animazioni, arte, storie, ecc.) ed ognuno può lavorare sui propri interessi;
- **Peers** (Pari): la creatività come processo di gruppo, persone che contribuiscono e condividono il proprio lavoro partendo da quello degli altri o combinandone le parti. Quindi, integrando la programmazione con una comunità online.
- **Play** (Giocare): il software incoraggia gli esperimenti, percorsi creativi che incoraggiano gli studenti a sperimentare nuove cose.

2.3 Coding e pensiero computazionale: l'importanza nelle scuole

Il pensiero computazionale è dunque strettamente correlato all'informatica ed è fondamentale svilupparlo sin dall'infanzia. Per farlo, uno dei modi più efficaci è quello di coltivarlo attraverso l'uso di strumenti del coding, ovvero il processo di scrittura di linguaggi ed istruzioni destinate alle macchine.

Negli ultimi anni molte scuole, comprese le scuole materne ed elementari, hanno incorporato il coding nelle attività svolte in classe per sviluppare la capacità dei bambini di programmare e imparare i linguaggi di programmazione in modo semplice e divertente, sia in modalità "unplugged", cioè senza l'uso di strumenti digitali, che con computer, tablet e software facili da usare e comprensibili anche per bambini più piccoli, giochi interattivi e robotica educativa, programmi di programmazione visiva che permettono di programmare in modo semplice andando poi a vedere i risultati del proprio processo mentale.

Il coding comprende tutti i processi fondamentali del pensiero computazionale, ovvero quei percorsi mentali che portano alla soluzione di un problema mediante l'utilizzo di specifici strumenti intellettuali. Esso è uno strumento fondamentale per lo sviluppo del pensiero computazionale ed è esso stesso correlato alla linguistica dell'informatica in quanto insegna i fondamenti del problem solving e del pensiero logico, a suddividere e segmentare un problema in parti più semplici che consentano di trovare una soluzione.

Il coding rappresenta inoltre un modo pratico ed immediato di applicare la teoria del pensiero computazionale insieme alle sue fasi e strumenti illustrati precedentemente divertendosi, facendo spazio alla fantasia e alla creatività, imparando allo stesso tempo un nuovo linguaggio ed un nuovo modo di vedere i problemi e le situazioni esprimendo in modo chiaro ed efficace le proprie soluzioni e le proprie idee, andando a sviluppare così anche la propria intelligenza e pensiero critico.

Jeannette Wing riteneva che il pensiero computazionale dovesse essere aggiunto alle capacità analitiche di ogni bambino insieme alle capacità di scrittura, lettura e calcolo, il quale deve essere in grado di pensare ad un livello astratto e combinare il rigore con il processo creativo. È importante che gli insegnanti aiutino gli studenti a sviluppare capacità di problem solving e di pensiero critico, invitandoli a scomporre i problemi e costruire ragionamenti per trovare soluzioni chiare, efficaci e semplificate. Da qui l'importanza di includere la programmazione ed il pensiero computazionale nella scuola dell'infanzia.

Imparare il pensiero computazionale è una competenza fondamentale per i bambini, non tanto per destreggiarsi nell'informatica o per imparare a programmare, ma per poter sviluppare strumenti logici e mentali cruciali all'apprendimento, alla risoluzione dei problemi, da utilizzare sia nella vita quotidiana che nello studio.

Come detto in precedenza il coding è una delle vie più divertenti, stimolanti ed immediate e può essere introdotto nelle lezioni per sviluppare il pensiero computazionale negli studenti. Anche Seymour Papert, padre del pensiero computazionale ed uno dei primi docenti ad applicare il coding anche alla didattica, sosteneva ciò, di fatti, riconosceva come questo fosse il mezzo ideale per mettere i bambini di fronte ad un problema da risolvere in maniera diretta ed efficace ed il computer uno strumento importante per l'apprendimento.

Oggigiorno, per insegnare il coding ai ragazzi, vi è una grande quantità di programmi a disposizione dei genitori o degli insegnanti, da inserire nello studio giornaliero e nella didattica per migliorare sia le capacità informatiche ma anche quelle cognitive. L'obiettivo di questi programmi non è necessariamente quello di educare al coding quanto quello di favorire nei ragazzi l'attitudine al problem solving e al pensiero critico.

Nei documenti ufficiali della Commissione Europea si afferma che l'acquisizione di competenze digitali, incluso il coding, è fondamentale per lo sviluppo, la competitività e per il mercato del lavoro, ma l'educazione dovrebbe invece puntare sullo sviluppo di capacità generali relative ad identificazione, rappresentazione, comunicazione e condivisione di situazioni problematiche e relative soluzioni, per poter far fronte alla nostra società in continua trasformazione.

Per questo motivo, bisogna essere consapevoli di tutte le sue potenzialità trasversali, in quanto favorisce lo sviluppo della creatività perché un problema può essere affrontato e risolto in differenti modi. Il coding è proficuo perché produce risultati visibili. Apprenderlo aiuta a saper affrontare la complessità perché imparare a risolvere problemi informatici aiuta ad essere capaci di risolvere problemi in altri ambiti. Inoltre, contribuisce allo sviluppo della precisione poiché un programma per funzionare bene necessita della correttezza di tutte le istruzioni e quindi di individuare e correggere eventuali errori.

Inoltre, programmare è un potente strumento di pensiero. Programmare un computer per fargli risolvere un problema richiede di aver analizzato il problema ed aver pianificato in modo dettagliato una soluzione, e ciò permette di apprendere qualcosa di nuovo anche indipendentemente dalla programmazione stessa. L'unico

vincolo è che se si vuole programmare per imparare è necessario prima aver imparato a programmare.

Un ulteriore qualità è che può essere un potentissimo strumento di espressione personale poiché programmare permette di produrre qualcosa di personale in modo creativo, da una presentazione, al videogioco, all'animazione digitale. Infine, può essere uno strumento di crescita personale, quando si programma si procede in modo incrementale, cioè si scrive del codice, si verifica se viene correttamente eseguito e, se così non avviene, si modifica il codice, procedendo in modo ciclico (codice-esecuzione-codice) per perfezionare continuamente il programma. Operare in questo modo permette di sviluppare l'intelligenza, grazie all'impegno nell'affrontare le difficoltà, e una forma mentis aperta all'imprevisto, che vede l'imperfezione e l'errore come occasione per migliorare.

Non è un caso, infatti, che "Learn to code, code to learn" sia lo slogan di Scratch, uno dei più noti programmi utilizzati per il coding, a significare che imparando come si programma "si impara ad imparare": si apprendono strategie per risolvere problemi, fare progetti e comunicare idee, utili a tutti, in modo indipendente dall'età, dall'occupazione o dal livello socioeconomico-culturale.

Appresa l'importanza del coding, il suo apprendimento risulta molto importante anche nella scuola primaria e dell'infanzia. I motivi per cui si dovrebbe insegnare ai bambini a programmare sono molteplici. Il primo motivo che possiamo includere è la forte capacità dei bambini di apprendere rapidamente nuovi concetti, infatti, i bambini sono in grado di familiarizzare con nuovi concetti in un periodo di tempo più breve rispetto agli adulti. Questo è il motivo per cui sono indotti fin dalla tenera età ad apprendere una nuova lingua straniera. Se a questa particolare propensione all'apprendimento dei bambini si aggiungono le caratteristiche ludiche dei vari programmi educativi di coding, allora oltre a portare grandi vantaggi, l'educazione dei bambini sarà anche svolta in maniera più semplice. Il coding li favorirà nel processo di apprendimento dei linguaggi di programmazione e della gestione di programmi, capacità indispensabili nel mondo del lavoro ma anche nella vita di tutti i giorni.

Altra ragione per cui è fondamentale promuovere l'insegnamento del coding nelle scuole è legata all'importanza dello sviluppo del pensiero computazionale. Infatti, l'importanza del pensiero computazionale è stata riconosciuta anche a livello istituzionale nel 2015, anno in cui il Ministero dell'Istruzione ha lanciato il "Piano Nazionale della Scuola Digitale", che comprende il progetto "Programma il Futuro", il cui obiettivo è includere lo sviluppo delle competenze del pensiero computazionale nelle scuole per i più giovani.

Un altro vantaggio deriva proprio dalla possibilità di sviluppare abilità sociali e relazionali importanti come il lavoro di squadra attraverso programmi di coding educativi. Inoltre, se consideriamo l'enorme impatto della rivoluzione tecnologica sul mondo del lavoro, l'apprendimento del coding e dei linguaggi di programmazione può dare ai bambini una migliore comprensione di come funzionano i computer e la tecnologia fin dalla tenera età. Infine, imparare un linguaggio di programmazione è un ottimo esercizio per i bambini per imparare una nuova lingua e migliorare le proprie capacità di risoluzione dei problemi e degli errori, il tutto divertendosi.

È proprio grazie al divertimento che è possibile far apprendere ai bambini il linguaggio di programmazione. Difatti, ci sono molte persone che non posseggono alcun concetto sui linguaggi di programmazione, o che eseguono operazioni banali o di intrattenimento utilizzando a stento la tecnologia a disposizione. Online, però, si possono trovare numerosi giochi e linguaggi di programmazione per bambini studiati appositamente per favorire l'apprendimento del coding nella scuola primaria e non solo, riuscendo a sfruttare l'aspetto ludico che li contraddistingue.

2.4 Coding Unplugged

Quando si parla di coding unplugged [3] si intendono quelle attività che utilizzano strumenti non digitali per la realizzazione di attività che introducono ai concetti fondamentali dell'informatica e alle logiche della programmazione. Esso si basa sugli stessi principi del coding ma nel modo di operare "unplugged", ossia senza l'ausilio dell'elettricità, e quindi senza computer o altri dispositivi elettronici.

Sono sempre più numerose le esperienze didattiche che vedono coinvolti bambini e ragazzi nella realizzazione di attività analogiche che affiancano l'utilizzo degli strumenti digitali. Il coding unplugged è adatto per persone di ogni età ma è particolarmente interessante per gli studenti dall'infanzia alla scuola secondaria di primo grado. Questi tipi di attività, infatti, sono caratterizzate da elementi che stimolano la curiosità e l'apprendimento degli alunni. Fare coding a scuola, ma senza computer. Sembrerebbe un ossimoro, o quanto meno una contraddizione in termini, eppure non solo questo è possibile, ma è anche molto utile nel processo mirato all'acquisizione del pensiero computazionale.

La programmazione presuppone che ci sia un esecutore ideale in grado di leggere, interpretare ed eseguire istruzioni tratte da un repertorio finito e non ambiguo. Quando l'esecutore ideale è una macchina (computer, smartphone, robot, ...) il programma viene eseguito in modo automatico e l'esito dell'esecuzione ne rivela la correttezza. Nel coding unplugged il ruolo dell'esecutore ideale è svolto da una

persona, che deve comportarsi come un robot eseguendo solo ed esclusivamente le istruzioni scritte nel programma.

La didattica unplugged può rappresentare un punto di vista privilegiato per comprendere l'architettura e la logica delle macchine e per interpretare l'ambiente circostante come un linguaggio programmabile con il quale giocare e apprendere. Essa presenta numerosi vantaggi:

- non pone barriere tecnologiche, può essere praticato dovunque con materiali fai-da-te o, addirittura, senza alcun materiale;
- consente ai ragazzi di saper scrivere le tecnologie e non solo di leggerle;
- rende istruttiva sia la scrittura del codice che la sua interpretazione ed esecuzione;
- consente di porre tutta l'attenzione sul pensiero computazionale e sugli aspetti algoritmici, mettendo in secondo piano gli strumenti tecnologici e sviluppando competenze logiche;
- richiede una pianificazione di passi da svolgere, serve coerenza;
- è un atto creativo, in quanto l'alunno è stimolato a creare un prodotto attraverso il proprio ragionamento, con le proprie idee;
- può essere praticato a qualsiasi età, come gioco da tavolo, gioco di società, attività di gruppo o attività motoria;
- a chi svolge il ruolo di esecutore ideale permette di comprendere a fondo la natura di computer e robot;
- pone l'attenzione sulla separazione di ruoli tra programmatore ed esecutore ideale, che è l'essenza del coding.

Ma presenta uno svantaggio: non permette di verificare automaticamente la correttezza di un programma, la cui esecuzione è comunque affidata all'essere umano che svolge il ruolo di esecutore, muovendo una pedina o eseguendo in prima persona le istruzioni. Questo limite intrinseco, nella pratica del coding unplugged offre ulteriori opportunità di gioco, di sfida e di confronto. Tipicamente, nei giochi competitivi o collaborativi, sono le squadre avversarie o i propri compagni di squadra a dover prestare l'attenzione necessaria a verificare la correttezza di ogni mossa.

Per fare coding unplugged ci sono diversi strumenti, può bastare un foglio di carta a quadretti, qualche matita colorata e tanta fantasia. Ma ci si può spingere anche oltre, integrando il coding con l'educazione motoria, quella artistica ecc.

2.5 Pixel-art

La Pixel-art [4] (Figura 2.2) è una delle attività di coding unplugged più utilizzate per l'introduzione del pensiero computazionale e l'insegnamento della programmazione nella scuola dell'infanzia e primaria.

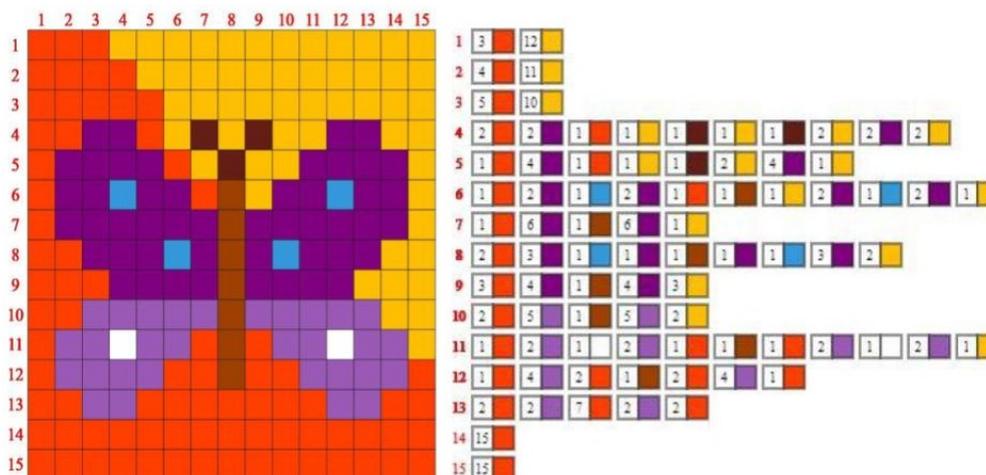


Figura 2.2: Pixel-art

I computer per rappresentare le immagini devono necessariamente costruire una griglia e colorare i quadretti. Non sanno fare diversamente. Ogni quadretto è un pixel. Chiamiamo Pixel-art ogni disegno che mette in risalto la struttura a quadretti e ne fa un espediente artistico, facendo di necessità virtù.

Tanto più piccoli e numerosi sono i pixel, tanto meno evidente è la quadrettatura e tanto più continua e definita ci appare l'immagine. Moltiplicando il numero di righe con il numero di colonne si ottiene il numero di pixel che costituiscono un'immagine. I video ad alta risoluzione per esempio hanno 1920 colonne e 1080 righe, quindi circa 2 milioni di pixel. Gli schermi e le fotocamere dei nostri cellulari, tablet, computer, anche di più.

Una volta definita la griglia e scelto il punto di partenza (per convenzione, il pixel in alto a sinistra) e l'ordine (per convenzione la scansione per righe da sinistra a destra e dall'alto al basso), per la rappresentazione dell'immagine basta dire il colore di ogni pixel. Nella rappresentazione del codice-colore è possibile fare riferimento al cambio della riga oppure scegliere di non rappresentarlo, poiché si decide una volta per tutte che quando si arriva alla fine si ricominci dall'inizio della riga successiva, esattamente come quando si legge. La sequenza di colori per ciascun pixel è semplicemente una descrizione dell'immagine. Non è una procedura

per disegnarla ma per fare ciò è necessario spiegare come usarla. Ad esempio, partendo dal pixel più in alto a sinistra e procedendo per righe, leggendo un colore finché sono presenti nella sequenza, questo si usa per colorare il pixel su cui ci si trova e successivamente si va avanti.

Esistono anche descrizioni più compatte. In particolare la codifica RLE (run length encoding) usa il concetto di “run” per evitare di ripetere più volte di seguito lo stesso colore. Un "run" non è altro che una serie di pixel consecutivi dello stesso colore, rappresentati indicando il colore una sola volta e specificando per quanti pixel esso va mantenuto.

L'utilizzo della Pixel-art come attività di coding unplugged trae ispirazione dai formati di descrizione delle immagini adottando convenzioni e descrivendo le immagini in modo coerente. In pratica, come se questa procedura facesse parte di una convenzione adottata una volta per tutte e la descrizione dell'immagine permettesse all'esecutore di ricostruirla. In termini informatici, la descrizione di un'immagine è dichiarativa.

Possiamo utilizzare un approccio imperativo, dotandoci di una serie di istruzioni che descrivono in maniera diretta come disegnare un'immagine. In questo caso, la descrizione dell'immagine e le azioni eseguite per riprodurla, sono un tutt'uno.

Il disegno generato dalla colorazione dei pixel è denominato pixelmap o bitmap, caratterizzato da una risoluzione predefinita. Per contro, un disegno potrebbe essere descritto senza limiti di risoluzione come combinazione di figure geometriche e curve di cui siano dati i parametri o le regole di costruzione. Se si vuole disegnare una circonferenza basta indicare il centro, il raggio, lo spessore della linea ed il colore. Qualunque sia la risoluzione del disegno, con questa descrizione è possibile sempre ricostruirla sfruttandola pienamente. Questa tecnica si chiama grafica vettoriale. Da un disegno vettoriale è sempre possibile generare una bitmap di qualsiasi risoluzione. Da una bitmap, invece, non è possibile rigenerare un disegno vettoriale o una bitmap di risoluzione più alta.

Capitolo 3

Tecnologie

La piattaforma PixelCodingStudio è un'applicazione web, termine utilizzato per riferirsi alle applicazioni accessibili e fruibili attraverso il browser web con una connessione attiva, che offre determinati servizi all'utente. Essa pertanto è progettata seguendo le logiche di sviluppo delle applicazioni web implementando un'architettura suddivisa in tre livelli.

Al primo livello, lato client è assegnata la parte grafica, il browser si occupa di creare le pagine che l'utente visualizzerà in seguito. Al secondo livello vi è la parte server, che gestisce la logica di amministrazione del software oltre ad elaborare ed eseguire le richieste ricevute dal client. Il terzo livello è costituito dalla base dati, mediante il quale vengono memorizzate e lette le informazioni utili alla piattaforma per operare.

L'ambiente di sviluppo di questa piattaforma è costituito da diversi linguaggi di programmazione, librerie e framework che verranno spiegati in maniera più dettagliata in seguito. Per la scrittura del codice è stato utilizzato come editor di codice Visual Studio Code.

La parte di front-end è sviluppata con HTML, CSS e JavaScript, che insieme permettono la scrittura, l'impaginazione e la gestione degli eventi delle pagine web progettate per l'esperienza dell'utente. La parte di back-end è sviluppata in Python attraverso l'utilizzo del framework Django, che rende il processo lavorativo per la creazione di un'applicazione web complessa, più semplice e veloce. La parte del database è gestita dal database relazionale MySQL per una facile raccolta e recupero dei dati.

L'intera applicazione è gestita tramite la tecnologia di virtualizzazione messa a disposizione da Docker per eseguire e far comunicare tra loro i diversi processi informatici in un ambiente isolato.

3.1 Docker

Docker [5] è una piattaforma di containerizzazione open source. Consente agli sviluppatori di impacchettare le applicazioni in container, che sono componenti eseguibili standardizzati che combinano il codice sorgente dell'applicazione con le dipendenze del sistema operativo e le librerie necessarie per eseguire quel codice in qualsiasi ambiente. I container semplificano la distribuzione di applicazioni distribuite. Docker rende più semplice, facile e sicuro creare, implementare e gestire dei container anche se gli sviluppatori possono creare dei container senza questa piattaforma. Docker è essenzialmente un toolkit che permette agli sviluppatori di creare, distribuire, eseguire, aggiornare e arrestare i container utilizzando comandi semplici e un'automazione a basso sforzo attraverso un'unica API.

I container sono resi possibili dalle funzionalità di isolamento e virtualizzazione dei processi integrate nel kernel Linux. Queste funzionalità, come i namespace per limitare l'accesso o la visibilità di un processo in altre aree di sistema o risorse ed i gruppi di controllo (Cgroup) per l'allocazione delle risorse tra i processi, permettono a più componenti dell'applicazione di condividere le risorse di una singola istanza del sistema operativo host nella stessa maniera in cui hypervisor abilita più VM (virtual machine) a condividere CPU, memoria e altre risorse di un singolo server hardware. Pertanto, la tecnologia dei container offre tutte le funzionalità e i vantaggi delle macchine virtuali, incluso l'isolamento delle applicazioni ed una scalabilità efficiente in termini di costi. Conveniente è l'eliminabilità, nonché importanti vantaggi aggiuntivi: maggiore efficienza delle risorse; aumento della produttività degli sviluppatori; peso più leggero.

Docker si avvale di diversi strumenti assemblati in una particolare architettura (Figura 3.1) che verranno citati di seguito.

Ogni contenitore Docker inizia con un semplice file di testo che contiene le istruzioni su come creare l'immagine del container Docker. **DockerFile** automatizza il processo di creazione di immagini Docker. È essenzialmente un elenco di istruzioni dell'interfaccia della riga di comando (CLI) che il motore Docker eseguirà per assemblare l'immagine.

Un'**immagine Docker** contiene il codice sorgente di un'applicazione eseguibile e tutte le librerie, gli strumenti e le dipendenze necessarie per l'esecuzione del codice dell'applicazione come container. Quando si esegue un'immagine Docker, questa diventa un'istanza (o più istanze) di un container. Si possono creare immagini Docker da zero, ma la maggior parte degli sviluppatori le estrae da repository pubblici. Da una singola immagine di base, si possono creare più immagini Docker

che andranno a condividere elementi comuni del loro stack. Un'immagine Docker è composta da livelli, ognuno dei quali corrisponde a una versione dell'immagine. Ogni volta che uno sviluppatore apporta una modifica a un'immagine, viene creato un nuovo livello superiore per sostituire quello precedente come versione corrente dell'immagine. I livelli precedenti vengono salvati per il riutilizzo in altri progetti o per il rollback.

Ogni volta che viene creato un container da un'immagine Docker, viene creato un altro nuovo livello chiamato livello container. Le modifiche apportate al container, come l'aggiunta o la rimozione di file, vengono mantenute solo nel livello container ed esistono solo mentre è in esecuzione il container. Questo processo iterativo di creazione di immagini migliora l'efficienza complessiva, poiché possono essere eseguite più istanze container dal vivo partendo da una singola immagine di base, quando ciò avviene, esse sfruttano uno stack comune.

Un **container Docker** è un'istanza di un'immagine Docker che viene eseguita in tempo reale. Le immagini Docker sono file di sola lettura, mentre i container sono contenuti eseguibili dal vivo e temporanei. Gli utenti possono interagire con esso e gli amministratori possono modificarne le impostazioni e le condizioni utilizzando i comandi docker.

Il **daemon Docker** è un servizio che viene eseguito sul proprio sistema operativo, come Microsoft Windows, Apple MacOS o iOS. Questo servizio crea e gestisce le proprie immagini Docker per proprio conto utilizzando i comandi dal client, agendo come centro di controllo della propria implementazione di Docker.

Il **registro Docker** è un sistema di archiviazione e distribuzione di immagini Docker open source e scalabile. Il registro permette di tenere traccia delle versioni delle immagini nei repository, utilizzando l'assegnazione di tag per l'identificazione. Questa operazione viene eseguita utilizzando uno strumento di controllo delle versioni (git).

Il Docker dispone inoltre di **Compose**, uno strumento che consente l'esecuzione di applicazioni facenti uso di svariati container. Questo strumento utilizza un apposito file di configurazione di tipo YAML contenente l'elenco dei container e dei volumi di cui l'applicativo necessita. YAML è un formato per la memorizzazione di dati facilmente utilizzabile da macchine ed esseri umani. Questo file è utilizzabile su qualunque altra macchina su cui sia installato Compose, al pari dei container da esso generati. Nel file vengono inoltre memorizzate le informazioni necessarie per la creazione del sistema distribuito. Per ogni container vengono specificati le porte utilizzate per la comunicazione, il nome ed eventuali dipendenze da altri container,

quali ad esempio le unità destinate ad ospitare i database. Per la preparazione dell'ambiente di esecuzione viene lanciato il comando di compilazione del file di configurazione. Questa operazione elabora il file creando i container ed i volumi in esso specificati, successivamente vengono processati i singoli Dockerfile per l'installazione degli applicativi nei rispettivi container. Al termine della procedura, è possibile lanciare il comando per l'esecuzione vera e propria dell'applicazione e di tutti i servizi ad essa collegati precedentemente definiti. Compose consente quindi di creare un sistema distribuito virtuale utilizzando un singolo host fisico, ogni container può essere visto come un processo a sé stante.

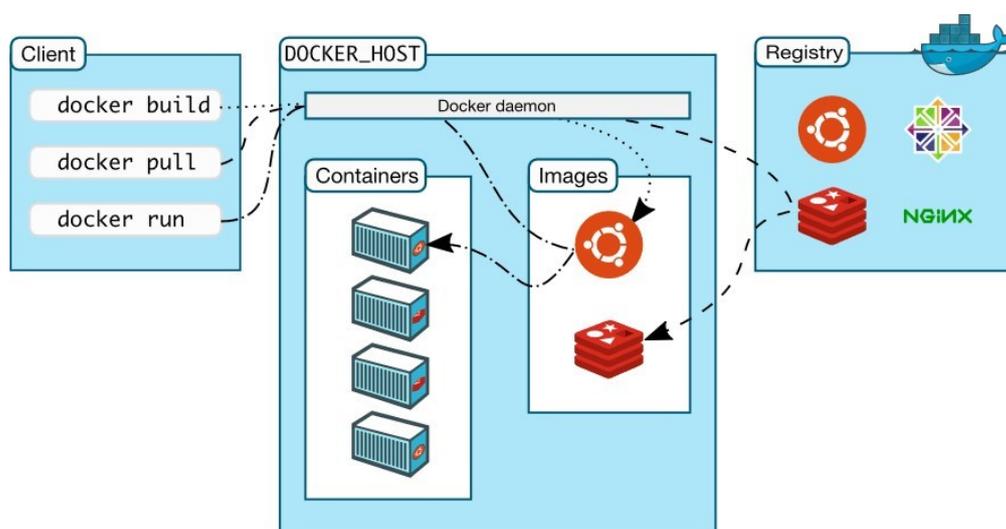


Figura 3.1: Architettura Docker

In questa piattaforma vengono creati tre container differenti:

- **unplugged:** è il container più importante che garantisce l'esecuzione dell'intera applicazione. Esso gestisce tutta la parte di back-end e front-end del sito e la relativa comunicazione.
- **unpluggeddb:** è un container che gestisce la configurazione di MySQL come base dati dell'applicazione.
- **phpmyadmin:** è il container che ospita l'applicazione web phpMyAdmin, un sistema che permette la gestione e l'amministrazione del database MySQL attraverso un qualsiasi browser.

3.2 Python

Il linguaggio Python [6], inventato a partire dal 1991 da Guido van Rossum , è un linguaggio di programmazione di "alto livello", orientato a oggetti, che supporta pienamente la programmazione strutturata. Esso è progettato appositamente per garantire una semplicità estrema nella codifica del programma e al tempo stesso una grandissima leggibilità del codice. Python è un linguaggio che si presta molto bene per usi didattici, ma al tempo stesso è un linguaggio potentissimo, con il quale è possibile scrivere software estremamente complessi, tanto da essere adottato tra i linguaggi di programmazione standard per l'implementazione di servizi da aziende come Google e la NASA, oltre a moltissime altre aziende in Italia che all'estero. Tra gli altri usi, è adatto a sviluppare applicazioni distribuite, scripting, computazione numerica e system testing.

Le caratteristiche più immediatamente riconoscibili di Python sono le variabili non tipizzate e l'uso dell'indentazione per la sintassi delle specifiche, al posto delle più comuni parentesi. Altre caratteristiche distintive sono l'overloading di operatori e funzioni tramite delegati, la presenza di un ricco assortimento di tipi e funzioni di base e librerie standard, sintassi avanzate quali slicing e list comprehension. Il controllo dei tipi è forte (strong typing) e viene eseguito in runtime (dynamic typing): una variabile è un contenitore a cui viene associata un'etichetta (il nome) che può essere associata a diversi contenitori anche di tipo diverso durante il suo tempo di vita. Fa parte di Python un sistema garbage collector per liberazione e recupero automatico della memoria di lavoro.

Come descritto in precedenza Python è stato utilizzato per la gestione del back-end dell'applicazione attraverso il framework Django che si basa sull'uso di questo linguaggio di programmazione.

3.3 OpenCV

OpenCV [7] (Open Source Computer Vision Library) è una libreria di software open source per la visione artificiale e l'apprendimento automatico originariamente sviluppata da Intel. Il linguaggio di programmazione principalmente utilizzato per sviluppare con questa libreria è il C++, ma è possibile interfacciarsi anche attraverso il C, Python e Java. OpenCV è stato creato per fornire un'infrastruttura comune per le applicazioni di visione artificiale e per accelerare l'uso della percezione della macchina nei prodotti commerciali. La libreria semplifica l'utilizzo e la modifica del codice da parte degli utenti.

Dispone di oltre 2500 algoritmi ottimizzati, che includono un set completo di algoritmi di computer vision e machine learning sia classici che all'avanguardia. Questi algoritmi possono per esempio essere utilizzati per rilevare e riconoscere volti, identificare oggetti, classificare azioni umane nei video, tenere traccia dei movimenti della telecamera, tracciare oggetti in movimento, estrarre modelli 3D di oggetti, trovare immagini simili da un database di immagini, rimuovere gli occhi rossi dalle immagini scattate con il flash, seguire i movimenti degli occhi, ecc.

Nell'applicazione web questa libreria è stata utilizzata per il processamento dell'immagine caricata dall'utente e per l'utilizzo dell'algoritmo di clustering K-means.

3.4 Scikit-learn

Scikit-learn [8] è una libreria open source che fornisce una gamma di algoritmi di apprendimento supervisionati e non supervisionati tramite un'interfaccia python. È stata inizialmente sviluppata da David Cournapeau come progetto di Google Summer of Code nel 2007. La prima versione è stata pubblicata a fine gennaio 2010.

La libreria è basata su SciPy (Scientific Python): libreria fondamentale per il calcolo scientifico; si appoggia inoltre su altre librerie come NumPy: pacchetto di matrice n-dimensionale di base. La libreria è focalizzata sulla modellazione dei dati. Non si concentra sul caricamento, la manipolazione e il riepilogo dei dati. Contiene algoritmi di classificazione, regressione e clustering.

Questa libreria è stata utilizzata per eseguire un confronto tra i diversi algoritmi di clustering. In particolare sono stati utilizzati il K-medoids e il DBSCAN, non presenti in OpenCV.

3.5 xhtml2pdf

xhtml2pdf [9] è un modulo Python che consente agli utenti di generare documenti PDF dal contenuto HTML in modo semplice e con il controllo del flusso automatizzato come l'impaginazione e il mantenimento del testo. Il pacchetto supporta vari stili dei caratteri e riconosce molte opzioni di formattazione nei documenti HTML. xhtml2pdf può essere utilizzato in qualsiasi ambiente Python, incluso Django.

Nell'applicazione web il pacchetto è utilizzato per la generazione del PDF scaricabile dall'utente contenente il codice unplugged, la legenda dei colori, la griglia da colorare e la griglia colorata.

3.6 Django

Django [10] è un framework Web Python di alto livello che incoraggia uno sviluppo rapido e un design pulito e pragmatico. Nato nell'autunno del 2003 quando un gruppo di programmatori Web, Adrian Holovaty, Simon Willison, Jacob Kaplan-Moss e Wilson Miner, presso la redazione di un giornale (il "Lawrence Journal-World", per la cronaca), ha iniziato ad usare Python per costruire applicazioni Web e rilasciato con licenza open source nel Luglio 2005. Ciò ha permesso la crescita di una comunità internazionale che ha contribuito allo sviluppo del framework.

Django si occupa di gran parte del problema dello sviluppo web, così da concentrarsi sulla scrittura della propria applicazione senza dover reinventare la ruota. Sorprendentemente veloce è stato progettato per aiutare gli sviluppatori a portare le applicazioni dall'ideazione al completamento il più rapidamente possibile. Prende sul serio la sicurezza, infatti, per gli aspetti più comuni legati alla sicurezza in ambito web quali la prevenzione di attacchi di SQL injection e CSRF (Cross-site Request Forgery), sono implementati accorgimenti automatici. Inoltre, è estremamente scalabile.

Ogni applicazione basata su Django utilizza un file di configurazione contenente le impostazioni del progetto. Questo file contiene le configurazioni per la connessione ai database utilizzati e per la sicurezza dell'applicazione, in più vi sono le impostazioni relative all'autenticazione dell'utente e alle sessioni di accesso. È inoltre disponibile la configurazione delle operazioni del server relative all'invio di e-mail, all'utilizzo della cache e alla gestione dei caricamenti di file lato client.

Django si basa sul paradigma MTV Model-Template-View (Figura 3.2), il quale si discosta leggermente dal più noto MVC Model-View-Controller.

Il **Model** è un'insieme di classi Python per la rappresentazione del modello dei dati. Queste classi forniscono una rappresentazione per le tabelle del database e questo consente di utilizzare gli oggetti per effettuare le operazioni CRUD (Create, Read, Update, Delete) sui dati, al posto delle classiche query SQL. Si parla in questo caso di ORM, Object-Relational Mapping. Tutte le definizioni delle classi sono contenute nel file `models.py` nella cartella dell'applicazione.

La **View** si tratta di funzioni Python che gestiscono il flusso dell'applicazione e implementano la componente Controller del pattern MVC. Grazie a queste funzioni si possono definire le pagine all'interno dell'applicazione e i comportamenti che queste pagine avranno in funzione dell'interazione con l'utente. Lo scopo di una view è ricevere richieste HTTP e restituire risposte. Inoltre, è possibile prendere

una serie di dati dal Model, elaborarli e decidere quale Template utilizzare per mostrare i dati all'utente finale. In alternativa possiamo richiamare un'altra View, spedire una mail, etc.

Il **Template** rappresenta l'interfaccia grafica, insieme di file di testo che descrivono la presentazione dei dati implementando la componente View del pattern. Questa suddivisione consente a diverse figure professionali, di lavorare in modo indipendente alle parti dell'applicazione di propria competenza: gli sviluppatori si dedicheranno al modello dei dati e alle view, i Web designer ai template.

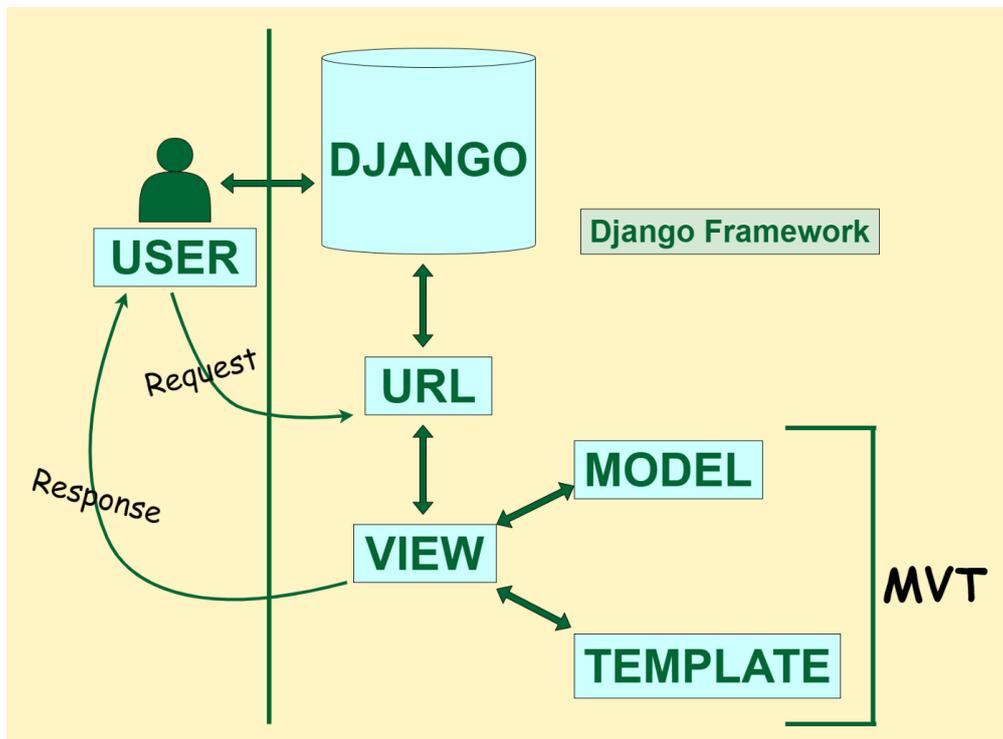


Figura 3.2: Logica di funzionamento Django

La peculiarità di questa architettura è che le tre componenti sono separate tra loro a livello di organizzazione di codice, ma sono molto collegate. Ad esempio, in una interazione classica con una piattaforma, sviluppata mediante Django, avviene prima una risoluzione dell'url tramite l'URLConf di django che in base all'url digitato richiama la view corrispondente insieme a dati aggiuntivi, se necessari. Successivamente dentro la view, vengono raccolti dei dati nel database, i quali vengono mandati al template corrispondente in modo tale da poter generare un front-end con dati dinamici.

3.7 Alternative a Django

I framework Web sono un aspetto essenziale dello sviluppo di app Web guidate dall'utente. Forniscono struttura alle applicazioni web, aumentando la produttività e l'efficienza degli sviluppatori oltre a ridurre i tempi di sviluppo consentendo agli sviluppatori di concentrarsi sulla logica dell'applicazione anziché sugli elementi di routine. Ma le nuove tendenze di sviluppo hanno portato a un numero enorme di framework e strumenti innovativi.

Visto che Python è un linguaggio sviluppato nei primi anni '90, la community di programmatori ha prodotto negli anni una quantità considerevole di framework per tale linguaggio. I framework Python si dividono in tre famiglie: full stack, micro framework e framework asincrono.

- I framework full-stack sono soluzioni complete per qualsiasi necessità di sviluppo. Tipicamente offrono funzionalità come convalida di moduli, generatori di moduli, layout di modelli e livelli di astrazione del database.
- I microframework, invece, seguono la filosofia opposta. Essi sono pacchetti ottimizzati per una funzione specifica al posto di offrire una suite completa di funzionalità. Piccoli e veloci, necessitano però l'aggiunta manuale delle funzionalità aggiuntive richieste.
- I framework asincroni, la cui popolarità è in ascesa, sono essenzialmente microframework che sfruttano la libreria asincrona di Python per gestire un ampio numero di connessioni simultanee.

Come detto in precedenza ci sono molte alternative a Django nel mercato ma esso rimane il framework Python più diffuso e più utilizzato al mondo. Di seguito andremo ad analizzare le due alternative più popolari. Flask che si contende il titolo di framework Python più diffuso e più utilizzato al mondo con Django. Molto meno utilizzato ma comunque al terzo posto vi è Web2py.

3.7.1 Flask

Flask [11] è un micro framework Python, ispirato da Sinatra, framework di Ruby. Disponibile con licenza BSD, Flask dipende dal toolkit WSGI Werkzeug e dal template Jinja2. L'architettura di Flask fornisce instradamento URL pronto all'uso, gestione di richieste ed errori, modelli, cookie, supporto per unit test, un debugger e un server di sviluppo. Rispetto a Django, tuttavia, non fornisce alcune funzionalità tipiche delle applicazioni Web, tra cui un ORM e la gestione dell'autenticazione e delle autorizzazioni. Spetta al programmatore scegliere come personalizzare l'applicazione utilizzando codice ad hoc o librerie di terze parti.

La prima grande differenza tra i due framework è la gestione del modello di dati. Django fornisce il proprio Django ORM (Object-Relational Mapping), mentre Flask no. I modelli di dati, soprattutto quando si lavora con database relazionali, consentono agli sviluppatori di connettere tabelle di database con classi di linguaggio di programmazione. In questo modo il programmatore utilizzerà le funzioni associate alle classi per accedere al database senza dover necessariamente scrivere i comandi SQL corrispondenti. Ciò semplifica notevolmente il codice, sebbene in alcuni casi possa essere limitante. Flask non fornisce un ORM anche se quasi tutte le applicazioni si interfacciano con un database perchè fa parte della sua filosofia: lo sviluppatore è libero di scegliere come interagire con il database senza alcuna imposizione dal framework.

Un'altra differenza significativa è l'uso da parte di Django del modello Model-Template-View, un aspetto che non è presente in Flask. L'uso di questo modello consente ai programmatori di partizionare il processo di sviluppo rendendo la struttura del codice più pulita e definita e migliorando i tempi di sviluppo. Flask, d'altra parte, dà allo sviluppatore la libertà di scegliere come strutturare il codice. Ciò significa che il team di sviluppo deve decidere in anticipo come strutturare il progetto e definire le linee guida per una buona programmazione.

Per questi motivi, Flask viene solitamente scelto per siti di più piccole dimensioni, come forum e blog personali, per prototipi o per applicazioni a pagina singola (SPA). Django, invece, grazie alla propria versatilità viene usato dai piccoli progetti personali ai siti web di livello enterprise.

3.7.2 Web2py

Web2py [12] è un framework full-stack scalabile per Python. Open source e senza dipendenze esterne, esso viene fornito insieme al proprio IDE web-based, che include un editor di codice, un debugger e la distribuzione attraverso un clic.

La semplicità d'uso di web2py è un obiettivo alla base della creazione di web2py, dall'apprendimento alla configurazione, dalla codifica alla distribuzione. Questo pur riuscendo ad offrire un set di funzionalità molto ampio e una grande personalizzazione.

Web2Py segue lo schema MVC e fornisce supporto per l'internazionalizzazione. Inoltre offre ottime funzionalità integrate di sicurezza dei dati che permettono di prevenire diverse vulnerabilità comuni, tra i quali difetti di injection, cross-site scripting ed esecuzione di file dannosi. Altra funzionalità specifica di Web2Py è il

sistema di ticketing, utilizzato per la registrazione completa degli errori e per il loro monitoraggio.

Web2py quindi copre sostanzialmente tutte le funzionalità di Django, poiché offre molte caratteristiche già pronte, ma potrebbe non essere intuitivamente facile nell'apprendimento e ancora nell'uso poco sviluppato, poiché è molto recente e ha una comunità più piccola rispetto a Django,

3.8 Django-SES

Django-SES [13] è un back-end di posta drop-in per Django . Invece di inviare e-mail tramite un tradizionale server di posta SMTP, Django-SES instrada le e-mail tramite l'eccellente Simple Email Service (SES) di Amazon Web Services.

Amazon SES è una piattaforma e-mail che offre un metodo semplice e conveniente per inviare e ricevere e-mail usando domini e indirizzi e-mail personali. Questo servizio consente di impostare nomi utente e password. Se configuri questi parametri, il back-end e-mail predefinito di Django è in grado di autenticarsi con Amazon SES e inviare correttamente le e-mail.

Nello sviluppo dell'applicazione Django-SES è stato utilizzato per l'invio delle e-mail per la conferma di registrazione sul sito da parte di un utente e per il recupero della password.

3.9 JavaScript

JavaScript [14] ideato nel 1995 da Netscape è un linguaggio di programmazione multi paradigma orientato agli eventi, interpretato o compilato just-in-time con funzioni di prima classe. Comunemente utilizzato nella programmazione web lato client (esteso poi anche al lato server) per la creazione, in siti web e applicazioni web, di effetti dinamici interattivi tramite funzioni di script invocate da eventi innescati a loro volta in vari modi dall'utente sulla pagina web in uso (tastiera, mouse, caricamento della pagina ecc...). Esistono molte librerie in commercio, molte delle quali open source, che integrano e semplificano l'utilizzo di questo linguaggio.

Nell'applicazione web creata vengono utilizzati JavaScript e alcune librerie solamente lato client, vengono quindi eseguiti direttamente sul browser dell'utente. Il linguaggio integrato nelle pagine HTML rende le pagine dinamiche andando a gestire gli eventi generati dagli utenti.

3.10 jQuery

jQuery [15] è una libreria JavaScript sviluppata da John Resig a partire dal 2006 con il preciso intento di rendere il codice più sintetico e di limitare al minimo l'estensione degli oggetti globali per ottenere la massima compatibilità con altre librerie. Da questo principio è nata una libreria veloce, piccola e in grado di offrire un'ampia gamma di funzionalità, che vanno dalla manipolazione degli stili CSS e degli elementi HTML, agli effetti grafici per passare a comodi metodi per chiamate AJAX cross-browser. Il tutto, appunto, senza toccare nessuno degli oggetti nativi JavaScript. In jQuery tutto ruota attorno all'oggetto/funzione \$, a sua volta un'abbreviazione (o alias) di jQuery.

Due caratteristiche fondamentali di jQuery sono anzitutto la brevità del codice utilizzato, ma soprattutto il fatto che l'elemento da ricercare sia passato alla funzione `$()` sotto forma di selettore CSS. Uno dei punti di forza della libreria è proprio il potente e performante motore di selezione. Un'altra caratteristica che va sottolineata è che molti metodi di jQuery sono concatenabili, per rendere la scrittura e la lettura del codice molto più lineare.

Oltre ai vantaggi già citati, vi sono alcuni buoni motivi per cui è una buona idea usare jQuery: anzitutto perché è possibile usarla in tutti i progetti senza paura di incappare in incompatibilità nel codice. In secondo luogo, jQuery ha un semplice sistema di estensione che permette di aggiungere nuove funzionalità (plugin) oltre a quelle predefinite. Inoltre, la sua diffusione ha fatto in modo che attorno al team di sviluppo ufficiale crescesse una numerosa community che mette a disposizione plugin e supporto di ottimo livello. Infine, il motto di jQuery è "Write less, do more", ed effettivamente la sua sintassi sintetica ed efficiente è particolarmente apprezzabile quando si tratta di scrivere svariate linee di codice.

Nella piattaforma la libreria jQuery è stata utilizzata proprio per la sua comodità e semplicità nella scrittura del codice JavaScript.

3.11 D3.js

D3.js [16] è una libreria JavaScript per la manipolazione di documenti in base ai dati. D3 aiuta a dare vita ai dati utilizzando HTML, SVG e CSS. L'enfasi di D3 sugli standard web offre tutte le funzionalità dei browser moderni senza essere vincolati a un framework proprietario, combinando potenti componenti di visualizzazione e un approccio basato sui dati alla manipolazione DOM.

D3 consente di associare dati arbitrari a un Document Object Model (DOM) e quindi di applicare trasformazioni basate sui dati al documento. Ad esempio, si può utilizzare D3 per generare una tabella HTML da una matrice di numeri. Oppure utilizza gli stessi dati per creare un grafico a barre SVG interattivo con transizioni e interazioni fluide. D3 non è un framework monolitico che cerca di fornire ogni caratteristica concepibile. Invece, risolve il nocciolo del problema: la manipolazione efficiente dei documenti sulla base dei dati. Ciò evita la rappresentazione proprietaria e offre una flessibilità straordinaria, esponendo tutte le funzionalità degli standard Web come HTML, SVG e CSS. Con un sovraccarico minimo, D3 è estremamente veloce e supporta set di dati di grandi dimensioni e comportamenti dinamici per l'interazione e l'animazione. Lo stile funzionale di D3 consente il riutilizzo del codice attraverso una raccolta diversificata di moduli ufficiali e sviluppati dalla comunità.

D3 utilizza un approccio dichiarativo, operando su insiemi arbitrari di nodi chiamati selezioni. I selettori sono definiti dall'API W3C Selectors e supportati in modo nativo dai browser moderni. Inoltre, fornisce numerosi metodi per la mutazione dei nodi: impostazione di attributi o stili; registrazione di ascoltatori di eventi; aggiungere, rimuovere o ordinare nodi; e modificare il contenuto HTML o di testo. Questi sono sufficienti per la stragrande maggioranza dei bisogni. È anche possibile l'accesso diretto al DOM sottostante, poiché ogni selezione D3 è semplicemente un array di nodi. Infine, D3 non introduce una nuova rappresentazione visiva. Il vocabolario dei segni grafici deriva direttamente dagli standard web: HTML, SVG e CSS.

In PixelCodingStudio la libreria D3.js è utilizzata principalmente per la creazione, gestione e colorazione della griglia e la creazione del codice-colore relativo.

3.12 HTML

HTML [17] è un acronimo che significa letteralmente "Hyper Text Markup Language", ovvero "Linguaggio a marcatori per ipertesti". Nato a fine anni '80, è quello che permette ancora oggi di costruire pagine web. È un linguaggio che permette di impaginare e formattare pagine collegate fra di loro attraverso link, ovvero, i siti web. Iper testo è un termine ormai datato (ma ancora valido) per definire l'albero di navigazione che collega le pagine web: un flusso infinito di pagine collegate fra di loro attraverso dei link che permettono di spostarsi da un contenuto all'altro.

Alla base dell'HTML ci sono i cosiddetti tag, che corrispondono a determinati tipi di contenuto. Ogni tag può avere degli attributi specifici, cosa che permette di costruire pagine diverse fra di loro e che rispondano alle necessità di chi le scrive.

Le pagine in HTML vanno pensate come la struttura di un albero: via via che si ramifica, si possono trovare più o meno elementi che costruiscono la pagina stessa seguendo una gerarchia ben precisa. In combinazione con altri linguaggi, come CSS, è possibile scrivere pagine semplici o complesse, interattive, che rispondono in maniera attiva alle interazioni dell'utente e che si adattano alla dimensione dello schermo del dispositivo dal quale le stiamo guardando.

3.13 CSS

L'acronimo CSS [18] sta per Cascading Style Sheets (fogli di stile a cascata) e designa un linguaggio di stile per i documenti web. I CSS istruiscono un browser o un altro programma utente su come il documento debba essere presentato all'utente, per esempio definendone i font, i colori, le immagini di sfondo, il layout, il posizionamento delle colonne o di altri elementi sulla pagina, etc.

La storia dei CSS procede su binari paralleli rispetto a quelli di HTML, di cui vuole essere l'ideale complemento. Da sempre infatti, nelle intenzioni degli uomini del W3C, HTML dovrebbe essere visto semplicemente come un linguaggio strutturale, alieno da qualunque scopo attinente la presentazione di un documento. Per questo obiettivo, ovvero arricchire l'aspetto visuale e la presentazione di una pagina, lo strumento designato sono appunto i CSS. L'ideale perseguito da anni si può sintetizzare con una nota espressione: separare il contenuto dalla presentazione. È grazie a CSS ad esempio se i siti web possono adattarsi in maniera dinamica passando da un computer allo schermo dello smartphone.

3.14 Bootstrap

Bootstrap [19] è uno dei framework CSS più utilizzati ed imitati, nato nel nido di Twitter ed oggi indipendente. Bootstrap può a buon diritto considerarsi lo standard de facto dei framework CSS per lo sviluppo di interfacce Web, trovando applicazione in ambiti come la creazione di template HTML preconfezionati ed i temi per i principali CMS, soprattutto in un'ottica responsiva.

Esso non è altro che un tool per gestire al meglio la fase di avvio di un progetto, un modo per poter contare su una serie di componenti riusabili e personalizzabili, comunque adattabili in termini stilistici ed estetici alle richieste del progetto e alla creatività di chi concepisce il sito.

Nella piattaforma è utilizzato solo per la rappresentazione di alcuni componenti, come le card per la visualizzazione dei progetti.

3.15 MySQL

MySQL [20] è un RDBMS (Database Management System relazionale) open source e libero, e rappresenta una delle tecnologie più note e diffuse nel mondo dell'IT. MySQL nacque nel 1996 per opera dell'azienda svedese Tcx, basato su un DBMS relazionale preesistente, chiamato mSQL. Il progetto venne distribuito in modalità open source per favorirne la crescita. Dal 1996 ad oggi, MySQL si è affermato molto velocemente prestando le sue capacità a moltissimi software e siti Internet. I motivi di tale successo risiedono nella sua capacità di mantenere fede agli impegni presi sin dall'inizio: alta efficienza nonostante le moli di dati affidate; integrazione di tutte le funzionalità che offrono i migliori DBMS: indici, trigger e stored procedure ne sono un esempio; altissima capacità di integrazione con i principali linguaggi di programmazione, ambienti di sviluppo e suite di programmi da ufficio. Quotidianamente, immense quantità di informazioni vengono affidate a tecnologie che ne garantiscono la conservazione duratura ed un recupero efficiente che ne permetta l'analisi. Da anni, questo ruolo viene interpretato molto bene da un prodotto software completo, efficiente ed affidabile: MySQL.

I concetti centrali in tema di memorizzazione di dati sono due: database e DMBS.

Il primo indica un sistema di file finalizzato a memorizzare informazioni a supporto di un qualsivoglia software. La struttura interna di un database deve rispettare una certa architettura di immagazzinamento dei dati per poterne permettere il corretto salvataggio, il rispetto dei tipi impiegati e soprattutto agevolarne il recupero, un'operazione generalmente molto onerosa.

Un **DBMS** è un servizio software, realizzato in genere come server in esecuzione continua, che gestisce uno o più database. I programmi che dovranno interagire quindi con una base di dati non potranno farlo direttamente, ma dovranno dialogare con il DBMS. Esso sarà l'unico ad accedere fisicamente alle informazioni. Quanto detto implica che il DBMS è il componente che si occupa di tutte le politiche di accesso, gestione, sicurezza ed ottimizzazione dei database.

I DBMS esistenti non sono tutti della stessa tipologia. Al giorno d'oggi, ad esempio, si parla molto di DBMS NoSQL, nati per venire incontro alle esigenze dei più recenti servizi Web. MySQL invece, come detto in precedenza è un RDBMS (Relational DBMS), ispirato dalla Teoria Relazionale. Questa nacque nel 1970 ad opera del britannico Edgar f. Codd. Nonostante i 40 anni abbondanti trascorsi, i suoi principi si dimostrano tuttora attuali.

Un **database relazionale** è costituito da tabelle, ognuna delle quali è composta

da righe e colonne. Ogni riga della tabella è un record identificato da un codice univoco denominato chiave. Le colonne, invece, contengono gli attributi dei dati e ogni record di solito ha un valore per ogni attributo, rendendo facile stabilire le relazioni. Le tabelle che compongono il database non sono del tutto indipendenti tra loro ma relazionate da legami logici e ne esistono di vari tipi.

Strumento fondamentale nell'interazione con i DBMS relazionali, è il linguaggio **SQL** (Structured Query Language). Si tratta di un formalismo che permette di indicare al DBMS quali operazioni svolgere sui database che gestisce. Tramite SQL si può attivare qualsiasi tipo di operazione, sia sui dati che sulla struttura interna del database, sebbene le principali e più frequenti operazioni ricadono in una delle seguenti quattro tipologie: inserimento, lettura, modifica e cancellazione di dati, tipicamente indicate con l'acronimo CRUD (Create-Read-Update-Delete).

In questa applicazione web il linguaggio SQL non è utilizzato direttamente poiché Django fornisce automaticamente un'API di estrazione del database che consente di creare, recuperare, aggiornare ed eliminare oggetti.

Come detto in precedenza, in alternativa ai database relazionali vi sono i database non relazionali, i cosiddetti **database NoSQL**. A differenza dei database relazionali, i database non relazionali, non memorizzano i dati in tabelle e record, ma la struttura di archiviazione dei dati è progettata e ottimizzata per esigenze specifiche. Invece di SQL, i database NoSQL si appoggiano a vari linguaggi a seconda del problema che si deve andare a risolvere. I quattro tipi più diffusi di database NoSQL sono: orientati alla colonna, orientati al documento, coppie chiave-valore e database a grafo. È possibile utilizzare questi tipi singolarmente o combinarli. La scelta dipende dall'applicazione e dal tipo di dati da archiviare.

Uno dei vantaggi è che i database NoSQL sono generalmente scalabili e altamente disponibili. Inoltre, hanno modelli di dati flessibili, infatti, la maggior parte dei sistemi non relazionali non richiede agli sviluppatori di assumere impegni iniziali nei confronti dei modelli di dati, gli schemi possono spesso essere cambiati al volo. Infine, forniscono prestazioni di alto livello.

La scelta di impiegare MySQL nella piattaforma e quindi di scegliere un database relazionale è dettata dalla semplicità di configurazione e messa in funzione del database derivanti da anni di sviluppo. È ricco di documentazione sia online che cartacea e sono disponibili numerose applicazioni di terze parti. Inoltre, è una soluzione sicura e la struttura dei dati da memorizzare si sposa bene con la struttura relazionale.

3.16 PhpMyAdmin

PhpMyAdmin [21] è uno strumento software gratuito scritto in PHP, destinato a gestire l'amministrazione di MySQL sul Web. È un'interfaccia grafica con cui è possibile visualizzare il contenuto del proprio database; creare, modificare e cancellare intere tabelle o singoli record; fare un backup dei dati contenuti e visualizzare informazioni interessanti sul database.

Nella piattaforma fornisce un supporto ulteriore per la gestione del database.

3.17 Clustering

Con il termine “Clustering” (che letteralmente significa “addensamento” o “raggruppamento”) si denota una famiglia di tecniche non supervisionate, in grado di individuare raggruppamenti intrinseci (detti cluster) tra i dati nello spazio multidimensionale, e di etichettare le classi in corrispondenza di tali raggruppamenti.

Un cluster è formato da un insieme di oggetti che sono “simili” tra loro, e sono “dissimili” da tutti gli oggetti appartenenti ad altri cluster.

L'analisi dei cluster è un'attività umana importante. Persino nell'infanzia uno impara a distinguere tra gatti e cani o tra animali e piante migliorando continuamente i suoi schemi di classificazione subconsci. L'analisi dei cluster è stata largamente utilizzata in numerose applicazioni, comprese il riconoscimento dei pattern, l'analisi dei dati, il trattamento delle immagini e la ricerca di mercato. Uno dei vantaggi di tale processo riguarda il fatto che esso è adattabile ai cambiamenti e consente di scegliere quali sono le caratteristiche di interesse per distinguere i vari gruppi.

Nella maggior parte dei casi, i criteri di clustering vengono definiti sulla base delle due seguenti osservazioni:

- Coppie di oggetti presi da uno stesso cluster devono sempre essere più simili, rispetto a coppie prelevate da cluster differenti.
- I cluster sono costituiti per loro natura da “nuvole” di punti a densità relativamente elevata, separati da zone con densità inferiore.

In letteratura esistono diverse tecniche di clustering, che vengono suddivise secondo criteri differenti. Una delle principali categorizzazioni degli algoritmi di clustering è basata sulla scelta del modello di suddivisione dello spazio dei cluster.

Clustering Gerarchico (Hierarchical Clustering): gli algoritmi di clustering

appartenenti a questa categoria organizzano i dati in sequenze innestate di gruppi, le quali vanno a formare una gerarchia di dati. Gli strumenti grafici più utilizzati per la rappresentazione di questa tecnica sono alberi e dendrogrammi.

Le fasi intermedie del processo di clustering gerarchico, ovvero le fasi che portano dallo stato iniziale (dati grezzi, non raggruppati) allo stato finale (dati suddivisi in cluster), possono essere caratterizzate da due “filosofie di raggruppamento” diverse:

- Metodo aggregativo (o Bottom-Up): questo metodo prevede che inizialmente tutti gli elementi siano considerati cluster indipendenti, e che procedendo nelle fasi dell’algoritmo, i cluster più vicini si uniscano. Gli algoritmi Bottom-Up procedono fino al raggiungimento del numero di cluster prefissato, o fino a che la “distanza minima” tra i cluster non supera un determinato valore di soglia.
- Metodo divisivo (Top-Down): L’algoritmo prevede che allo stato iniziale ogni elemento appartenga ad un unico cluster, e procedendo nell’esecuzione dell’algoritmo, i cluster vengano divisi in gruppi di elementi sempre più piccoli e più omogenei. Gli algoritmi Top-Down necessitano di conoscere un numero di cluster prefissato, per poterlo utilizzare come condizione di arresto.

Clustering Partizionale (Partitional Clustering): i metodi appartenenti a questa categoria cercano di suddividere gli elementi del dataset in un insieme di cluster disgiunti. Il partizionamento dei dati avviene per mezzo di algoritmi euristici iterativi, che cercano di minimizzare i criteri di dissimilarità dei dati.

Un’altra categorizzazione importante delle tecniche di clustering dipende dalla possibilità che un elemento possa essere o meno assegnato a cluster differenti:

- Clustering Deterministico (o esclusivo): si parla di clustering esclusivo o Hard Clustering, quando ogni elemento può appartenere a uno e un solo cluster. (Con questa soluzione i cluster non possono avere elementi in comune).
- Clustering Non-deterministico (o non-esclusivo): si parla di clustering non-esclusivo o Soft Clustering, quando ogni elemento può appartenere a più cluster differenti, con gradi di appartenenza diversi.

La risoluzione dei problemi di clustering è affidata a numerosi modelli e algoritmi, che vengono scelti ed utilizzati in base alla tipologia di dati posseduti. Di seguito andremo ad approfondire tre diversi algoritmi analizzati per l’implementazione della pixelizzazione delle immagini nella piattaforma.

3.17.1 K-Means

Il K-Means è un algoritmo di clustering di tipo partizionale esclusivo, center-based, nel quale ogni oggetto della collezione dei dati è assegnato al cluster con il centroide

più vicino. Lo scopo dell'algoritmo è quello di trovare la migliore divisione di n entità in k gruppi (Figura 3.3), tali che la distanza tra i membri del gruppo e il centroide, rappresentante del gruppo, corrispondente sia minimizzata.

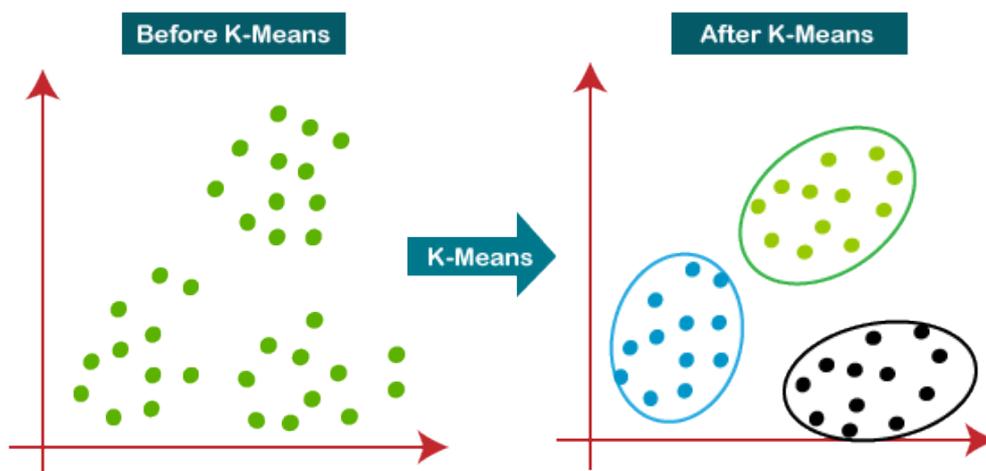


Figura 3.3: K-Means

L'algoritmo K-means riceve in input un parametro k e partiziona un insieme di n oggetti in k cluster in modo tale che la similarità intra-cluster risultante sia alta mentre la similarità inter-cluster sia bassa. La similarità dei cluster è misurata rispetto al valore medio degli oggetti in un cluster; tale valore può essere visto come il centro di gravità del cluster. Deciso il numero di classi in cui si vuole suddividere il dataset stesso, l'algoritmo procede nel seguente modo:

- Seleziona randomicamente k centroidi appartenenti allo spazio delle features, ciascuno dei quali rappresenta inizialmente la media o il centro di un cluster. L'unica condizione è che non siano coincidenti, anzi solitamente ci si assicura che siano abbastanza distanti tra loro.
- Si calcola la distanza di ogni punto del dataset rispetto ad ogni centroide.
- Ciascuno degli oggetti rimanenti viene associato al cluster più simile basandosi sulla distanza tra l'oggetto e la media del cluster.
- Esso, quindi, calcola la nuova media per ciascun cluster, ovvero, viene ricalcolata la posizione di ogni centroide.
- Tale processo viene iterato fino a quando non si raggiunge la convergenza di una determinata funzione criterio. Tipicamente viene utilizzato come criterio l'errore quadratico.

L'algoritmo quindi tenta di determinare k partizioni che minimizzano la funzione errore quadratico. Esso si comporta bene quando i cluster sono compatti e piuttosto ben separati uno dall'altro. È relativamente scalabile ed efficiente nel processare grandi insiemi di dati perché la complessità computazionale dell'algoritmo è $O(nkt)$, dove n è il numero totale di oggetti, k è il numero di cluster e t è il numero di iterazioni. Normalmente, $k \ll n$ e $t \ll n$. Il metodo spesso termina in un ottimo locale. Tuttavia, può essere applicato soltanto quando è definita la media di un cluster. Questo può non accadere in molte applicazioni, ad esempio quando sono coinvolti dati con attributi categorici. La necessità per gli utenti di specificare all'inizio k , ovvero il numero dei cluster, può essere vista come uno svantaggio. K-means non è adatto per scoprire cluster con forme non convesse o cluster di dimensioni molto differenti. Inoltre, è sensibile al rumore e agli outlier dal momento che un piccolo numero di questi dati possono influenzare sostanzialmente il valore medio.

3.17.2 K-Medoids

Il K-Medoids è un algoritmo di clustering di tipo partizionale esclusivo, center-based, correlato al K-Means, ma se nel K-means il punto designato per essere al centro è il baricentro, quindi può essere un punto artificiale, nel K-medoids è usato il punto tra quelli dati, collocato più centralmente (Figura 3.4).

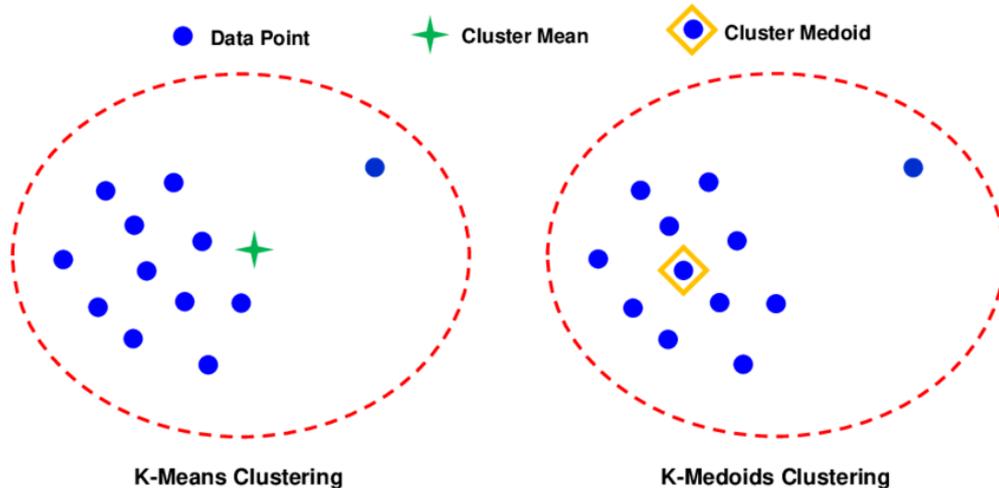


Figura 3.4: K-Medoids

Questo algoritmo nasce poiché l'algoritmo K-means è sensibile agli outlier, dal momento che un oggetto con un valore estremamente grande può distorcere sensibilmente la distribuzione dei dati. Per diminuire la sensibilità, invece di prendere

come punto di riferimento il valore medio degli oggetti in un cluster, può essere utilizzato il medoid, ovvero l'oggetto localizzato più centralmente in un cluster. In questo modo è possibile comunque eseguire il metodo di partizionamento cercando di minimizzare la somma delle dissimilarità tra ciascun oggetto e il suo punto di riferimento corrispondente. In altre parole, viene utilizzato il criterio di errore assoluto.

La strategia di base dell'algoritmo K-medoids è quella di raggruppare n oggetti in k cluster, deciso il numero di classi in cui si vuole suddividere il dataset stesso, l'algoritmo procede nel seguente modo:

- Esso trova arbitrariamente un oggetto rappresentativo per ciascun cluster.
- Ciascun oggetto rimanente viene inserito nel cluster associato al medoid più simile.
- La strategia sostituisce iterativamente medoidi con non-medoidi.
- L'algoritmo termina quando la qualità del clustering risultante non può più essere migliorata. Questa qualità viene stimata utilizzando una funzione di costo che misura la dissimilarità media tra un oggetto e il medoide del suo cluster.

In un algoritmo di tipo K-medoid il costo di una singola iterazione è $O(k(n - k)^2)$. Per grandi valori di n e k tale calcolo diventa molto costoso. Tale metodo è più robusto di K-means in presenza di rumore e di outlier perché un medoide è meno influenzato dagli outlier o da altri valori estremi rispetto ad una media. Tuttavia, l'elaborazione di K-medoids è più costosa rispetto al metodo K-means.

3.17.3 DBSCAN

Per individuare cluster di forma arbitraria sono stati sviluppati i metodi di clustering basati sulla densità. Questi tipicamente considerano i cluster come regioni dense di oggetti nello spazio dei dati separate da regioni a bassa densità (che rappresentano rumore).

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) è un algoritmo di clustering basato sulla densità. L'algoritmo costruisce ciascun cluster mettendo insieme regioni con densità sufficientemente alta (Figura 3.5). Esso è in grado di individuare cluster di forma arbitraria e fornisce risultati interessanti anche in presenza di rumore. In DBSCAN un cluster viene definito come un insieme massimale di punti density-connected.

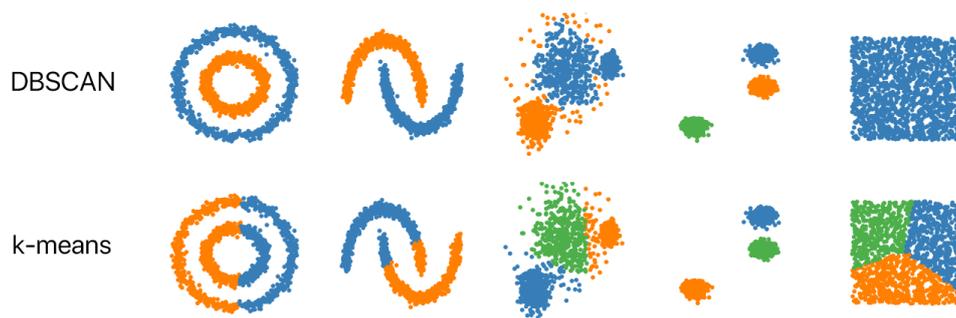


Figura 3.5: DBSCAN

DBSCAN stima la densità attorno a ciascun punto (item) contando il numero di punti in un intorno ε (o *eps*) specificato dall'utente, ed applica delle soglie chiamate *minPts* anch'esse specificate dall'utente per identificare i punti "core", "border" e "noise". Esso costruisce i cluster determinando l' ε -neighborhood (vicinato di raggio ε di un determinato oggetto) di ciascun punto nel database. I passi successivi dell'algoritmo sono:

- Se l' ε -neighborhood di un punto p contiene più di $MinPts$, viene creato un nuovo cluster con p come core object.
- Vengono individuati gli oggetti density-reachable ("raggiungibili per densità", cioè se esiste una catena di punti core in cui ogni punto ricade all'interno dell' ε -intorno del successivo) dai vari core object.
- se due core object sono direttamente density-reachable i cluster corrispondenti vengono fusi.
- Il processo termina quando nessun nuovo punto può essere aggiunto ad un cluster.

Se un oggetto non è contenuto in nessun cluster viene considerato rumore. Se viene utilizzato un indice spaziale la complessità computazionale di DBSCAN è $O(n \log(n))$, dove n è il numero di oggetti del database; in caso contrario tale complessità diventa $O(n^2)$. Con una definizione appropriata dei parametri ε e *minPts* definiti dall'utente, l'algoritmo è efficace nel trovare cluster di forma arbitraria.

Capitolo 4

Implementazioni

Per l'implementazione di PixelCodingStudio dopo aver deciso le funzionalità di cui l'applicazione web deve disporre, il primo passo è stato quello di definire la struttura dei dati da memorizzare, necessario per una corretta successiva implementazione di quest'ultime. Come detto in precedenza si è scelto un database relazionale poiché si sposa bene con le necessità richieste per lo sviluppo della piattaforma. Successivamente sono state implementate le funzionalità seguendo la logica Model-Template-View di Django. Ogni funzionalità quindi è caratterizzata da una parte di front-end con cui viene gestita l'interazione dell'utente con la pagina ed una parte di back-end con cui vengono elaborate, gestite ed eseguite le richieste dell'utente.

La prima funzionalità ad essere stata implementata è la gestione dell'utente, permettendone la registrazione, il login, il logout e la successiva visualizzazione e modifica del profilo con il recupero della password in caso di smarrimento. Una volta permesso all'utente di far parte della piattaforma memorizzando le proprie credenziali si è passati all'implementazione della funzionalità di creazione e memorizzazione dei progetti attraverso la Pixel-art. Data la possibilità all'utente di creare progetti si è proseguito con la costruzione e visualizzazione del codice-colore relativo ad ogni progetto. Terminato lo sviluppo di queste funzionalità non restava altro che permettere la visualizzazione e la modifica dei progetti salvati. Successivamente si è passati alla gestione della community dove ogni utente è in grado di visualizzare non solo i propri progetti ma anche quelli altrui. Infine si è deciso di implementare un'ulteriore funzionalità, permettendo all'utente di caricare un'immagine, processarla e visualizzarla sulla griglia. Per questa funzionalità sono stati eseguiti dei test per l'identificazione dell'algoritmo di clustering più adatto.

4.1 Database

Nelle prime fasi di sviluppo di un sistema software, dopo la raccolta di requisiti e funzionalità e prima di decidere quali tecnologie utilizzare, è necessario costruire un modello dei dati o data model. Un data model è un modello astratto utilizzato per descrivere tutti gli aspetti dei dati che il sistema si ritroverà a dover maneggiare, dalla struttura alle loro relazioni reciproche. Definire correttamente il data model è fondamentale per evitare alcuni problemi che non è facile prevedere nei primi stadi di sviluppo di un sistema informativo ma che emergono più tardi, quando è molto più complicato gestirli. Se non vengono adottati dei protocolli o standard nel design di un data model, è possibile che il sistema si ritrovi a gestire dati con formati e definizioni particolari, che non sono facilmente condivisibili tra i vari componenti dell'applicazione. Altro problema generato da un data model mal compilato è il rischio di ritrovarsi a gestire entità e funzioni duplicate o molto simili tra di loro, che di conseguenza causerebbero ridondanza.

Il database come detto nel capitolo precedente viene gestito tramite MySQL e la sua struttura è di tipo relazionale, per questo per la rappresentazione del data model si è utilizzato prima un modello ER e di conseguenza un modello logico.

4.1.1 Modello ER

Un diagramma entità relazione (ER) è un tipo di diagramma di flusso che illustra come le "entità", quali persone, oggetti o concetti, si relazionano tra loro all'interno di un sistema. I modelli ER sono spesso utilizzati per progettare o eseguire il debug di database relazionali nei settori dell'ingegneria del software, dei sistemi informativi aziendali, dell'istruzione e della ricerca. Essi utilizzano un set definito di simboli, come rettangoli, diamanti, ovali e linee di collegamento, per rappresentare l'interconnessione tra entità, relazioni e i loro attributi. Rispecchiano la struttura grammaticale, con le entità come sostantivi e le relazioni come verbi.

Più dettagliatamente un modello Entità-Relazione (ER) è un modello concettuale di dati, e come tale fornisce una serie di strutture (costrutti), atte a descrivere la realtà in una maniera facile da comprendere e che prescinde dai criteri di organizzazione dei dati nei calcolatori. Il modello ER usa simboli grafici per favorire l'immediatezza della comprensione. Gli schemi ER sono infatti schemi essenzialmente grafici con aggiunte di frasi di specifica e di vincolo.

- L'**entità** rappresenta una classe di oggetti del mondo reale (oggetti sia materiali che immateriali). Ogni entità è caratterizzata da un nome e viene rappresentata con un rettangolo.

- **Associazione o relazione** rappresenta un legame logico tra entità. Ogni associazione è caratterizzata da un nome ed è rappresentata da un rombo.
- Le **proprietà o attributi** rappresentano caratteristiche delle entità e delle associazioni. Ogni istanza di entità e di associazione possiede un valore per ciascuna proprietà e ogni proprietà è caratterizzata da un nome.
- Il **simbolo (n,m)** esprime la cardinalità della proprietà dove n e m rappresentano il numero minimo e massimo di valori dell'attributo associati ad ogni occorrenza di entità o relazione.

In Figura 4.1 vi è il modello ER utilizzato per la rappresentazione della struttura dei dati in PixelCodingStudio.

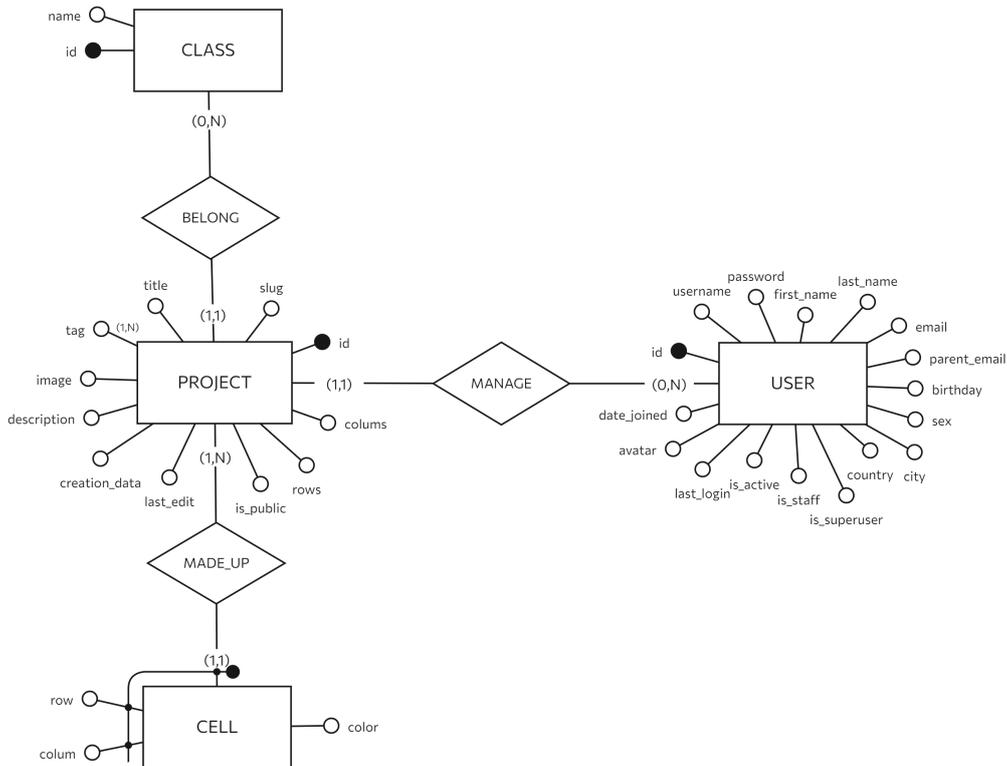


Figura 4.1: Modello ER

Come si può notare vi sono quattro entità principali: user, project, class e cell. Ogni entità è caratterizzata da diversi attributi e queste sono legate tra loro da delle relazioni la cui cardinalità è espressa dal simbolo (n,m). Le relazioni sono le seguenti:

- Tra user e project vi è una relazione uno a molti poiché l'utente registrato può gestire zero o infiniti progetti. Zero perché l'utente può essere presente all'interno della piattaforma anche senza creare progetti. Un progetto creato, invece, appartiene ad uno ed uno solo utente.
- Tra project e class vi è una relazione uno a molti poiché al progetto è assegnata una ed una solo categoria, invece, ad una categoria possono appartenere zero o infiniti progetti.
- Tra project e cell vi è una relazione uno a molti poiché un progetto è costituito da più celle, invece, una cella appartiene ad uno ed uno solo progetto.

4.1.2 Modello logico

La traduzione del modello ER in un modello logico ci permette di comprendere quelle che sono le tabelle create nel database con i relativi attributi.

La classe **User** (Tabella 4.1) rappresenta gli utenti presenti nell'applicazione. Ogni volta che un utente effettua la registrazione vengono memorizzate le credenziali inserite. La chiave, ovvero un valore univoco che identifica ogni record è l'id e viene generato automaticamente con valori numerici progressivi. Successivamente l'utente deve inserire un username, il nome, cognome, il sesso, la data di compleanno, un'email ed una password. Tutti questi attributi sono obbligatori per la registrazione. Inoltre, se l'utente risulta minorenne, deve inserire anche l'email del proprio genitore. Altri dati che l'utente può memorizzare all'interno del database sono la città, la nazione ed un avatar, ovvero il percorso di un'immagine da mettere come profilo. Questi ultimi sono dati facoltativi. Infine, vi sono ulteriori dati che vengono inseriti automaticamente senza poter essere modificati dall'utente. Questi sono data_joined che corrisponde alla data di registrazione, last_login che rappresenta l'ultimo accesso effettuato all'interno dell'applicazione, is_active è un valore che indica se l'utente è considerato attivo oppure no ed infine is_staff che indica se l'utente ha i permessi di amministratore oppure no. Nell'implementazione vera è proprio però Django fornisce una classe Utente predefinita dove sono presenti solo alcuni degli attributi citati. Per l'aggiunta dei restanti attributi si è creata un'altra classe denominata Profile. Tra queste due classi vi è una relazione uno ad uno poiché all'utente corrisponde uno e un solo profilo ed ad un profilo corrisponde uno ed un solo utente.

USER	
ATTRIBUTO	TIPO VARIABILE
id	INT
username	VARCHAR(150)
email	VARCHAR(254)
password	VARCHAR(128)
date_joined	DATETIME(6)
first_name	VARCHAR(150)
last_login	DATETIME(6)
last_name	VARCHAR(150)
is_active	TINYINT(1)
is_staff	TINYINT(1)
birthday	DATE(6)
sex	VARCHAR(150)
city	VARCHAR(250)
country	VARCHAR(250)
avatar	VARCHAR(100))
email_genitore	VARCHAR(250)

Tabella 4.1: User

La classe **Project** (Tabella 4.2) rappresenta i progetti creati dagli utenti sulla piattaforma. Ogni progetto è caratterizzato da un id: un codice univoco generato automaticamente con valori numerici progressivi. Oltre all'id vi è lo slug, un ulteriore valore univoco generato automaticamente dalla combinazione dell'id con il titolo. In questa classe vi sono due chiavi esterne: lo `user_id` che fa riferimento all'id dell'utente a cui è associato il progetto e `category_id` che fa riferimento all'id della categoria a cui è appartiene il progetto. Nella creazione di un progetto i dati obbligatori da memorizzare sono `title` ovvero il titolo che l'utente dà al proprio progetto, `rows` e `columns` ovvero il numero di righe e colonne presenti nella griglia creata dall'utente, `image` ovvero il percorso associato all'immagine della griglia colorata generata dall'utente e `is_public` che indica se il progetto è pubblico,

quindi visibile all'interno della community oppure privato, quindi visibile solo dall'autore del progetto. `description` invece, è un attributo facoltativo che contiene la descrizione del progetto. Infine, vi sono `creation_data` che indica la data di creazione e `last_edit` che indica la data di ultima modifica, due attributi assegnati automaticamente.

PROJECT	
ATTRIBUTO	TIPO VARIABILE
id	INT
slug	VARCHAR(250)
user_id	INT
title	VARCHAR(250)
description	LONGTEXT
creation_data	DATETIME(6)
last_edit	DATETIME(6)
is_public	TINYINT(1)
rows	INT
columns	INT
image	VARCHAR(100)
category_id	INT

Tabella 4.2: Project

La classe **Category** (Tabella 4.3) rappresenta le categorie di appartenenza di un progetto. È una classe molto semplice composta da un attributo univoco generato automaticamente con valori numerici progressivi che è l'id e dal nome ovvero il nome della categoria. All'interno del database è presente una lista predefinita in fase di sviluppo di categorie. Queste non possono essere aggiunte o modificate dall'utente ma solo dall'amministratore del sito. I nomi delle categorie che è possibile associare ad un progetto sono: animali, cibo, famiglia, fantasia, feste, giochi, natura, oggetti, paesaggi, personaggi, sport.

CATEGORY	
ATTRIBUTO	TIPO VARIABILE
id	INT
name	VARCHAR(250)

Tabella 4.3: Category

La classe **Tag** (Tabella 4.4) rappresenta i tag associati ad un progetto. Un utente può decidere di inserire uno o più tag per lo stesso progetto. Anche questa è una classe molto semplice composta dall'id: un attributo univoco assegnato automaticamente con valori numerici progressivi. Il name ovvero il nome del tag inserito dall'utente ed infine project_id che rappresenta l'id identificativo del progetto associato.

TAG	
ATTRIBUTO	TIPO VARIABILE
id	INT
name	VARCHAR(250)
project_id	INT

Tabella 4.4: Tag

La classe **Cell** presente nel modello ER rappresenta le celle che compongono la griglia di un progetto. In realtà questa rappresentazione è utile solo per il modello relazionale del database. Nell'implementazione non viene creata una tabella poiché sarebbe una soluzione troppo onerosa, che andrebbe ad occupare un'elevata porzione di memoria. Il numero di record che si andrebbero a generare per un singolo progetto e poi per tutti quelli salvati sarebbe spropositato. Per questo si è adottata una soluzione che andasse ad occupare molta meno memoria, andando ad inserire all'interno della classe Project un ulteriore attributo chiamato oggetto in formato json, contenente tutte le informazioni della griglia creata. Le informazioni utili sono le coordinate di ogni cella sotto forma di riga e colonna ed il colore associato.

4.2 Funzionalità

PixelCodingStudio è composto da diverse pagine web che l'utente è in grado di navigare. Ognuna di queste ha una funzionalità diversa, fondamentale per il funzionamento dell'applicazione e la maggior parte di esse sono collegate fra loro attraverso un menù di navigazione. Gli utenti però non sono in grado di accedere a tutte le pagine presenti nel sito, ma vi sono alcuni permessi che cambiano in base a se l'utente ha effettuato il login oppure no. Per effettuare il login un utente deve essere necessariamente registrato. Le funzionalità le possiamo suddividere in tre macro aree: gestione dell'utente, gestione dei progetti, community.

4.2.1 Gestione utente

La gestione dell'utente fa riferimento alla possibilità che l'utente ha una volta atterrato sul sito di potersi registrare, effettuare l'accesso, visualizzare e modificare il proprio profilo, modificare o recuperare la password ed infine effettuare il logout. All'interno dell'architettura di Django la gestione lato server di queste funzionalità è contenuta nella sezione accounts mentre per la parte di front-end vi sono diverse pagine scritte in html, javascript e css contenute nelle sezioni templates e static.

La registrazione di un utente sull'applicazione è possibile accedendo alla pagina Registrati disponibile attraverso un link nel menù di navigazione presente nella Home. L'utente per registrarsi e quindi memorizzare i propri dati in modo permanente all'interno dell'applicazione deve inserire una serie di credenziali, alcune obbligatorie ed altre facoltative. Durante l'inserimento dei dati, attraverso una funzione javascript viene controllata l'età dell'utente. Se risultasse minorenni per poter procedere con il salvataggio dei dati egli dovrà inserire anche l'indirizzo email di un proprio genitore. Successivamente verranno inviati i dati al server che attraverso una funzione che riceve in ingresso la richiesta li processerà e li salverà all'interno del database. Questa funzione oltre a salvare i dati si occupa anche dell'invio di una email di conferma registrazione all'indirizzo email inserito dall'utente e se minorenni verrà inviata un'ulteriore email di avviso registrazione all'indirizzo email del genitore. L'invio di queste email di notifica è reso possibile grazie a Django-SES.

Una volta registrato, l'utente sarà automaticamente loggato all'interno del sito e verrà reindirizzato alla pagina di visualizzazione del proprio profilo dove potrà controllare la correttezza dei dati inseriti. A questa pagina è possibile accedere anche attraverso il menù di navigazione cliccando sul proprio username. All'interno della pagina sono inoltre presenti due link che permettono il primo di accedere alla pagina per la modifica del profilo, aggiornandolo o inserendo i dati mancanti, il secondo di accedere alla pagina per la modifica della password. Infine, è possibile

cancellare la propria iscrizione premendo il pulsante dedicato all'interno della pagina.

Un utente registrato, inoltre, può effettuare il login ed il logout attraverso le pagine Accedi ed Esci raggiungibili dal menù di navigazione. Se l'utente decidesse di effettuare il login dovrebbe inserire il proprio nome utente, la password e cliccare il bottone Login. La richiesta verrà inviata al server che attraverso una funzione preimpostata in Django modificherà lo stato dell'utente come loggato. In caso di smarrimento della password, nella pagina di login è presente un link per il recupero. Grazie a questo link si accede ad una pagina di recupero e digitando l'indirizzo email inserito in fase di registrazione l'utente riceverà un email con la nuova password. Per effettuare il logout, invece, basterà solamente cliccare il link Esci ed una funzione gestita direttamente da Django provvederà al logout dell'utente.

4.2.2 Gestione progetti

La gestione dei progetti è il core dell'applicazione. Comprende tutte quelle funzionalità che permettono all'utente di creare esercizi di coding unplugged attraverso la Pixel-art e quindi di approcciare il pensiero computazionale. L'utente sarà in grado di creare un progetto, salvarlo, modificarlo e scaricare il relativo codice-colore. All'interno dell'architettura di Django la gestione lato server di queste funzionalità è contenuta nella sezione public mentre per la parte di front-end vi sono diverse pagine scritte in html, javascript e css contenute nelle sezioni templates e static.

La creazione di un progetto è la parte più articolata dell'intera applicazione. L'utente per creare un progetto può partire da zero generando una griglia bianca, oppure può caricare un'immagine che verrà trasferita sulla griglia. Per accedere alla pagina Crea presente nel menù di navigazione che permette di usufruire di questa funzionalità l'utente dovrà essere necessariamente loggato.

Per creare la propria Pixel-art da zero l'utente dovrà in un primo momento scegliere il numero di righe e colonne per dimensionare la griglia. Cliccato il bottone Crea Griglia verrà generata e visualizzata una griglia bianca dove ogni quadrato corrisponde ad un pixel. L'utente potrà colorare ogni quadratino trascinando il mouse con il tasto sinistro cliccato e scegliere il colore più appropriato. Per selezionare il colore bisogna cliccare il bottone denominato Colore, a questo punto verrà visualizzata una palette contenente dei colori predefiniti. Visualizzata la palette l'utente dovrà cliccare sopra il colore che desidera per poterlo utilizzare. È possibile anche utilizzare colori non selezionabili all'interno della palette inserendo un codice hex valido nella casella di testo dedicata. Se si vuole modificare la griglia in fase di colorazione, vi sono dei bottini che se cliccati permettono di aggiungere o

eliminare una riga dal basso, aggiungere o eliminare una colonna da destra, pulire la griglia facendola ritornare completamente bianca, oppure eliminarla per crearne una successiva. La gestione di tutti questi eventi avviene lato client attraverso apposite funzioni scritte in javascript con l'ausilio delle librerie D3.js e jQuery.

Come detto in precedenza è possibile anche creare la propria Pixel-art caricando un'immagine. Questo processo è gestito da una funzione lato server. L'utente dovrà selezionare un'immagine da caricare ed una dimensione corrispondente al numero di quadratini in cui si vuole suddividere il lato più lungo dell'immagine. Il lato più corto verrà suddiviso di conseguenza per mantenere le proporzioni tra i lati e non distorcere l'immagine. Inviata l'immagine al server essa verrà segmentata in immagini di dimensione più piccole tante quanti saranno i quadratini della griglia da creare. Ad ogni immagine verrà applicato l'algoritmo di clustering K-means. Impostato il numero k di gruppi in cui suddividere la collezione di dati pari a 1 e il numero di ripetizione pari a 10, esso calcolerà il centroide. I dati della collezione non sono altro che i colori dei pixel che compongono l'immagine, invece, il centroide calcolato è il colore che rappresenta la collezione. Terminato il calcolo dei centroidi per ogni segmento dell'immagine, il server invierà questi dati al client ed attraverso una funzione javascript verrà creata una griglia in cui il colore di ogni quadratino corrisponde al centroide dell'immagine segmentata relativa alla stessa posizione.

Di default la griglia verrà colorata con i colori generati dall'algoritmo K-means, l'utente però spuntando una casella prima di inviare l'immagine al server può decidere di colorare la griglia usando solo i colori presenti all'interno della palette. Il processo descritto in precedenza non cambia ma vi è un ulteriore passaggio in cui i colori dei centroidi vengono normalizzati con i colori della palette, ovvero si cerca il colore che si avvicina di più.

Una volta finito il progetto l'utente può decidere di salvarlo nel database. Egli dovrà inserire un titolo, almeno un tag, scegliere una categoria tra quelle proposte e decidere se il progetto sarà pubblico (valore di default) e quindi visibile dalla community o privato e quindi visibile solo dall'autore levando il segno di spunta dall'apposita casella. Facoltativa invece è la descrizione del progetto. Cliccato il bottone Salva Progetto i dati inseriti verranno inviati al server insieme alle caratteristiche della griglia in formato json come la posizione di ogni cella con il relativo colore che permettono una seguente riproduzione di essa. Lato server grazie ad una funzione verranno processati i dati, verrà generata la data di creazione, la data di ultima modifica e verrà salvato il progetto all'interno del database.

Salvato il progetto, l'utente sarà reindirizzato ad una pagina dove è possibile visualizzare la Pixel-art creata ed i dati inseriti. Da questa pagina, inoltre, potrà

eliminare il progetto dal database inviando la richiesta al server o accedere alla pagina di modifica. Nella pagina di modifica egli potrà inserire eventuali dati mancanti come la descrizione del progetto o aggiornare i dati inseriti in precedenza ad eccezione del titolo. Inoltre, potrà modificare la griglia nelle dimensioni e nella colorazione ed infine salvare il tutto inviando i dati al server. Attraverso una funzione verranno aggiornati all'interno del database gli eventuali dati modificati e la data di ultima modifica.

Altra funzionalità molto importante è la creazione e visualizzazione del pdf contenente il codice-colore con cui gli utenti potranno effettuare gli esercizi di coding unplugged. È stato scelto un formato pdf rispetto ad una pagina web poiché è possibile scaricarlo in locale e stamparlo. Essendo una funzionalità principale si accede al pdf o dalla pagina di creazione del progetto o da quella di modifica o da quella in cui si visualizza il progetto salvato. Cliccando sul bottone Visualizza Codice verrà creato il codice-colore relativo alla griglia generata e colorata attraverso funzioni scritte in javascript lato client. Successivamente viene effettuata una chiamata ajax in cui il client comunica con il server in background. Vengono quindi passati i dati al server ed attraverso una funzione e all'ausilio della libreria xhtml2pdf viene generato un pdf contenente il codice. Infine, il server ritornerà il link del pdf al client e l'utente sarà in grado di visualizzarlo. Il codice-colore non è altro che una sequenza di quadratini colorati che l'utente dovrà leggere per poter colorare la griglia. È suddiviso in righe, tante quante sono le righe della griglia e va letto partendo dal quadratino in alto a sinistra, seguendo la riga da sinistra a destra e passando successivamente a quella sottostante. Per facilitare la lettura ogni riga è numerata. Il codice, inoltre, tiene conto delle ripetizioni in sequenza dei colori, per questo accanto ad ogni quadratino è presente un quadratino bianco con il numero di occorrenze del colore. In questa maniera il numero di colonne non è lo stesso per ogni riga e non è uguale al numero di colonne della griglia a meno che non capitino mai che due quadratini adiacenti orizzontalmente abbiano lo stesso colore.

All'interno del pdf oltre al codice-colore sono contenute:

- Una legenda dei colori utilizzati in formato hex. Questo perché durante la stampa sfumature dello stesso colore potrebbero non essere distinguibili.
- Una griglia bianca avente lo stesso numero di righe e colonne di quella generata per poter essere colorata una volta stampata.
- L'immagine della griglia colorata per poter controllare il risultato dell'esercizio.

4.2.3 Community

La community è un altro aspetto fondamentale della piattaforma. Permette la condivisione dei propri progetti con altri utenti. All'interno dell'architettura di Django la gestione lato server di questa funzionalità è contenuta nella sezione public mentre per la parte di front-end vi sono diverse pagine scritte in html, javascript e css contenute nelle sezioni templates e static. Per accedere alla pagina Community presente nel menù di navigazione l'utente non ha bisogno di effettuare il login e quindi di essere necessariamente registrato. Questo per permettere a tutti di comprendere e visualizzare ciò che la piattaforma consente di fare.

Effettuato l'accesso alla community è possibile vedere all'interno di una serie di cards l'anteprima di tutti i progetti resi pubblici dai loro autori. Per non affollare la pagina con troppi progetti si è optato per la pagination, ovvero vengono visualizzati solo un tot di progetti per pagina e con una barra presente alla fine della schermata l'utente è in grado di accedere a tutte le altre. È possibile spostarsi nella pagina successiva, tornare a quella precedente oppure accedere direttamente alla pagina iniziale o finale.

Ogni card contiene una miniatura dell'immagine della griglia colorata, il titolo, l'autore, la data di creazione e la categoria di appartenenza. Inoltre, è presente un bottone per accedere alla pagina di visualizzazione del progetto dove sono presenti tutte le informazioni e l'immagine della griglia nel formato originale.

Di default vengono visualizzati tutti i progetti da quello con la data di creazione più recente a quello con la data di creazione più obsoleta. Nella pagina però è possibile inserire dei filtri di ricerca per visualizzare solo i progetti con determinate caratteristiche e dei filtri di ordinamento. I filtri di ricerca presenti sono:

- Autore: inserendo in una casella di testo lo username dell'autore o parte di esso verranno visualizzati solo i progetti appartenenti all'autore cercato o i progetti degli autori contenenti nel proprio username il testo cercato.
- Titolo: inserendo in una casella di testo il titolo del progetto o parte di esso verranno visualizzati solo i progetti con il titolo cercato o i progetti che hanno il titolo contenente il testo cercato.
- Categoria: selezionando una categoria tra quelle proposte verranno visualizzati solo i progetti appartenenti alla categoria scelta.

I filtri descritti in precedenza possono anche essere inseriti contemporaneamente ed i progetti visualizzati avranno come caratteristiche la combinazione dei valori selezionati. I filtri di ordinamento invece sono:

- Autore: è possibile ordinare i progetti da visualizzare per autore decrescente o crescente.
- Titolo: è possibile ordinare i progetti da visualizzare per titolo decrescente o crescente.
- Data di creazione: è possibile ordinare i progetti da visualizzare per data di creazione decrescente o crescente.

Anche i filtri di ordinamento possono essere inseriti contemporaneamente e quindi l'ordinamento sarà la combinazione dei parametri selezionati. Una volta inseriti i filtri, l'utente dovrà cliccare il bottone Cerca, i dati verranno inviati al server che attraverso una funzione interrogherà il database con i dovuti filtri e ritornerà la lista dei progetti desiderati.

In maniera analoga alla community è presente una pagina chiamata I miei progetti a cui si può accedere dal menù di navigazione solo se si è loggati all'interno della piattaforma. In questa pagina sono visibili attraverso una serie di cards tutti i progetti creati dall'utente che ha effettuato l'accesso. Il funzionamento come detto in precedenza è analogo a quello della community ma sono presenti una serie di differenze. La prima differenza è che in ogni card non è presente il nome dell'autore poiché sarebbe sempre lo stesso, ma sono presenti in più la data di ultima modifica ed un bottone che permette di accedere in maniera diretta alla pagina di modifica del progetto. La seconda differenza è che non è presente il filtro per autore per il motivo già citato ma è possibile filtrare i progetti selezionando solo quelli pubblici o solo quelli privati. Infine, l'ultima differenza è che nei filtri di ordinamento non è presente ancora una volta l'ordinamento per autore ma è presente l'ordinamento per data di ultima modifica decrescente o crescente.

Capitolo 5

Analisi algoritmi di clustering

Per la pixelizzazione dell'immagine caricata dall'utente viene utilizzato l'algoritmo di clustering K-means. Come detto in precedenza però per questa funzionalità sono stati eseguiti dei confronti e dei test tra tre diversi algoritmi di clustering per comprendere quale fosse il più adeguato. Gli algoritmi analizzati sono quelli descritti e spiegati nel capitolo delle Tecnologie, essi sono il K-means, il K-medoids ed il DBSCAN. I primi due sono algoritmi partizionali mentre l'ultimo è un algoritmo basato sulla densità. Per questo motivo si è deciso di scartare subito il DBSCAN in quanto non permette di impostare a priori il numero di cluster da identificare e quindi di ottenere un unico centroide. Il DBSCAN calcola il numero di cluster in maniera automatica e questo anche se utile in molte applicazioni non risulta adatto alle necessità richieste per questa funzionalità.

Diverso è il confronto effettuato tra il K-means ed il K-medoids, essendo due algoritmi appartenenti alla stessa tipologia si è potuto valutare al meglio le prestazioni. La differenza sostanziale tra i due metodi è che nel K-means ogni oggetto della collezione è assegnato al cluster con il centroide più vicino. Il centroide calcolato è il baricentro e quindi può essere un punto artificiale non appartenente alla collezione. Nel K-medoids, invece, il centroide individuato è il punto tra quelli dati collocato più centralmente, di conseguenza è un punto appartenente alla collezione.

Per eseguire il test sono state scelte tre immagini (Figura 5.1) con caratteristiche differenti tra loro. In particolare, ciò che cambia è il numero di dettagli presenti e la risoluzione, ovvero il numero di pixel.



HOMER



MARE



SUPER MARIO

Figura 5.1: Immagini per il test

L'immagine con la risoluzione minore è 'Homer'. Essa è anche la meno dettagliata della tre poiché raffigura un unico personaggio. Aumentando la risoluzione abbiamo 'Super Mario' in cui aumenta anche il numero di dettagli dato che i personaggi raffigurati diventano due ed è presente uno sfondo. Infine, 'Mare' è l'immagine con la risoluzione più alta e presenta molti più dettagli in quanto è una foto scattata da un cellulare di un paesaggio reale. Un'altra differenza riguarda il formato, 'Homer' è un PNG rispetto alle altre due che sono JPG. Questo per valutare come si comportano i due algoritmi anche in caso di uno sfondo trasparente, caratteristica del formato PNG.

Scelte le immagini il test è stato effettuato facendo variare la dimensione della griglia per ogni immagine. Le dimensioni scelte sono 50, 40, 30, 20 ed indicano in quanti quadratini si vuole suddividere il lato più lungo dell'immagine. Moltiplicato il numero di righe e colonne di una griglia si ottiene il numero di quadrati indicato nelle tabelle successive. Questo, a livello computazionale, indica quante volte viene

ripetuto l'algoritmo, ovvero quanti centroidi si andranno a calcolare. La collezione, invece, è il numero di dati che l'algoritmo deve esaminare per calcolare un unico centroide. I parametri valutati per la scelta dell'algoritmo di clustering più adatto sono stati la resa dell'immagine pixellata ed il tempo computazionale dell'algoritmo, ovvero quanto ci vuole per processare l'immagine e fornire il risultato all'utente.

Prendendo in esame l'immagine 'Homer' possiamo notare (Figura 5.2) come i due algoritmi processano l'immagine al variare di tutte e quattro le dimensioni in ordine decrescente. Nella prima riga vi è il risultato del K-medoids mentre nella seconda vi è il risultato del K-means.

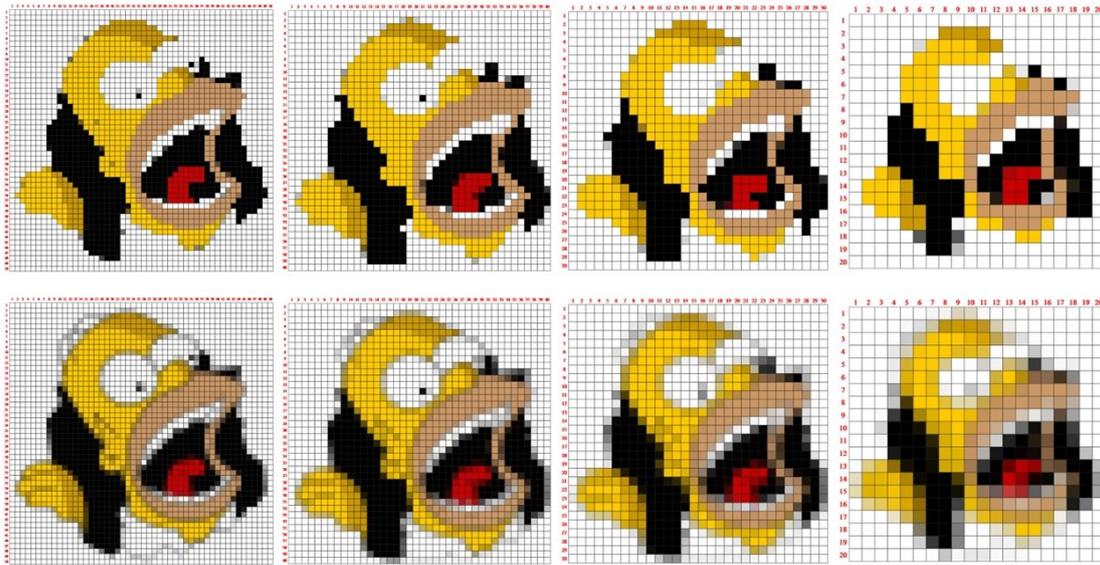


Figura 5.2: Test HOMER

Di seguito sono riportati la Tabella 5.1 che riassume le caratteristiche dell'immagine ed il grafico (Figura 5.3) contenente i tempi computazionali dei due algoritmi a confronto.

HOMER				
DIMENSIONE IMMAGINE	PIXEL TOTALI	DIMENSIONI GRIGLIA	NUMERO QUADRATI	COLLEZIONE
500x500	250000	20x20	400	625
500x500	250000	30x30	900	256
500x500	250000	40x40	1600	144
500x500	250000	50x50	2500	100

Tabella 5.1: Caratteristiche immagine HOMER

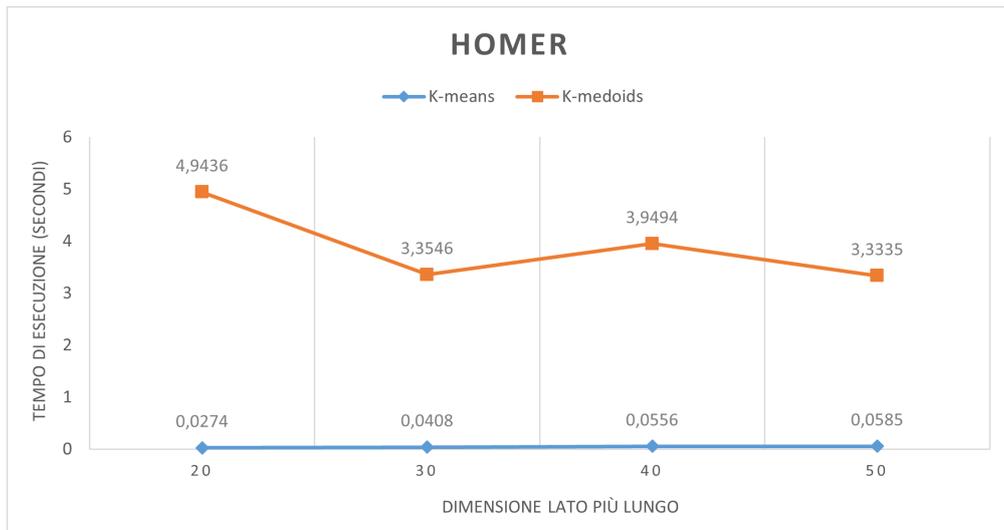


Figura 5.3: Grafico HOMER

Dai risultati ottenuti possiamo notare come in questo caso l'immagine è processata meglio dall'algoritmo di clustering K-means, poiché riesce ad interpretare meglio i contorni ed alcuni dettagli più sottili, come i capelli e la pupilla dell'occhio che con il K-medoids scompare al diminuire della dimensione. Per quanto riguarda i tempi di esecuzione dei due algoritmi entrambi risultano accettabili, il K-medoids ci impiega dai 3 ai 5 secondi a seconda della dimensione mentre il K-means rimane sempre sotto il secondo.

Analizzando l'immagine 'Super Mario', di seguito sono riportati la Tabella 5.2 che riassume le caratteristiche dell'immagine ed il grafico (Figura 5.4) contenente i tempi computazionali dei due algoritmi a confronto. In Figura 5.5 possiamo invece notare come i due algoritmi processano l'immagine per le dimensioni 50 e 40, rispettivamente a sinistra e a destra. Nella prima riga vi è il risultato del K-medoids mentre nella seconda vi è il risultato del K-means.

SUPER MARIO				
DIMENSIONE IMMAGINE	PIXEL TOTALI	DIMENSIONI GRIGLIA	NUMERO QUADRATI	COLLEZIONE
900x1600	1440000	11x20	220	6400
900x1600	1440000	16x30	480	2809
900x1600	1440000	22x40	880	1600
900x1600	1440000	28x50	1400	1024

Tabella 5.2: Caratteristiche immagine SUPER MARIO

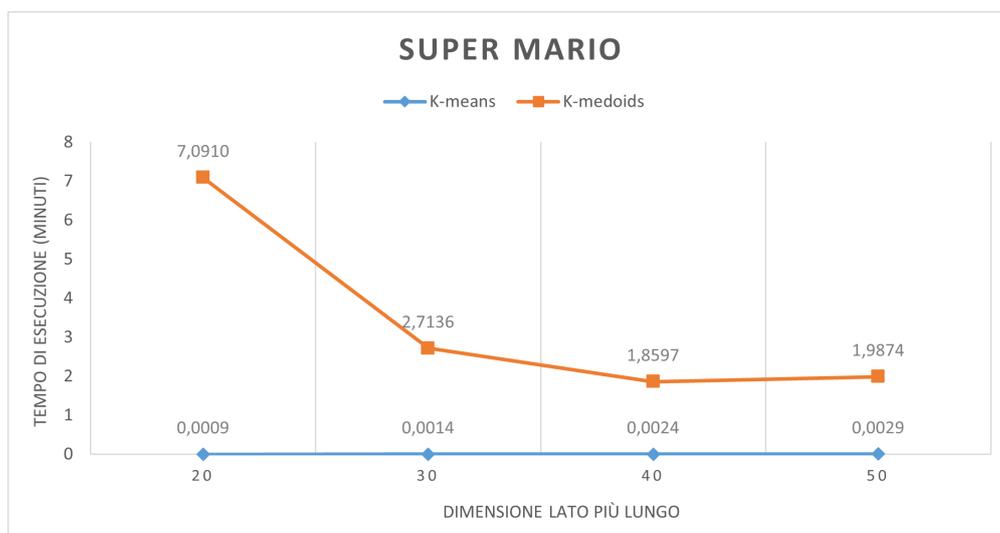


Figura 5.4: Grafico SUPER MARIO



Figura 5.5: Test SUPER MARIO

Confrontando i risultati si può notare questa volta che il K-medoids processa meglio l'immagine. In particolare i colori appaiono più accesi rispetto al K-means che, invece, dà un effetto più sfocato all'immagine. Questo è dovuto al fatto che i colori utilizzati dal K-medoids sono colori realmente presenti all'interno dell'immagine al contrario del K-means che calcolando il baricentro ottiene dei colori più sfumati.

In contrapposizione però i tempi di esecuzione dei due algoritmi si discostano di molto. Il tempo computazionale del K-medoids aumenta notevolmente al diminuire della dimensione impiegandoci dai 7 ai 2 minuti mentre il K-means rimane sempre sotto il secondo.

Infine, prendendo in considerazione l'immagine 'Mare' sono riportati la Tabella 5.3 che riassume le caratteristiche dell'immagine ed il grafico (Figura 5.6) contenente i tempi computazionali dei due algoritmi a confronto. In Figura 5.7 possiamo invece notare come i due algoritmi processano l'immagine per le dimensioni 50 e 30, rispettivamente a sinistra e a destra. Nella prima riga vi è il risultato del K-medoids mentre nella seconda vi è il risultato del K-means.

MARE				
DIMENSIONE IMMAGINE	PIXEL TOTALI	DIMENSIONI GRIGLIA	NUMERO QUADRATI	COLLEZIONE
1536x2048	3145728	15x20	300	1600
1536x2048	3145728	22x30	660	2601
1536x2048	3145728	30x40	1200	4624
1536x2048	3145728	38x50	1900	10404

Tabella 5.3: Caratteristiche immagine MARE

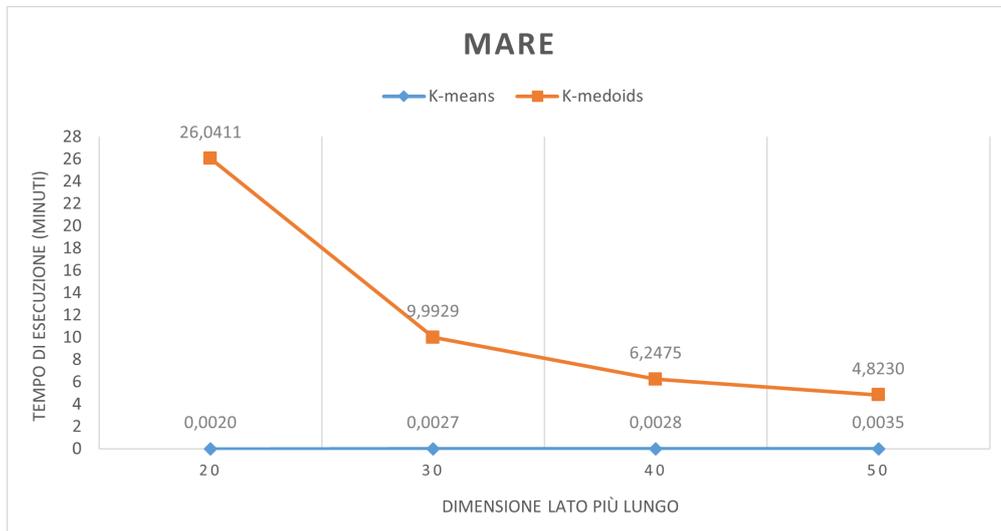


Figura 5.6: Grafico MARE



Figura 5.7: Test MARE

Il risultato ottenuto da quest'ultimo esperimento è molto simile al risultato analizzato con l'immagine 'Super Mario' in cui il K-medoids processa meglio l'immagine rispetto al K-means, il quale genera un'immagine dai colori più sfocati. Con l'aumento del numero dei pixel dell'immagine però, il distacco tra il tempo di esecuzione del K-medoids e del K-means risulta ancora più netto. Il primo ci impiega un tempo compreso tra i 26 e i 5 minuti aumentando al diminuire della dimensione, mentre il K-means rimane sempre sotto il secondo.

In conclusione, l'algoritmo di clustering K-medoids anche se nella maggior parte dei casi processa meglio l'immagine, impiega un tempo computazionale troppo elevato e variabile rispetto al K-means, il quale rimane pressoché costante e sotto il secondo. Inoltre, al diminuire delle dimensioni della griglia il tempo di esecuzione del K-medoids aumenta anche se diminuisce il numero di ripetizioni dell'algoritmo. Questo perché aumenta la collezione dei dati da processare, ciò significa che il tempo impiegato per calcolare il centroide è molto maggiore rispetto alla ripetizione dell'algoritmo. Si è scelto quindi di utilizzare come algoritmo di clustering per la

pixellizzazione dell'immagine il K-means poichè offre in ogni caso ottimi risultati in un tempo decisamente ragionevole e costante.

Capitolo 6

Casi d'uso

Per casi d'uso si intende la descrizione di un insieme di interazioni tra un utente ed un sistema che consentono all'utente di raggiungere un obiettivo o di svolgere un compito. Solitamente questa descrizione viene prodotta a fronte dell'esplicitazione di una serie di requisiti da parte di un utente nei confronti del sistema da realizzare. In tal senso è finalizzata a precisare meglio i requisiti espressi ed a verificare insieme con l'utente che rispondano alle sue aspettative. La definizione di casi d'uso è quindi utilizzata durante la raccolta dei requisiti e guiderà le attività di controllo dei deliverable prodotti dal progetto. Ciò avviene in particolare nei progetti di ingegneria del software dove si utilizzano diagrammi UML come quello riportato in Figura 6.1.

Un caso d'uso è definito da alcuni elementi chiave. L'**attore** è l'utente che interagisce con il sistema, solitamente per svolgere un determinato compito e per raggiungere un proprio obiettivo. Non si tratta di un individuo specifico, ma qualcuno che ricopre un certo ruolo ovvero una categoria che interagisce in maniera analoga con il sistema informativo. L'**obiettivo** (Goal) è un qualcosa che l'attore può raggiungere tramite l'aiuto del sistema informativo; deve trattarsi di qualcosa che ha un valore specifico per l'utente. È espresso dal punto di vista dell'attore e rappresenta le funzioni che il sistema informativo può offrire all'attore. Il **sistema** rappresenta quello che deve essere specificato e costruito e viene visto come una scatola nera di cui si osserva il limite e come interagisce con gli attori, ma i cui dettagli interni sono lasciati non descritti, nascosti.

Di seguito verranno spiegati i casi d'uso più importanti delle parti elaborate nella piattaforma. In particolare verrà mostrato come le parti implementate e spiegate nel capitolo precedente, possono essere usate nella piattaforma.

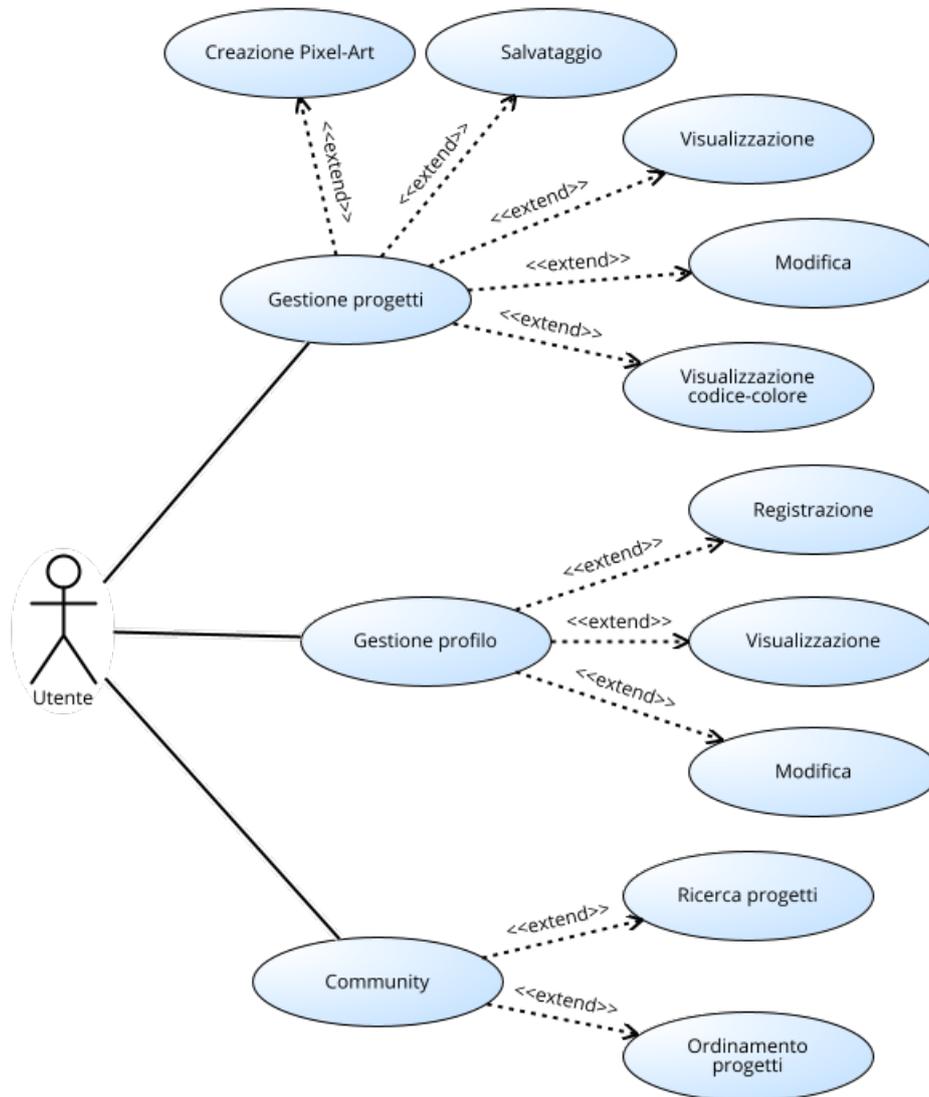


Figura 6.1: Diagramma UML dei casi d'uso del sistema

6.1 Creazione Pixel-art

Quando si parla della creazione di una Pixel-art si intendono tutte quelle funzionalità che l'utente dispone per generare la griglia, colorarla e modificarla prima di salvare il progetto. In Figura 6.2 è possibile visualizzare la pagina Crea che permette all'utente di accedere a queste funzionalità.

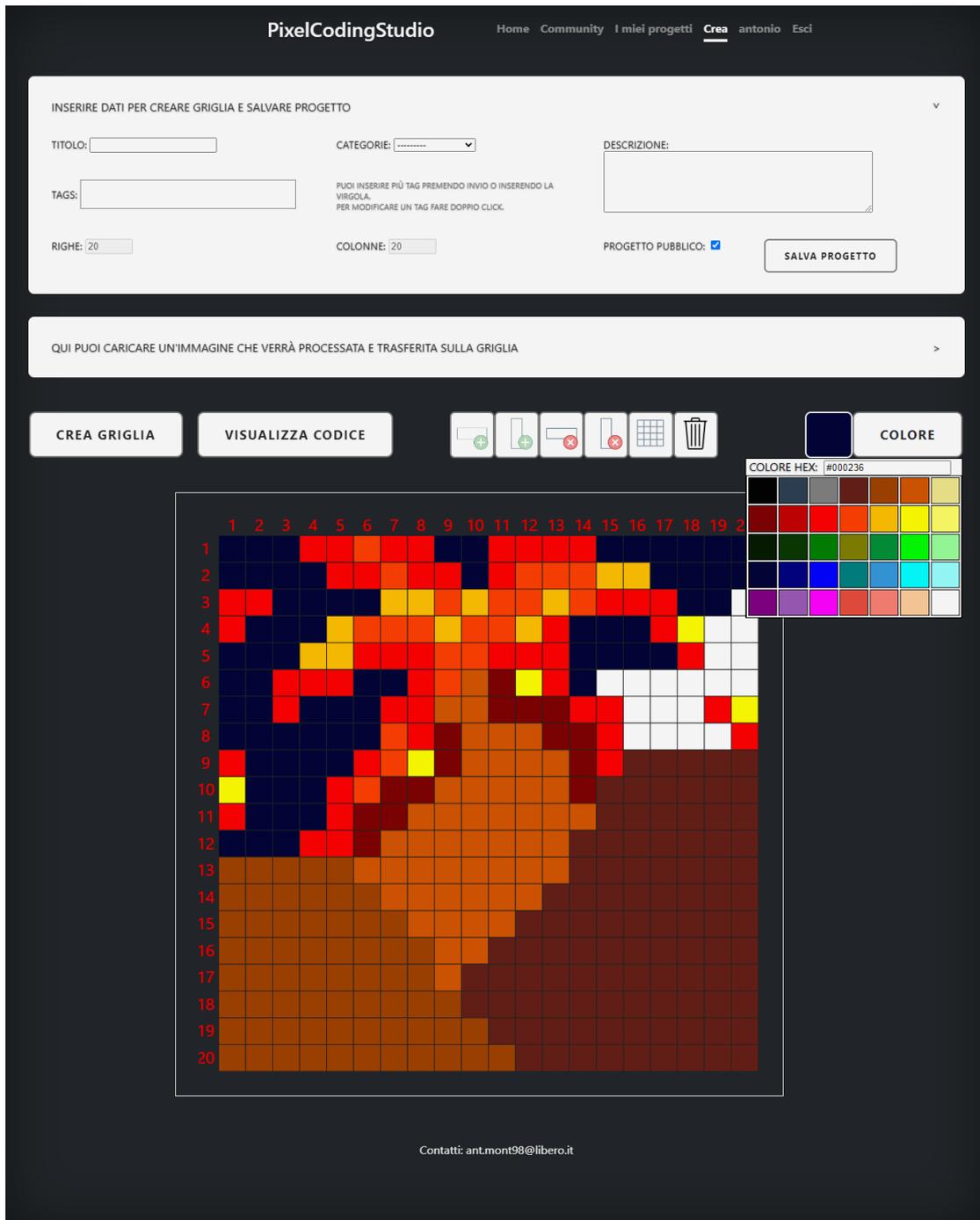


Figura 6.2: Pagina Crea

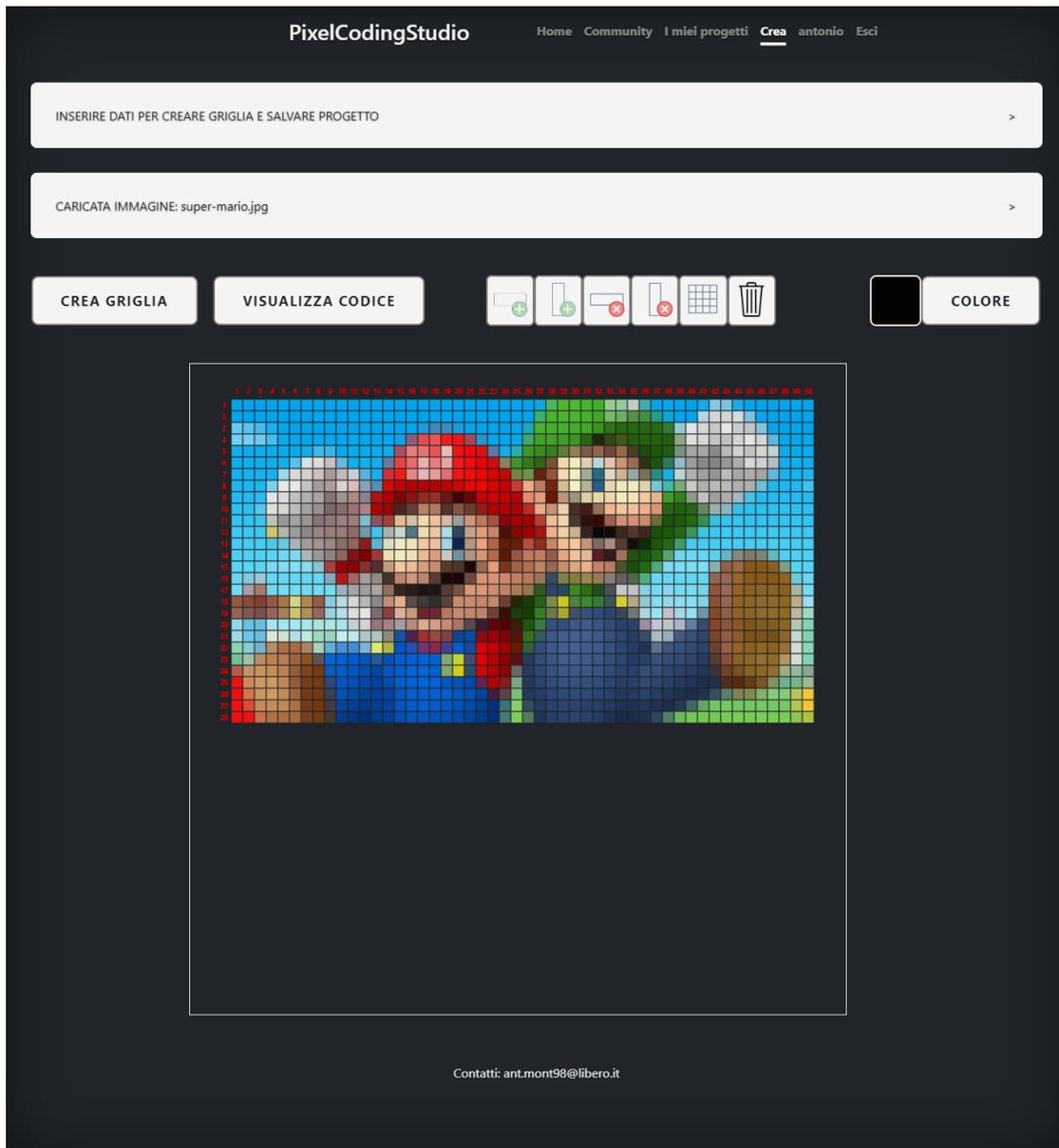
L'utente per generare la griglia deve prima impostare il numero di righe e colonne inserendo un numero nelle caselle di testo denominate rispettivamente RIGHE e COLONNE. Il numero deve essere compreso tra 2 e 50. Premendo il pulsante CREA GRIGLIA verrà successivamente visualizzata sullo schermo una griglia bianca dalle dimensioni scelte che l'utente potrà andare a colorare. Premendo il pulsante COLORE apparirà una tendina contenente una palette di colori predefinita ed una casella di testo in cui è possibile inserire il codice hex di un colore qualsiasi che si vuole utilizzare anche non presente all'interno della palette. Per selezionare, invece, un colore appartenente alla palette l'utente deve cliccarci sopra. Effettuata la scelta è possibile colorare la griglia cliccando con il mouse su ogni singolo quadratino o scorrendo con il mouse ed il tasto sinistro premuto sopra la griglia.

Creata la griglia, i valori inseriti in RIGHE e COLONNE non possono più essere modificati nel campo di testo direttamente dall'utente, per questo se si vogliono modificare le dimensioni della griglia in fase di colorazione sono presenti dei pulsanti con delle icone. Il pulsante con l'icona di un rettangolo orizzontale ed il simbolo + di colore verde permette di aggiungere una riga dal basso, invece, il pulsante con l'icona di un rettangolo verticale ed il simbolo + di colore verde permette di aggiungere una colonna da destra. In contrapposizione, il pulsante con l'icona di un rettangolo orizzontale ed il simbolo x di colore rosso permette di eliminare una riga dal basso, mentre il pulsante con l'icona di un rettangolo verticale ed il simbolo x di colore rosso permette di eliminare una colonna da destra. Così facendo il numero di righe e colonne verrà aggiornato di conseguenza aumentando o diminuendo il valore di uno alla volta. Inoltre, per pulire l'intera griglia facendola ritornare tutta bianca e mantenendo le dimensioni scelte in precedenza, al posto di selezionare il colore bianco e colorare tutti i quadratini, è presente un pulsante con l'icona di una griglia bianca che velocizza questa funzione. Infine, cliccando sull'icona del cestino è possibile cancellare l'intera griglia e successivamente andare a reimpostare le dimensioni per crearne una nuova. Per facilitare la comprensione di ciò che i pulsanti appena descritti permettono di fare, quando si passa con il mouse sopra l'icona verrà visualizzato un tooltip, ovvero un breve messaggio a comparsa che ne descrive la funzionalità.

Sappiamo dal capitolo precedente che un utente può creare la propria Pixel-art partendo da zero o caricando un'immagine. Precedentemente è stato spiegato cosa l'utente deve fare per poter creare la propria Pixel-art da zero. Di seguito, invece verrà spiegato in che modo l'utente può creare la propria Pixel-art caricando un'immagine. In Figura 6.1 è possibile visualizzare il modulo per il caricamento dell'immagine, mentre in Figura 6.4 è possibile visualizzare nuovamente la pagina Crea ma con il caricamento di un'immagine scelta dall'utente.



Figura 6.3: Modulo per caricare immagine



Per creare la Pixel-art, nell'apposito modulo è presente un pulsante denominato Scegli file che permette di scegliere un'immagine dal proprio computer e caricarla. Per generare la griglia è inoltre necessario inserire una dimensione nel campo di testo DIMENSIONE. Il numero può essere un valore compreso tra 15 e 50. Spuntando la casella sotto la dicitura UTILIZZARE SOLO COLORI PRESENTI NELLA PALETTE si sceglie appunto di visualizzare l'immagine processata colorata solo con i colori presenti nella palette. Inseriti tutti i parametri, per generare la griglia con l'immagine processata bisogna premere il pulsante CARICA IMMAGINE. Una volta visualizzata, anche in questo caso è possibile colorare ulteriormente la griglia o modificarla nelle dimensioni. Il funzionamento è lo stesso di quello descritto in precedenza.

6.2 Salvataggio progetti

Salvare un progetto vuol dire inserire tutti i dati necessari che caratterizzano la Pixel-art creata e memorizzarli all'interno del database. Per fare questo è presente un modulo da compilare nella pagina Crea, come si può vedere in Figura 6.5.

INSERIRE DATI PER CREARE GRIGLIA E SALVARE PROGETTO

TITOLO:

CATEGORIE:

DESCRIZIONE:

TAGS:

PUOI INSERIRE PIÙ TAG PREMENDO INVIO O INSERENDO LA VIRGOLA. PER MODIFICARE UN TAG FARE DOPPIO CLICK.

RIGHE:

COLONNE:

PROGETTO PUBBLICO:

Figura 6.5: Modulo per salvare progetto

L'utente per salvare un progetto deve inserire un titolo nel campo di testo denominato TITOLO, deve scegliere una categoria tra quelle presenti nella lista denominata CATEGORIE e inserire almeno un tag nella casella di testo denominata TAGS. Per inserire più di un tag basta premere invio o inserire una virgola tra un tag e l'altro. Infine, deve decidere se rendere il progetto pubblico e quindi visibile nella community spuntando la casella sotto la dicitura PROGETTO PUBBLICO che di default è già spuntata oppure renderlo privato e quindi visibile solo a se stesso nella pagina I miei progetti levando il simbolo di spunta. Inoltre, può decidere di inserire una descrizione nel campo di testo denominato DESCRIZIONE. Il numero di RIGHE e COLONNE è il valore inserito in precedenza per generare la griglia. Premendo il pulsante SALVA PROGETTO l'utente avrà quindi memorizzato i dati

della propria Pixel-art e potrà visualizzarla e modificarla in un secondo momento quante volte desidera.

6.3 Visualizzazione codice-colore

Con la creazione della Pixel-art è possibile visualizzare il relativo codice-colore. Cliccando sul pulsante VISUALIZZA CODICE presente nella pagina Crea (Figura 6.2) verrà generato un pdf in cui è possibile visualizzare il codice-colore. L'esempio riportato in Figura 6.6 mostra il pdf contenente il codice-colore relativo alla griglia generata e colorata in Figura 6.2.

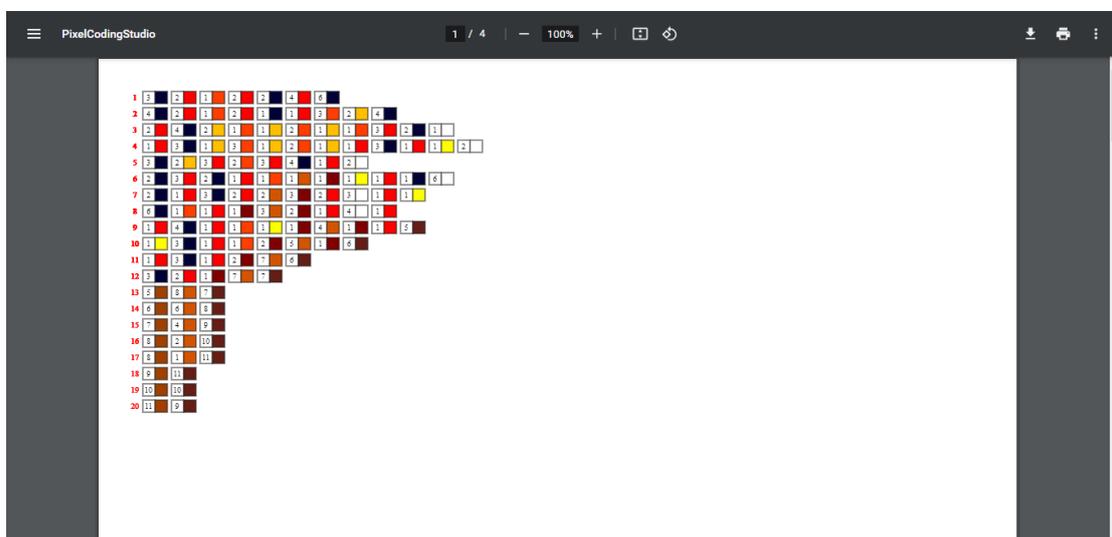


Figura 6.6: Pdf contenente codice-colore

All'interno del pdf è anche presente la legenda dei colori utilizzati in formato hex. La legenda è suddivisa in righe ed ogni codice hex corrisponde al colore del relativo quadratino presente nel codice-colore, come si può vedere in Figura 6.7.

Sono inoltre presenti una griglia bianca (Figura 6.8) avente lo stesso numero di righe e colonne della griglia generata dall'utente ed un'immagine della Pixel-art (Figura 6.9).

L'utente può quindi scaricare e stampare il pdf. In questo modo sarà capace di colorare la griglia bianca seguendo le indicazioni del codice-colore e confrontare il risultato ottenuto con la Pixel-art presente alla fine. In caso di dubbi sul colore da utilizzare può consultare la legenda.

Il pdf contenete queste informazioni può essere visualizzato non solo se il progetto è stato salvato ma anche in fase di colorazione. L'unico requisito necessario è che la griglia sia stata generata.

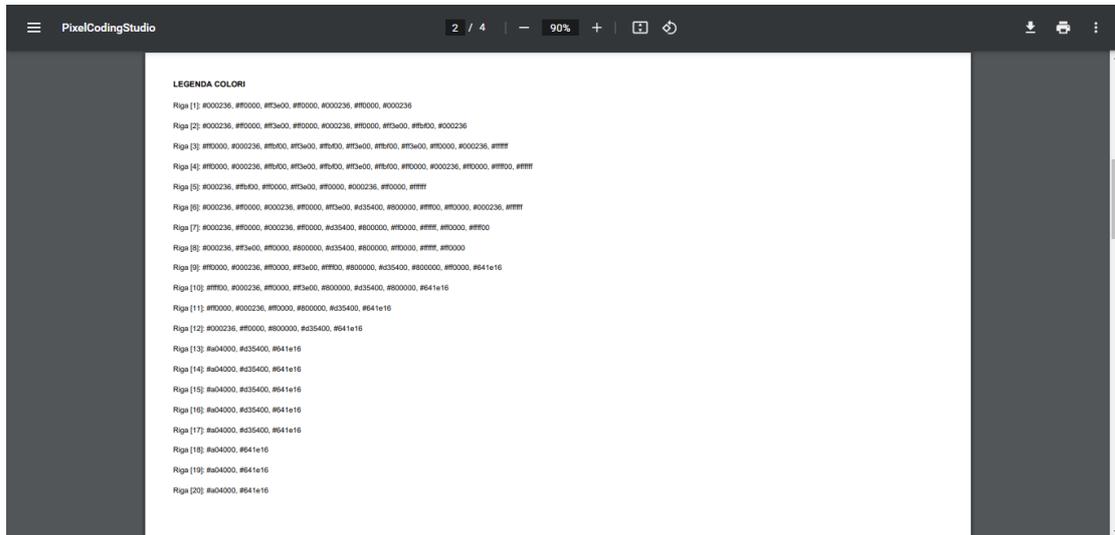


Figura 6.7: Pdf contenente legenda colori

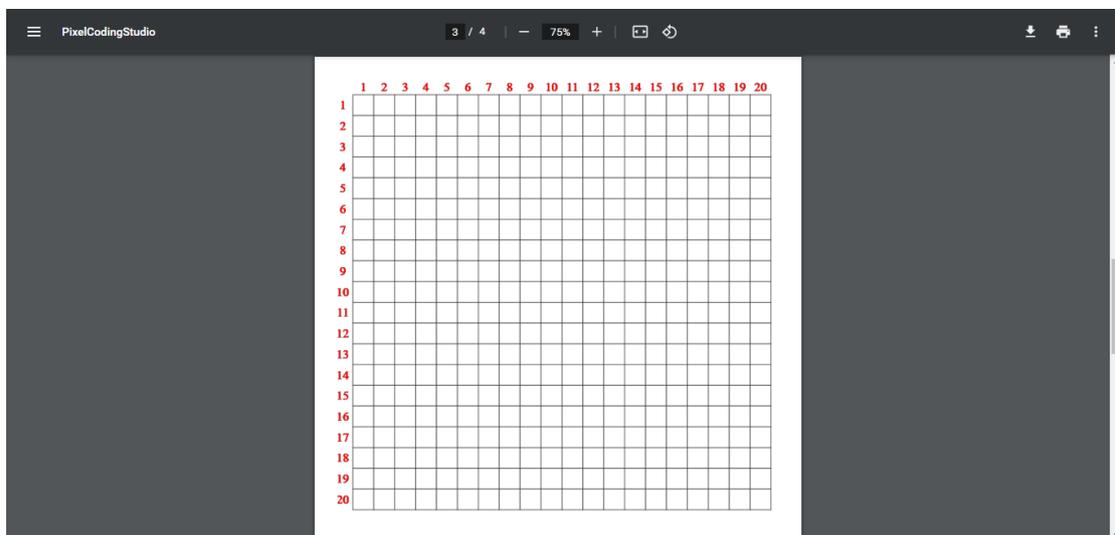


Figura 6.8: Pdf contenente griglia bianca

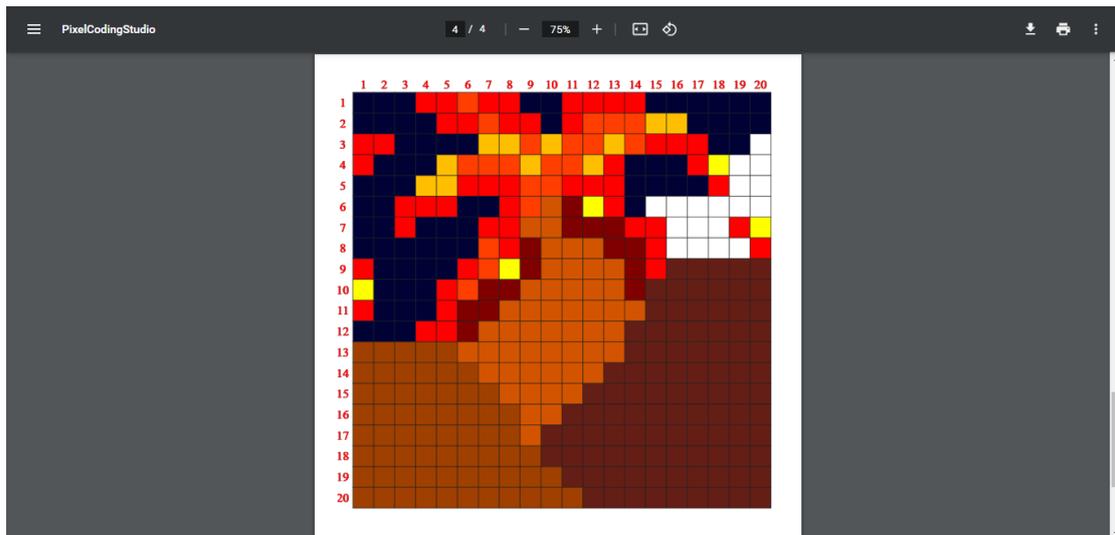


Figura 6.9: Pdf contenente Pixel-art

6.4 Ricerca progetti

Effettuando l'accesso alla pagina Community (Figura 6.10) l'utente sarà in grado di visualizzare, ricercare e ordinare tutti i progetti condivisi dagli altri utenti. I progetti sono visibili all'interno delle cards. Ogni card contiene un'anteprima della Pixel-art insieme ad alcune informazioni come il titolo del progetto, l'autore, la data di creazione e la categoria di appartenenza. È inoltre presente un pulsante VISUALIZZA che permette di accedere alla pagina contenente tutte le informazioni del progetto e l'immagine della Pixel-art per intero.

Nella community possono essere visualizzati solamente 6 progetti alla volta ma grazie alla barra presente in fondo alla schermata è possibile cambiare pagina visualizzando i progetti successivi o quelli precedenti. Il numero totale di progetti trovati e quindi visualizzabili è presente nel modulo di ricerca.

Per effettuare la ricerca dei progetti l'utente dispone di alcuni filtri da poter inserire nell'apposito modulo come l'autore, il titolo e la categoria. Di default i progetti sono mostrati dal più recente al più vecchio ma l'utente può decidere di visualizzarli in maniera differente impostando i filtri di ordinamento sul valore crescente o decrescente. I progetti possono essere ordinati per autore, titolo e data.

Impostati i filtri e cliccato sul pulsante CERCA verranno mostrati i progetti in base alla ricerca e all'ordine scelto dall'utente. Il pulsante CANCELLA, invece, permette di resettare tutti i filtri tornando alle impostazioni di default.

The screenshot shows the 'PixelCodingStudio' website's 'Community' page. At the top, there is a navigation bar with 'Home', 'Community', 'I miei progetti', 'Crea', 'admin', and 'Esci'. Below the navigation bar is a search filter section with the heading 'INSERIRE FILTRI PER EFFETTUARE LA RICERCA'. The filter includes fields for 'RICERCA PER:', 'AUTORE:', 'TITOLO:', and 'CATEGORIA:'. There are also dropdown menus for 'ORDINA PER:' and 'DATA:'. Below the filter are 'CERCA' and 'CANCELLA' buttons, and a message that says 'SONO STATI TROVATI 32 PROGETTI!'. The main content area displays a grid of six pixel art projects, each with a grid overlay and a description. The projects are: 'Homer' (created by admin), 'Super Mario' (created by admin), 'smile' (created by antonio), 'Quadro' (created by antonio), 'Vulcano' (created by antonio), and 'panino di ari' (created by ariii). Each project has a 'VISUALIZZA' button and a category label. At the bottom of the grid is a pagination control showing '« prima precedente 1 di 6 successiva ultima »'. The footer contains the contact information 'Contatti: ant.mont98@libero.it'.

Figura 6.10: Pagina Community

6.5 Registrazione

La registrazione a PixelCodingStudio è un aspetto fondamentale per poter effettuare il login e quindi accedere ad alcune funzionalità importanti come la creazione di un progetto e la visualizzazione del pdf contenente il codice-colore. Essa non è

necessaria invece per accedere alla community e visualizzare i progetti condivisi. In Figura 6.11 è mostrata la pagina Registrati contenente i campi che l'utente deve compilare per registrarsi nella piattaforma.

PixelCodingStudio Home Community Accedi Registrati

Compila modulo di registrazione:

Nome utente: Antonio98 Obbligatorio. 150 caratteri o meno.Solo lettere, cifre e @/./+/_

Nome: Antonio Obbligatorio.

Cognome: Monteleone Obbligatorio.

Sesso: maschio

Data di nascita: 2004-04-13 Obbligatorio. Formato: YYYY-MM-DD

Città: Taranto Opzionale.

Paese: Italia Opzionale.

Immagine Profilo: Scegli file Nessun file selezionato Opzionale.

Email: ant.mont98@libero.it Obbligatorio. Immetti un indirizzo email valido.

Email genitore: Obbligatorio. Immetti un indirizzo email valido di un genitore.

Password:

- La tua password non può essere troppo simile alle altre tue informazioni personali.
- La tua password deve contenere almeno 8 caratteri.
- La tua password non può essere una password comunemente usata.
- La tua password non può essere interamente numerica.

Conferma password: Inserisci la stessa password inserita sopra, come verifica.

Registrati

Contatti: ant.mont98@libero.it

Figura 6.11: Pagina Registrati

L'utente prima di cliccare il pulsante Registrati effettuando la registrazione, deve inserire alcuni dati obbligatori nei campi denominati Nome utente, Nome, Cognome, Sesso, Data di nascita, Email e Password. L'Email genitore invece, è un campo che viene visualizzato e di conseguenza diventa obbligatorio solo se l'utente risultasse minorenne. Infine, la Città, il Paese e l'Immagine Profilo sono campi facoltativi.

Capitolo 7

Conclusioni

L'obiettivo principale della Tesi è stato quello di introdurre gli utenti al pensiero computazionale grazie alla creazione di esercizi di coding unplugged e la successiva condivisione di essi interfacciandosi con la piattaforma web PixelCodingStudio. A seguito della progettazione, si può affermare di aver raggiunto i requisiti principali per avere un sistema che permette la creazione di progetti attraverso la Pixel-art e la condivisione di essi all'interno di una community. In particolare, è stato sviluppato un sistema semplice ed intuitivo che permette agli utenti di registrarsi all'interno della piattaforma, generare e colorare griglie per la creazione dalla propria Pixel-art, inserire i dati e salvare il proprio progetto potendolo condividere con la community, visualizzare un pdf scaricabile e stampabile contenente il codice-colore per ricreare la propria Pixel-art, la legenda dei colori da utilizzare, una griglia bianca da colorare e l'immagine della griglia colorata per verificare l'esito dell'esercizio. Infine, un sistema che permette la visualizzazione dei propri progetti, la modifica e la visualizzazione di tutti i progetti condivisi con la possibilità di ricerca e ordinamento.

Oltre ai requisiti minimi previsti per soddisfare l'obiettivo principale è stato raggiunto un ulteriore obiettivo che aggiunge una funzionalità interessante ed innovativa alla piattaforma. Grazie all'analisi, allo studio e al test di diverse tecniche di clustering si è creato un sistema che permette all'utente di caricare un'immagine per creare la propria Pixel-art e visualizzarne il relativo codice-colore, senza dover necessariamente generare una griglia da zero e colorarla. Sono state analizzate principalmente due tecniche di clustering, il K-means ed il K-medoids. Da questo confronto il K-means è risultato essere l'algoritmo più idoneo, in quanto, anche se in alcuni casi l'immagine ottenuta sia più sfocata e meno nitida, il tempo di esecuzione per il processamento dell'immagine è decisamente minore e più costante rispetto al tempo di esecuzione ottenuto con il K-medoids. Essendo PixelCodingStudio un'applicazione web, le funzionalità richiedono tempi di risposta immediati.

7.1 Sviluppi futuri

La progettazione e lo sviluppo di PixelCodingStudio ha quindi dato risultati positivi e molto soddisfacenti, ma ci potrebbero essere sviluppi futuri davvero interessanti. In primis un aspetto che potrebbe sicuramente essere migliorato è la staticità grafica della piattaforma. Allo stato attuale, tecnicamente è possibile accedere all'applicazione su qualsiasi dispositivo. Dal punto di vista pratico questo non è del tutto possibile in quanto le pagine non si adattano agli schermi di dimensioni ridotte andando a ridurre quasi del tutto a zero la possibilità di fruizione del sistema. Il target preso in considerazione per lo sviluppo dell'interfaccia grafica è rappresentato da computer e laptop, pertanto, l'obiettivo sarebbe quello di estendere l'utilizzo anche per tablet andando ad implementare una grafica più responsive. Inoltre, la progettazione della piattaforma non si è concentrata molto sul design e l'aspetto estetico, quindi, oltre ad una grafica adattabile a tutti i dispositivi, essa potrebbe essere migliorata anche nell'aspetto.

Infine, una funzionalità molto interessante che potrebbe essere aggiunta ed implementata è la creazione dei contests. L'amministratore del sito potrebbe avere la possibilità di creare e lanciare alcuni contests all'interno della piattaforma alla quale possono iscriversi e partecipare gli utenti condividendo i propri progetti. I contests potrebbero prevedere determinate regole di partecipazione come ad esempio la condivisione di progetti appartenenti ad una determinata categoria. Essi avrebbero una durata limitata ed al termine di ogni contest si potrebbero votare secondo alcuni criteri i progetti partecipanti, stilando una classifica con la successiva premiazione dei vincitori.

Bibliografia

- [1] «Coding». In: (2022). URL: <https://www.sprintlab.it/blog/coding/> (cit. a p. 4).
- [2] «Pensiero computazionale». In: (2022). URL: <https://www.sprintlab.it/blog/pensiero-computazionale/> (cit. a p. 6).
- [3] «Coding Unplugged». In: (2022). URL: <https://codemooc.org/teoria-e-pratica-del-coding-unplugged/> (cit. a p. 13).
- [4] «Pixel-art». In: (2022). URL: <https://codemooc.org/pixel-art/> (cit. a p. 15).
- [5] «Docker». In: (2022). URL: <https://www.ibm.com/it-it/cloud/learn/docker> (cit. a p. 18).
- [6] «Python». In: (2022). URL: <https://www.python.org/> (cit. a p. 21).
- [7] «OpenCV». In: (2022). URL: <https://opencv.org/> (cit. a p. 21).
- [8] «Scikit-learn». In: (2022). URL: <https://www.intelligenzaartificialeitalia.net/post/introduzione-a-scikit-learn-con-esempio-pratico-in-python> (cit. a p. 22).
- [9] «xhtml2pdf». In: (2022). URL: <https://xhtml2pdf.readthedocs.io/en/latest/> (cit. a p. 22).
- [10] «Django». In: (2022). URL: <https://www.djangoproject.com/> (cit. a p. 23).
- [11] «Flask». In: (2022). URL: <https://flask.palletsprojects.com/en/2.2.x/> (cit. a p. 25).
- [12] «Web2py». In: (2022). URL: <http://www.web2py.com/> (cit. a p. 26).
- [13] «Django-SES». In: (2022). URL: <https://github.com/django-ses/django-ses> (cit. a p. 27).
- [14] «JavaScript». In: (2022). URL: <https://www.javascript.com/> (cit. a p. 27).
- [15] «jQuery». In: (2022). URL: <https://www.html.it/guide/guida-jquery/> (cit. a p. 28).

- [16] «D3.js». In: (2022). URL: <https://d3js.org/> (cit. a p. 28).
- [17] «HTML». In: (2022). URL: <https://boolean.careers/blog/html-cose-come-funziona-e-a-cosa-serve> (cit. a p. 29).
- [18] «CSS». In: (2022). URL: <https://www.html.it/pag/14209/introduzione1/> (cit. a p. 30).
- [19] «Bootstrap». In: (2022). URL: <https://www.html.it/pag/44806/introduzione-a-bootstrap/> (cit. a p. 30).
- [20] «MySQL». In: (2022). URL: <https://www.html.it/guide/guida-mysql/> (cit. a p. 31).
- [21] «PhpMyAdmin». In: (2022). URL: <https://www.html.it/pag/32011/introduzione84/> (cit. a p. 33).