

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

Dipartimento di Informatica - Scienza e Ingegneria
Corso di Laurea in Ingegneria e Scienze Informatiche

**PROGETTAZIONE E SVILUPPO DI UN ECOSISTEMA
DIGITALE PER LA GESTIONE DEGLI AVVISTAMENTI DI
ANIMALI MARINI**

Elaborato in
Basi di Dati

Relatore:
Annalisa Franco
Correlatore:
Annalisa Zaccheroni

Presentata da:
Andrea Bedei

Anno Accademico 2022/2023

*La vita non è complessa.
Siamo noi a essere complessi.
La vita è semplice.*
— Oscar Wilde

Sommario

La presente tesi di Laurea nasce dalla volontà di creare uno strumento pratico e concreto a sostegno di un progetto del dipartimento di Acquacoltura dell’Università di Bologna. I ricercatori di tale Dipartimento stanno cercando di monitorare gli avvistamenti di diverse specie marine nel nostro territorio e, per farlo in maniera sistematica, necessitano di un’applicazione che permetta di catalogare, controllare e successivamente studiare, tutti gli avvistamenti effettuati durante le loro uscite in mare.

L’obiettivo di questo studio è quello di affiancare i ricercatori nel loro lavoro, in particolare nel compito di monitoraggio che effettuano per comprendere al meglio il comportamento dei singoli individui e le relazioni che instaurano tra loro e con l’ambiente. Lo scopo finale è quello di individuare delle soluzioni ambientali affinché gli esemplari studiati non si sentano minacciati nei nostri mari e possano continuare a popolarli.

Per riuscire a creare un sistema efficiente ed efficace, si ha la necessità di un’intera infrastruttura: in primis una banca dati che possa mantenere e conservare gli esiti dei monitoraggi, e poi un portale che permetta la registrazione di nuovi avvistamenti, ma anche la modifica e la cancellazione di elementi già inseriti. Per ogni avvistamento, è necessaria anche la possibilità di caricare immagini che consentano di individuare esemplari o addirittura di visualizzare le loro eventuali e talvolta numerose ferite. Per questo, è stata proposta ai ricercatori la possibilità dell’utilizzo di una tecnica di riconoscimento automatico durante la fase di inserimento o di modifica dei

dati per individuare più velocemente la specie di appartenenza di ogni singolo individuo.

L'applicazione creata con questo progetto può essere utilizzata non solo per scopi di ricerca, ma anche al fine di creare una grande comunità che abbia la possibilità di scambiarsi informazioni e immagini per conoscere, studiare e migliorare l'ambiente marino e la sua fauna.

RINGRAZIAMENTI

Sono orgoglioso di dedicare questo spazio del mio elaborato alle persone che mi hanno supportato nella stesura della tesi e anche durante l'intero corso di laurea. Sono convinto che senza di loro non sarei arrivato fino a questo punto.

In primis, ringrazio le relatrici della tesi, in particolare Annalisa Franco che, durante la stesura dell'elaborato, mi ha sempre aiutato con grande professionalità ed estrema disponibilità. Credo che il primo "seme" di questa tesi sia stato gettato durante le sue lezioni nelle quali, con il suo atteggiamento sereno e tranquillo, è riuscita a creare molta empatia con tutti noi studenti. Per questo, le ho chiesto di seguirmi in qualità di relatrice e, dopo aver trovato insieme un argomento avvincente ed interessante,abbiamo iniziato a lavorarci.

Ringrazio i miei genitori e i miei familiari che mi hanno sostenuto durante tutto il percorso universitario perché, senza di loro, non ce l'avrei fatta.

In particolare, ringrazio mia sorella che, nonostante i momenti di difficoltà, mi ha sempre aiutato: grazie alle sue parole sono riuscito a dare il meglio di me.

Un altro ringraziamento va a mia nonna che, insieme ai miei genitori, ha sempre creduto in me e mi ha aiutato nei momenti difficili grazie alla sua saggezza. Posso dire di sentirmi veramente in debito con tutte queste persone.

Ringrazio tutti i miei amici che mi sono stati vicino, che mi hanno supportato e sono stati al mio fianco anche quando mi passavano per la testa le idee più strane, molte volte irrealizzabili.

Infine, dedico questa tesi a me stesso, perché non pensavo minimamente di riuscire a diventare quello che sono e sono certo del fatto che, grazie a

tutto questo percorso, oggi riesco ad avere più fiducia in me stesso. Posso dire di essere estremamente soddisfatto di questa avventura.

Indice

1	Introduzione	9
1.1	Diagramma dei casi d'uso	10
2	Documentazione Base di dati	11
2.1	Analisi dei requisiti	11
2.1.1	Intervista	11
2.1.2	Rilevamento delle ambiguità e correzioni proposte	13
2.1.3	Estrazione dei concetti principali	15
2.2	Progettazione Concettuale	16
2.2.1	Schema scheletro	16
2.2.2	Raffinamenti attuati	19
2.2.3	Schema concettuale finale	19
2.3	Progettazione Logica	20
2.3.1	Stima del volume dei dati	21
2.3.2	Descrizione delle operazioni principali e stima della loro frequenza	22
2.3.3	Schemi di navigazione e tabelle degli accessi	22
Accessi totali	26
2.3.4	Raffinamento dello schema	27

2.3.5	Traduzione di entità e associazioni in relazioni	27
2.3.6	Schema relazionale finale	29
3	Architettura del sistema	30
3.1	Struttura	30
4	Progettazione dell’interfaccia Web	32
4.1	Progettazione	32
4.1.1	Usabilità	33
Personas e scenarios	33	
4.1.2	Design	34
Mockup	35	
4.1.3	Solution stack	39
LAMP	39	
PHP	40	
MYSQL	41	
4.1.4	Fruizione del servizio	41
URI	42	
HTTPS	42	
HTML	43	
4.1.5	Stile	43
CSS	43	
Bootstrap	44	
4.1.6	Javascript e Librerie	44
JQUERY	45	
AJAX	45	
PHPMailer	46	
LeafLet	47	

	Utilizzo delle librerie	48
4.1.7	Cookies	48
4.1.8	Sicurezza	50
	Invio dati	50
	Criptaggio password	50
4.2	Accessibilità	52
4.2.1	WCAG	52
	Linee guida	54
4.3	Struttura del progetto	56
4.3.1	Pagina di login	58
4.3.2	Pagina di Sign Up	59
4.3.3	Pagina della privacy	60
4.3.4	Pagina principale	61
4.3.5	Pagina delle impostazioni	63
4.3.6	Pagina degli avvistamenti	65
5	Documentazione Applicazione Mobile	72
5.1	Tecnologie utilizzate	72
5.1.1	Kotlin	72
5.1.2	Jetpack Compose	73
5.1.3	Material Design 3	74
5.2	Schema di navigazione	75
5.3	Mockup	76
5.3.1	Login e Registrazione	77
5.3.2	Homepage	78
5.3.3	Avvistamento	79
5.3.4	Aggiungi avvistamento	80
5.3.5	Statistiche	81

5.3.6	Impostazioni	82
5.4	Librerie esterne utilizzate	83
5.5	Activity	84
5.5.1	Login	84
5.5.2	Registrazione	85
5.5.3	Profilo	86
5.5.4	Homepage	87
5.5.5	Aggiunta avvistamento	88
5.5.6	Visualizzazione avvistamento	90
5.5.7	Impostazioni	91
5.5.8	Notifiche	92
5.5.9	Statistiche	93
5.6	Accessibilità e sicurezza	94
5.7	Connessione ad internet	94
6	Riconoscimento della specie	95
6.1	Il problema	95
6.2	Modello	96
6.2.1	Identificazione della specie	96
6.2.2	MobileNetV2	97
6.3	Dataset	98
6.3.1	Specie presenti e quantità di immagini	99
6.4	Algoritmo di riconoscimento	100
6.4.1	Fase di training	101
	Librerie utilizzate	102
	Caricamento dei dati	104
	Controllo caricamento e distribuzione delle categorie	105
	Associazione immagine etichetta	106

Divisione del dataframe	108
Istanziamento dei generatori	109
Creazione generatore di immagini da un dataframe	111
Creazione generatore di immagini di test	113
Creazione modello pre-addestrato di MobileNetV2	113
Definizione di un nuovo modello	115
Compilazione modello di apprendimento automatico	116
Fase di training del modello	118
Rappresentazione delle curve di perdita	121
Rappresentazione delle curve di accuratezza	122
Significato delle rappresentazioni	123
Determinazione dell'accuratezza media	124
Matrice di confusione e report di classificazione . .	124
Visualizzazione della matrice di confusione	126
Visualizzazione della precisione per ogni specie . .	127
6.4.2 Fase di riconoscimento dell'immagine	129
Librerie utilizzate	129
Caricamento della rete neurale	130
Funzione di predizione	130
Predizione delle immagini	131
6.5 Integrazione algoritmo di riconoscimento	132
7 Test dell'applicazione	134
7.1 Test con l'utente	134
7.1.1 Verifica del software	135
Testing	135
7.1.2 Analisi	136
7.1.3 Test	137

Capitolo 1

Introduzione

Durante il mio percorso accademico, mi è stata proposta la creazione di una piattaforma per la gestione degli avvistamenti di particolari specie marine. L'obiettivo era quello di aiutare il Dipartimento di Acquacoltura e Igiene delle produzioni ittiche nel tracciamento degli avvistamenti delle diverse specie marine.

Ho intrapreso questo progetto al fine di mettere alla prova le conoscenze apprese durante il mio percorso universitario, arrivando a creare un'intera infrastruttura centralizzata.

Il sistema si suddivide in diverse parti, tra cui spiccano:

- Creazione di un database centralizzato che permetta di salvare tutte le informazioni ricavate da un avvistamento e di poterle modificare in un secondo momento.
- Creazione di applicazione smartphone che permetta di inserire i nuovi avvistamenti.

- Creazione di una applicazione web che permetta di gestire, modificare e visualizzare gli avvistamenti e i relativi dati.
- Utilizzo di script di riconoscimento che, attraverso la visione artificiale e diversi algoritmi noti, permetta di individuare la specie precisa di determinati esemplari acquatici o anche di riconoscere il singolo individuo per poterne eventualmente tracciare gli spostamenti.

1.1 Diagramma dei casi d'uso

Viene sotto riportato un diagramma dei casi d'uso, figura 1.1, il quale descrive le principali operazioni del sistema:

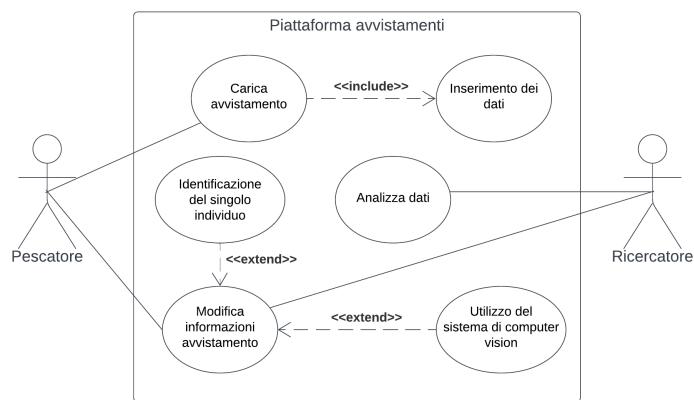


Figura 1.1: Schema dei casi d'uso per la piattaforma.

Capitolo 2

Documentazione Base di dati

2.1 Analisi dei requisiti

In questa sezione, analizzeremo tutte le specifiche che il committente ha richiesto per la piattaforma.

2.1.1 Intervista

Come punto di partenza, è bene riportare la prima intervista svolta col committente, poiché essa consente di individuare bene le caratteristiche essenziali necessarie perché il prodotto finale possa essere adatto alle esigenze dell'utilizzatore finale. "La facoltà di Acquacoltura e Igiene delle produzioni ittiche richiede un sistema informativo per la gestione di una banca dati al fine di permettere la registrazione di determinati avvistamenti fatti durante le uscite a largo della costa.

Ogni utente che avrà a disposizione uno smartphone, una volta effettuato l'accesso con login e password, deve avere la possibilità di inserire un nuovo avvistamento, nel momento stesso dell'avvistamento. Per ogni in-

serimento, gli elementi da memorizzare sono: data e ora di compilazione (inserite automaticamente dal sistema), numero di esemplari che sono stati avvistati e, opzionalmente, anche la velocità del vento (in km/h), le condizioni del mare ed eventuali ferite visibili sull'animale. In particolare, è necessario porre attenzione alla descrizione della ferita, alla sua posizione sul corpo dell'animale e alla sua gravità. Deve anche essere possibile scrivere note che possano aiutare ad individuare la specie e/o che contengano ulteriori informazioni sull'avvistamento.

Si deve inoltre tenere presente che un avvistamento può far riferimento ad un particolare animale, a una specie precisa o anche a un animale di cui non è riconoscibile né l'identità né la specie esatta. In quest'ultimo caso, il dato della specie deve risultare comunque inseribile in un secondo momento.

Oltre all'applicazione smartphone, deve essere realizzato un applicativo web che, una volta effettuato l'accesso, deve consentire all'utente di poter visualizzare tutti gli avvistamenti che sono stati effettuati da lui stesso o dagli altri utenti, potendone modificare i parametri. La modifica deve essere permessa per tutti i dati, eccetto gli elementi che riguardano l'utente e la data della rilevazione.

Ad ogni avvistamento, il personale addetto deve poter associare anche una o più immagini. Esse serviranno in seguito per il riconoscimento della specie di appartenenza e/o dei singoli individui. Dato che in un avvistamento possono essere presenti anche più animali, comunque tutti della stessa specie, la foto può essere suddivisa in più sottoimmagini in modo tale da poter isolare ogni singolo individuo.

La base di dati deve anche tenere in considerazione la possibilità di avvistamenti dello stesso individuo in momenti diversi. Nello specifico, le sottoimmagini possono essere riferite allo stesso individuo in avvistamenti diversi, in modo tale da consentire di monitorare i suoi cambiamenti nel

tempo e i suoi spostamenti.

Per aiutare l’utente nella decisione della specie da assegnare a un particolare animale, il sistema deve inoltre memorizzare una descrizione, in cui vengono fornite le informazioni base su ciascuna specie analizzabile.

Infine, l’applicativo web, attraverso l’uso di tecniche di visione artificiale, con particolare riferimento ai delfini, deve poter essere in grado di identificare in autonomia la specie dell’individuo attraverso la foto scattata durante il rilevamento. Deve cioè cercare di aiutare l’utente nella scelta della specie, stilando una classifica di compatibilità con le specie e gli individui che sono presenti in database.

Un aspetto importante da tenere in considerazione è il fatto che al largo della costa non è presente la connessione internet. Il sistema deve poter lo stesso mantenere i dati in memoria e caricali una volta che ha la possibilità di farlo.

2.1.2 Rilevamento delle ambiguità e correzioni proposte

Il testo dell’intervista presenta molte ambiguità che è necessario risolvere prima di procedere alla programmazione. Le principali sono:

- Utilizzo di sinonimi.
- Elenchi di attributi incompleti.
- Cardinalità non specificate.

Per quanto riguarda gli attributi parziali e le cardinalità, questi aspetti saranno corretti mediante l’uso della logica in fase di creazione dello schema concettuale. Invece per quanto concerne i sinonimi, è necessario costruire un glossario dei termini che saranno considerati al fine della progettazione concettuale come si può vedere dalla tabella 2.1:

Termine	Descrizione	Sinonimi	Collegamenti
Utente	Persona che accede al portale al fine di inserire un nuovo avvistamento oppure per modificare avvistamenti già presenti nella banca dati	Persona	Avvistamento
Avvistamento	Avvistamento di un esemplare, al fine di memorizzare informazioni	Rilevazione	Utente Animale Specie Immagine
Animale	Tipologia di animale avvistato	Specie marina	Avvistamento Specie
Specie	Specie dell'animale avvistato		Animale Avvistamento Descrizione
Descrizione	Descrizione specifica al fine di una migliore selezione della specie		Specie
Immagine	Immagine che rappresenta l'avvistamento	Foto	Avvistamento Sottoimmagine
Sottoimmagine	Sezioni di immagini dei singoli individui		Immagine Ferita Esemplare
Ferita	Ferita sul corpo di un animale, con una gravità		Sottoimmagine
Esemplare	Singolo animale che si monitora		Sottoimmagine

Tabella 2.1: Glossario dei termini.

2.1.3 Estrazione dei concetti principali

Dall'intervista si ricavano anche le operazioni principali richieste in fase di progettazione:

- Creazione di un nuovo utente.
- Registrazione di un nuovo avvistamento attraverso l'applicazione mobile.
- Inserimento di segnalazioni di ferite riconducibili al singolo animale con specifica della gravità.
- Possibilità di inserimento di una o più immagini dell'avvistamento.
- Possibilità di visualizzare, modificare ed eliminare un avvistamento.
- Possibilità di suddividere immagini in più parti ciascuna delle quali possa essere analizzata a seconda dell'animale attraverso un sistema di riconoscimento della specie.
- Monitoraggio degli avvistamenti di particolari individui al fine di poter compiere determinate verifiche.
- Possibilità di consultazione di informazioni relative alla specie che si vuole attribuire a un determinato animale, al fine di una selezione più accurata.
- Caricamento del nuovo avvistamento, comprensivo di dati e immagini (anche in un secondo momento, nel caso non sia presente connessione).

2.2 Progettazione Concettuale

La fase di progettazione concettuale consiste inizialmente nella formalizzazione dei requisiti, in particolare sui termini e sulle transazioni che sono stati raccolti e analizzati nella fase precedente.

2.2.1 Schema scheletro

Per maggiore chiarezza, di seguito si riportano i primi esempi di schemi concettuali divisi per ambiti.

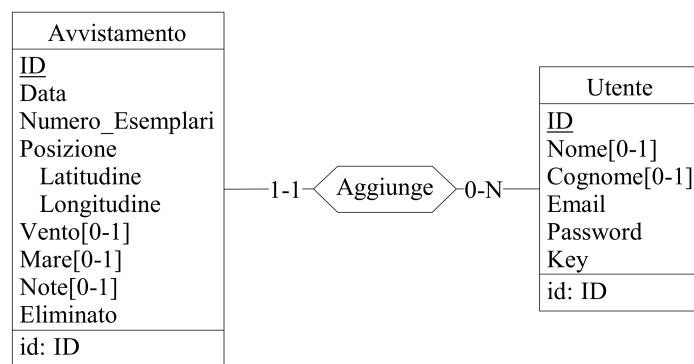


Figura 2.1: Schema concettuale per la modellazione degli avvistamenti e delle proprietà dell’utente.

Come mostrato in figura 2.1, sono presenti uno o più utenti, ognuno dei quali accede alla piattaforma attraverso la propria email, la propria password e grazie a una chiave personale univoca all’interno del database. Quest’ultima viene utilizzata al fine di criptare la password. Dell’utente, poi, possiamo tenere anche in considerazione il nome e cognome. Ogni utente può così aggiungere degli avvistamenti, ognuno dei quali contiene tutti i dati richiesti dal committente tra cui la data di avvistamento, il nume-

ro di esemplari presenti e, opzionalmente, anche il vento espresso in km/h, lo stato del mare ed eventuali note informative per finalità di ricerca e, soprattutto, per riuscire ad identificare meglio la specie.

In aggiunta, ad ogni avvistamento è associata una posizione che tiene in considerazione la longitudine e la latitudine registrate al momento dell'inserimento.

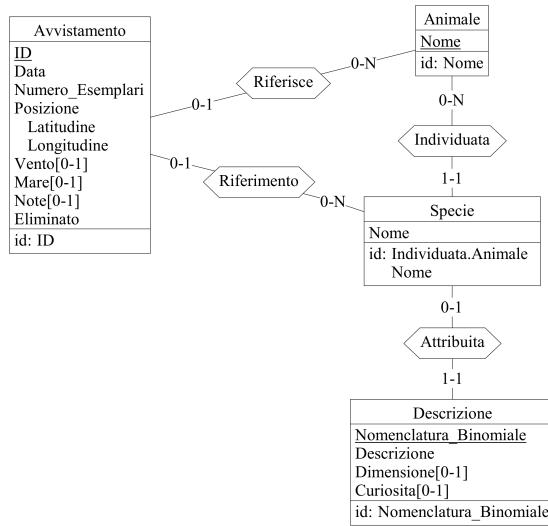


Figura 2.2: Modellazione dell’attribuzione dell’animale e della specie all’avvistamento.

Dalla figura 2.2 si può notare che un avvistamento può far riferimento a un animale (ad esempio un delfino oppure a un granchio), direttamente alla specie o anche a nessuno dei due. In quest’ultimo caso, tali dati verranno inseriti in un secondo momento quando saranno disponibili. Al fine di individuare al meglio la specie, ad ognuna di esse può essere attribuita una descrizione in cui vengono elencate le caratteristiche fondamentali. Ogni specie è identificata dal nome dell’animale e dal suo nome specifico. Come

si nota dalla figura, l'avvistamento può essere direttamente associato all'animale oppure alla specie, in questo caso nasce il vincolo per l'utente che, nel momento in cui dichiara una determinata specie, essa deve fare riferimento al giusto animale. Un ulteriore vincolo da rispettare è il fatto che in un avvistamento, anche se include più esemplari, i dati inseriti devono far riferimento allo stesso animale ed eventualmente alla relativa specie. Per segnalare la presenza di specie diverse, sarà quindi necessario inserire più avvistamenti.

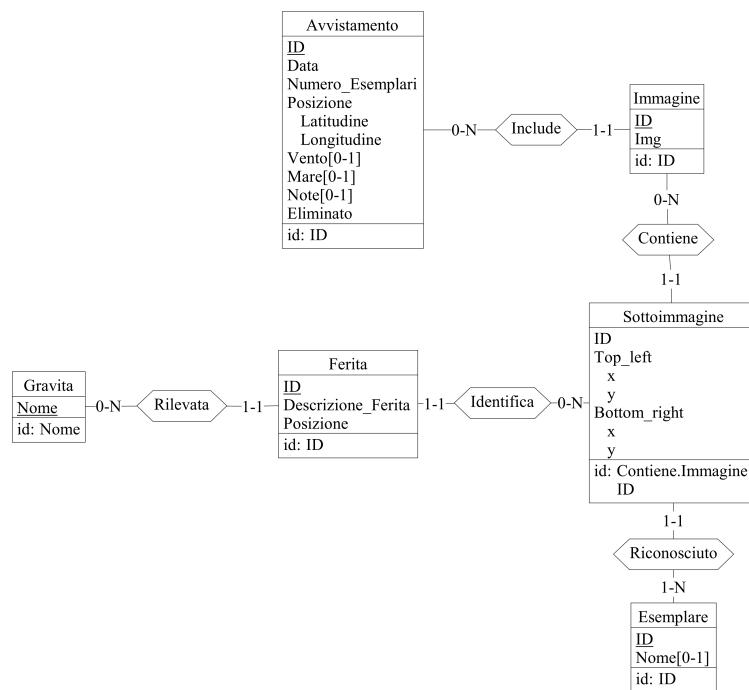


Figura 2.3: Schema concettuale raffigurante le immagini con relative sottimmagini e la gestione delle ferite.

Navigando lo schema 2.3 partendo dall'entità Avvistamento, si nota che per ogni avvistamento si possono includere una o più immagini che devono po-

ter essere suddivise in ulteriori sottoimmagini. Per fare questo, si è pensato di utilizzare le coordinate cartesiane al fine di risparmiare memoria e tempo di elaborazione. Visto che ogni sottoimmagine rappresenta un elemento singolo, se il fotogramma presenta delle ferite, queste possono essere inserite e ad ognuna attribuita una gravità. In aggiunta, il singolo individuo viene catalogato con un id e opzionalmente anche con un nome, in modo tale che se viene rincontrato in altri avvistamenti si abbia la possibilità di indicare che si tratta dello stesso esemplare.

2.2.2 Raffinamenti attuati

Elenchiamo in questa sezione i raffinamenti che sono stati attuati nei precedenti schemi al fine di modellare meglio entità indipendenti. In particolare:

- L'entità Descrizione è stata creata per non appesantire troppo l'entità Specie con attributi opzionali che, come si vede dallo schema, rappresentano una sezione a parte del dominio.

2.2.3 Schema concettuale finale

Di seguito si riporta lo schema concettuale finale, figura 2.4 , ottenuto unendo opportunamente gli schemi scheletro presentati in precedenza.

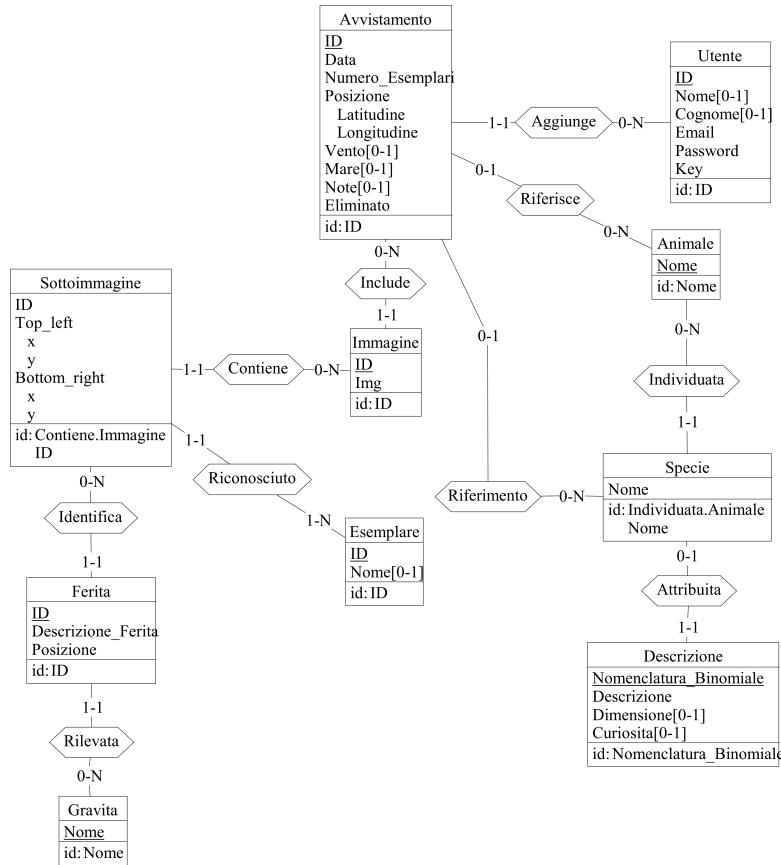


Figura 2.4: Schema concettuale finale

2.3 Progettazione Logica

La progettazione logica consiste nella traduzione dello schema concettuale finale in uno schema logico che rispecchi il modello relazionale. Lo schema logico progettato è indipendente dallo specifico Database management system.

ment system, che verrà scelto al termine della progettazione logica. Inoltre verranno definiti anche i vincoli di integrità sui dati.

2.3.1 Stima del volume dei dati

Nella tabella che segue 2.2, è riportato il volume atteso per ciascun costrutto presente nello schema concettuale. Si sottolinea che, per garantire maggiore compattezza, sono state omesse le stime dei volumi delle associazioni 1-N, in quanto equivalenti ai volumi delle entità che partecipano alle associazioni stesse, con cardinalità 1.

Concetto	Costrutto	Volume
Avvistamento	E	100
Utente	E	10
Immagine	E	200
Sottoimmagine	E	400
Ferita	E	50
Gravita	E	3
Esemplare	E	300
Animale	E	15
Specie	E	50
Descrizione	E	50

Tabella 2.2: Volume dei dati.

2.3.2 Descrizione delle operazioni principali e stima della loro frequenza

Di seguito si riporta la tabella 2.3 contenente la frequenza prevista e una descrizione delle principali operazioni, individuate già in fase di analisi.

Codice Operazione	Descrizione Operazione	Frequenza
1	Creazione di un nuovo utente	1/settimana
2	Monitoraggio di specifici individui	5/giorno
3	Registrazione di un nuovo avvistamento compreso di operazioni opzionali	10/giorno
4	Operazioni statistiche	3/giorno

Tabella 2.3: Operazioni principali.

2.3.3 Schemi di navigazione e tabelle degli accessi

Dopo aver stimato i volumi dei principali costrutti presenti nella base di dati e la frequenza delle principali operazioni, si può procedere a disegnare i relativi schemi di navigazione e a scrivere le tabelle degli accessi.

1. Creazione di un nuovo utente:

Per l'aggiunta dell'utente, tabella 2.4, omettiamo il corrispondente schema di navigazione, in quanto coincide con l'entità stessa:

Concetto	Costrutto	Accessi	Tipo
Utente	E	1	S
Totale = 1S			

Tabella 2.4: Tabella degli accessi per la creazione di un nuovo utente.

2. Monitoraggio di specifici individui per determinate verifiche 2.5:

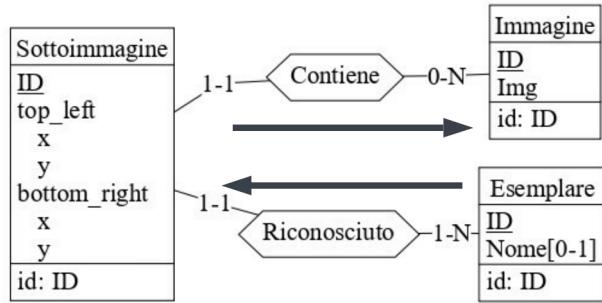


Figura 2.5: Schema concettuale raffigurante gli accessi alle tabelle per il monitoraggio di specifici individui.

Per svolgere questa operazione, come si può notare dalla tabella 2.5, occorre effettuare un totale di letture su **Sottoimmagine** pari al numero medio di foto a cui ogni esemplare è associato. In aggiunta, in base al numero di sottoimmagini a cui l'esemplare è associato, si dovrà selezionare lo stesso numero di immagini, visto che uno stesso esemplare non può comparire più volte nella stessa immagine.

Concetto	Costrutto	Accessi	Tipo
Esemplare	E	1	L
Sottoimmagine	E	$400 \div 200 = 2$	L
Immagine	E	2	L
Totale = 5L			

Tabella 2.5: Tabella degli accessi per il monitoraggio di individui marini.

3. Registrazione di un nuovo avvistamento compreso di operazioni opzionali:

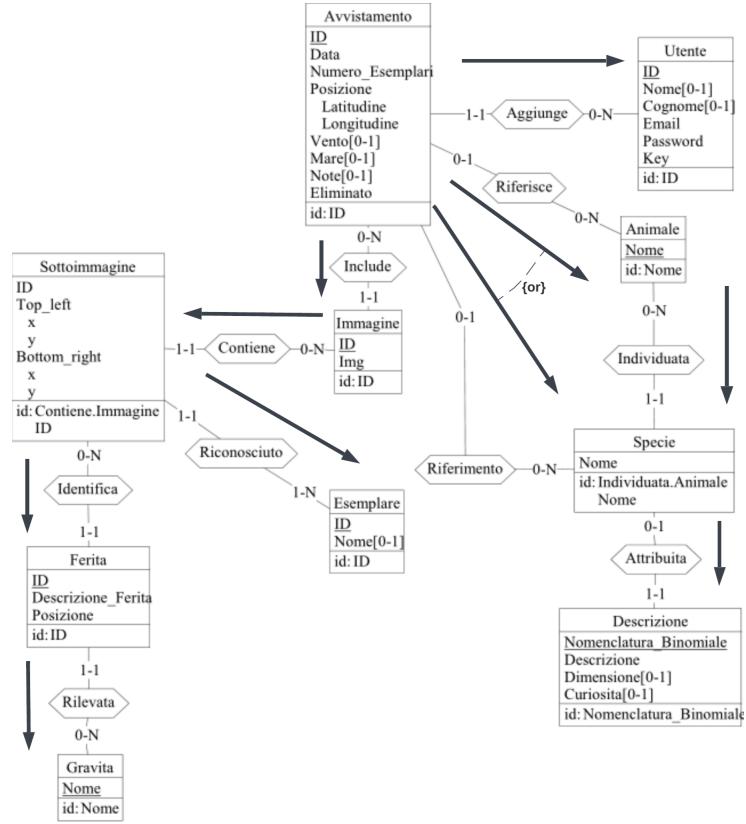


Figura 2.6: Schema concettuale raffigurante gli accessi alle tabelle per l'aggiunta di un nuovo avvistamento.

Lo schema di navigazione, figura 2.6, prevede di partire da Avvistamento, che legge l'utente, poi aggiunge più immagini che verranno suddivise in più sottoimmagini, ad ognuna delle quali verrà associato un esemplare. In più, ad ogni sottoimmagine si possono aggiungere delle indicazioni su possibili ferite, visibili in quella parte dell'immagine.

magine. Ad ogni ferita è obbligatorio associare una gravità. Nella tabella degli accessi sottostante 2.6 consideriamo il caso in cui l'esemplare e la coordinata geografica non siano ancora presenti nel database. Ogni avvistamento è associato ad un'animale e di conseguenza a una possibile specie. Al fine di individuare meglio la specie, essa è associata a una possibile descrizione.

Un'alternativa è la possibilità di scegliere direttamente la specie partendo dall'avvistamento, ma questo viene considerato un caso raro.

Concetto	Costrutto	Accessi	Tipo
Avvistamento	E	1	S
Utente	E	1	L
Animale	E	1	L
Specie	E	1	L
Descrizione	E	1	L
Immagine	E	$200 \div 100 = 2$	S
Sottoimmagine	E	$(400 \div 200) \times 2 = 4$	S
Ferita	E	$(50 \div 400) \times 4 = 0.5$	S
Gravita	E	1	L
Esemplare	E	4	S
		Totale = 11.5S + 5L	

Tabella 2.6: Tabella degli accessi per la registrazione di un nuovo avvistamento.

4. Operazioni statistiche:

Per semplicità omettiamo lo schema di navigazione. L'operazione statistica che si è deciso di rappresentare è quella della rappresentazione della distribuzione dei delfini. In particolare il numero di avvistamenti di ciascuna specie di delfino. Riportiamo la tabella 2.7 degli accessi.

Concetto	Costrutto	Accessi	Tipo
Avvistamento	E	100	L
Animale	E	100	L
Specie	E	$100 \div 15 = 6.66$	L
Totale = 206.66L			

Tabella 2.7: Tabella degli accessi per le operazioni statistiche.

Accessi totali

Si riporta di seguito la tabella 2.8 degli accessi totali per ogni operazione, considerando doppi gli accessi in scrittura:

Codice Operazione	Accessi	Frequenza	Totale
1	$1S = 2$	1/settimana	2/settimana
2	$5L = 5$	5/giorno	25/giorno
3	$11.5S + 5L = 28$	10/giorno	280/giorno
4	$206.66L = 206.66$	3/giorno	620/giorno

Tabella 2.8: Tabella degli accessi totali.

2.3.4 Raffinamento dello schema

In questa sezione ci occuperemo del raffinamento dello schema concettuale in vista della realizzazione dello schema logico. In particolare, bisognerà concentrarsi sulla seguente operazione:

- **Modifica degli attributi multipli e composti:**

Una volta individuate le entità che contengono attributi multipli o composti, si provvede a modificarli. In particolare:

- Nell’entità Sottoimmagine, gli attributi top_left e bottom_right formati da x e y vengono trasformati in singoli attributi.
- Nell’entità Avvistamento l’attributo posizione viene scomposto in due singoli attributi denominati Latitudine e Longitudine.

2.3.5 Traduzione di entità e associazioni in relazioni

La traduzione delle entità in relazioni è automatica e non richiede particolari passaggi.

Per quanto riguarda invece le associazioni, bisogna effettuare diversi passaggi in base alla cardinalità:

- Le associazioni 1-N vengono tradotte importando nell’entità che partecipa con cardinalità 1 la chiave dell’entità che partecipa con cardinalità N.
- Le associazioni 1-1 vengono tradotte importando nell’entità ritenuta più corretta, in base ad opzionalità e operatività, la chiave dell’altra entità.

L'operazione di traduzione porta alla creazione dello schema logico:

Gravita(Nome)

Ferite(ID, Descrizione_Ferita, Posizione, Gravi_Nome:Gravita,
(Sottoi_ID, Img_rif):Sottoimmagini)

Sottoimmagini(ID, tl_x, tl_y, br_x, br_y, (Immag_ID, Esemp_ID):Esemplari)

Esemplari(ID, Nome*)

Immagini(ID, Img, Avvis_ID:Avvistamenti)

Utenti(ID, Nome*, Cognome*, Email, Password, Key)

Animali(Nome)

Specie((Nome, Anima_Nome:Animali),

Nomenclatura_Binomiale*:Descrizioni)

Descrizioni(Nomenclatura_Binomiale, Descrizione, Dimensione*,
Curiosita*)

Avvistamenti(ID, Data, Numero_Esemplari, Vento*, Mare*, Note*,
Latid, Long, Utente_ID:Utenti, Anima_Nome*:Animali,
(Specie_anima_Nome, Specie_Nome)*:Specie, Eliminato)

2.3.6 Schema relazionale finale

Si riporta in figura 2.7 lo schema relazionale finale.

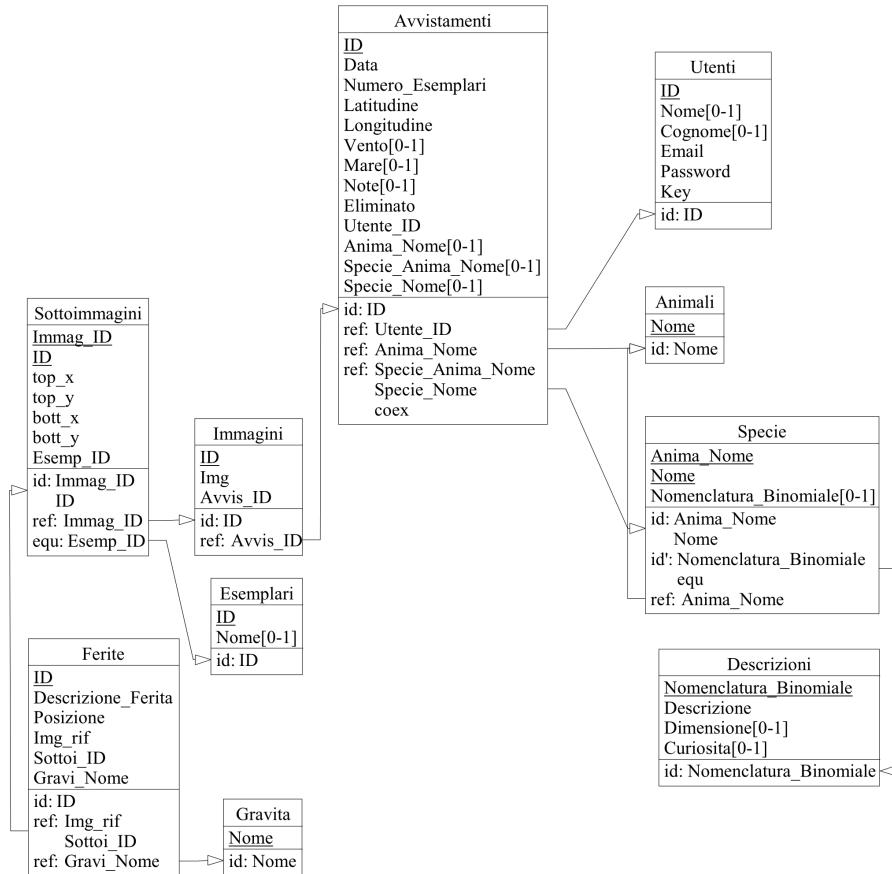


Figura 2.7: Schema logico finale.

Capitolo 3

Architettura del sistema

In questo capitolo si descrive come il sistema è strutturato, ossia come i vari tipi di utenti interagiscono col sistema e le parti che lo compongono.

3.1 Struttura

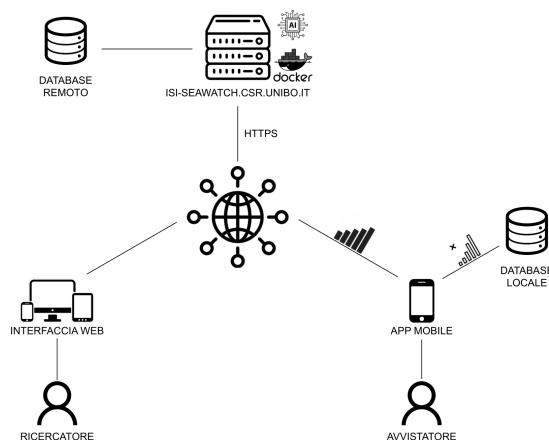


Figura 3.1: Schema di utilizzo della piattaforma.

Come è possibile vedere dall’immagine 3.1 esistono due tipologie di utenti: i ricercatori e gli avvistatori. I primi utilizzando l’interfaccia web hanno la possibilità di modificare i dati degli avvistamenti già caricati o aggiungerne dei nuovi. Essi possono anche inviare messaggi ad avvistatori o altri ricercatori al fine di contattarli e di ricevere ulteriori informazioni sul relativo avvistamento. Sul portale si ha la possibilità anche di selezionare singoli animali di un avvistamento e aggiungere nuove informazioni. Tutti i dati vengono salvati su un database remoto. Gli avvistatori invece utilizzano l’applicazione Android al fine di caricare nuovi avvistamenti, vedere le statistiche sugli avvistamenti e modificare i propri avvistamenti caricati in precedenza. Si noti che non sempre si ha la possibilità di essere collegati alla rete, per questo il dispositivo mobile salva tutti i dati di un relativo avvistamento in un database locale e solo quando è presente connessione si ha la possibilità di caricarli in rete. In entrambe le piattaforme si può utilizzare un algoritmo di visione artificiale al fine di riconoscere al meglio la specie.

Capitolo 4

Progettazione dell'interfaccia Web

4.1 Progettazione

Nella progettazione sia dell'applicativo web che mobile si è cercato di dare particolare attenzione alla comunicazione visiva della GUI, nello specifico si è cercato di rispettare i seguenti principi:

- **Affordance:** enfatizzare aspetti di un oggetto che invitano a manipolarlo in un certo modo.
Tali aspetti possono essere: tridimensionalità, ombreggiatura e punzettamento.
- **Metafora:** una parola, una frase o una figura che dipinge un oggetto o un concetto attraverso una somiglianza con un altro oggetto del mondo reale. Un esempio è un bottone che rappresenta un comando.

- **Layout:** la posizione degli elementi all'interno della pagina è uno strumento importante di comunicazione.
- **Colori:** utili per focalizzare l'attenzione o suscitare emozioni secondo la psicologia del colore.
- **Font:** leggibilità in relazione al tipo e alle caratteristiche del carattere.

4.1.1 Usabilità

Col termine usabilità si intende l'efficacia, l'efficienza e la soddisfazione con cui determinati utenti eseguono determinati compiti in particolari ambienti. Più nello specifico, in questo applicativo tali termini rappresentano:

- **Efficacia:** tutti i compiti richiesti in fase di analisi possono essere eseguiti senza problematiche.
- **Efficienza:** si è cercato di usare al meglio le risorse disponibili per svolgere i compiti richiesti.
- **Soddisfazione:** si è avuta prova dell'accettabilità del funzionamento da parte dell'utente.

Personas e scenarios

Sono state definite delle personas, cioè descrizioni di persone che rappresentano dei gruppi di utenti che saranno gli utilizzatori del programma ed a ognuno di loro è stato associato uno scenario che rappresenta le operazioni, il modo di muoversi nel sito e le loro modalità di utilizzo della piattaforma. Le personas che sono state identificate sono:

- Responsabile avvistamenti: persona addetta alla gestione delle informazioni al fine di modificare, aggiungere o eliminare i dati.
- Pescatore: persona che carica informazioni di un nuovo avvistamento direttamente dal portale.

A ciascuna di esse è associato uno scenario:

- Ricercare: iscrizione al sito con possibilità di modifica degli avvistamenti, aggiunta e cancellazione.

4.1.2 Design

In questa sezione viene descritto il processo che ha portato alla creazione dell’interfaccia web. Al fine di progettare la migliore interfaccia che rispetti le caratteristiche elencate in fase di analisi sono stati realizzati:

- Focus Group: è stato chiesto agli utilizzatori del sistema come si aspettavano la realizzazione di determinate parti dell’interfaccia, in particolare per le schede dei singoli avvistamenti sono state proposte diverse modalità di disposizione dei contenuti e, sulla base di un sondaggio ulteriore, è stata scelta la migliore strategia.
- Mockup: sulla base delle informazioni sono stati creati degli schemi base che danno un’idea della struttura del sistema.
- Experience Prototyping: è stato realizzato un semplice prototipo di tipo usa e getta al fine di verificare alcune parti incerte del programma, in particolare la tabella degli avvistamenti e le loro specifiche.

Per quanto riguarda il design, si è cercato di rispettare i seguenti principi:

- Mobile First
- User Centered
- Responsive
- Accessibile

Mockup

Prima dell'inizio della realizzazione effettiva delle pagine, sono stati creati i mockup principali al fine di definire la struttura delle pagine stesse. In particolare, sono stati creati i mockup per la pagina principale, in cui vengono visualizzati tutti gli avvistamenti, e anche per la pagina del singolo avvistamento, in cui si definiscono le informazioni chiave. Il primo mockup che viene mostrato è quello della pagina principale in modalità mobile:

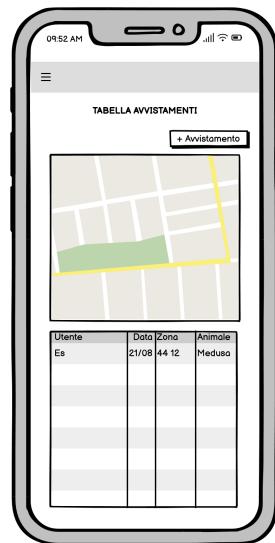


Figura 4.1: Mockup homepage del sito in versione mobile.

Come si può vedere dall’immagine 4.1, si tratta di un design molto semplice in cui sono presenti un menù, una mappa e una tabella in cui saranno elencati gli avvistamenti che sono caricati attraverso l’apposito pulsante oppure attraverso l’applicazione mobile.

Nel mockup di seguito 4.2 viene invece mostrata la stessa pagina web in versione desktop:

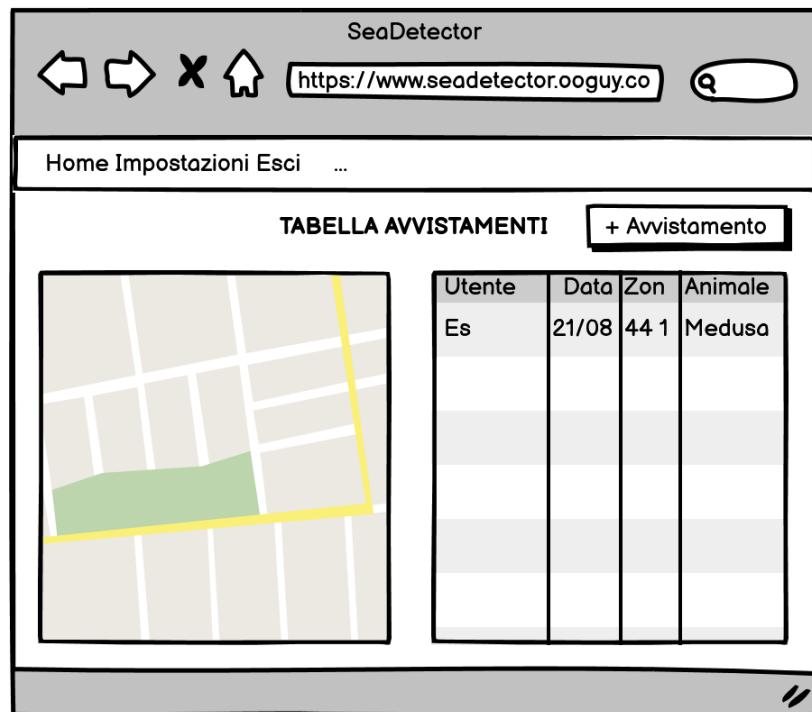


Figura 4.2: Mockup homepage del sito in versione desktop.

La pagina web contiene gli stessi elementi descritti nell’immagine precedente, ma disposti in modo differente al fine di ottimizzare lo spazio.

Nel terzo mockup 4.3, invece, viene rappresentata la struttura della pagina degli avvistamenti in versione mobile:

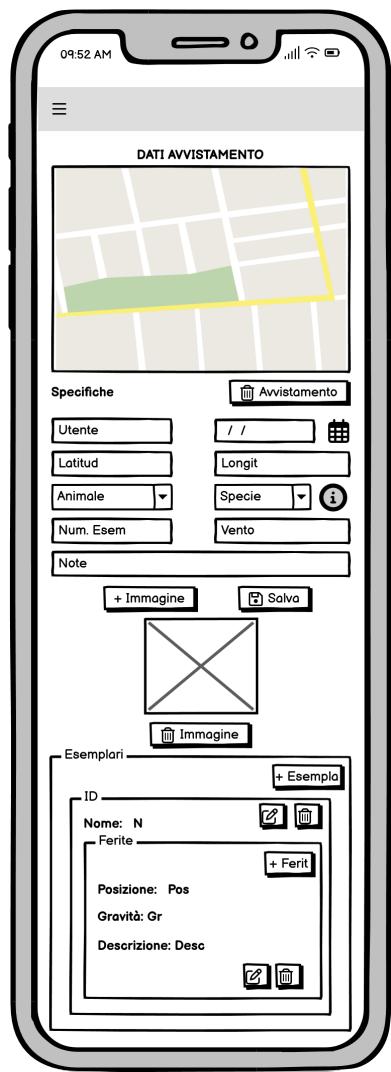


Figura 4.3: Mockup avvistamento del sito in versione mobile.

La pagina web contiene una semplice mappa in cui vengono raffigurati i luoghi dell'avvistamento e, per ognuno, vengono mostrati i relativi dati, che sono modificabili al fine di variare le informazioni precedentemente inserite. In aggiunta, per ogni avvistamento, c'è la possibilità di aggiungere immagini e di cancellarle. Per ogni immagine possono essere aggiunte più sottoimmagini, ognuna delle quali rappresenta un individuo al quale possono essere scritte o eliminate le indicazioni delle eventuali ferite.

Nel quarto ed ultimo mockup 4.4, viene raffigurata la struttura della pagina degli avvistamenti in versione desktop:

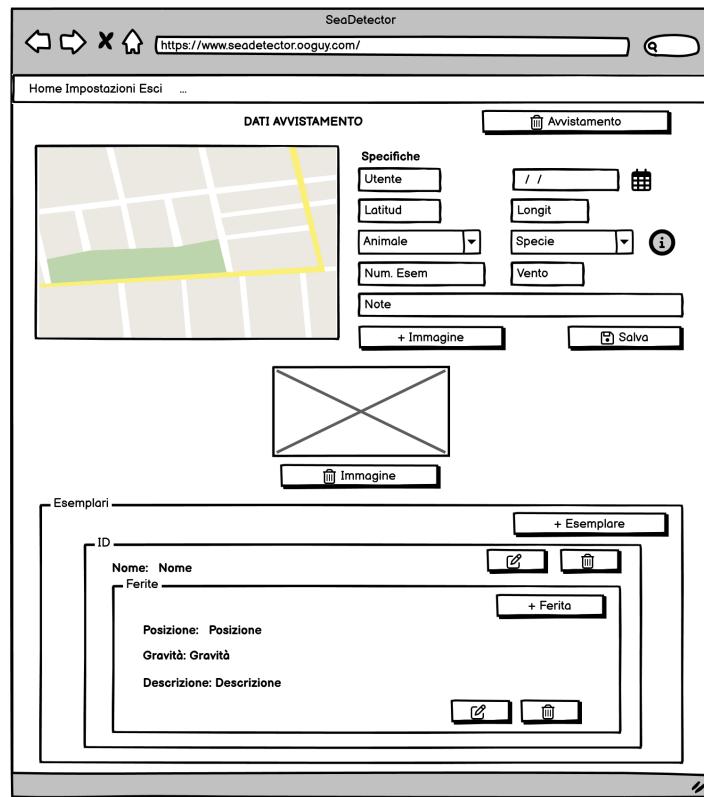


Figura 4.4: Mockup avvistamento del sito in versione desktop.

Il contenuto della pagina è lo stesso della versione mobile, solo con una disposizione diversa degli elementi.

4.1.3 Solution stack

Un solution stack è un insieme di tecnologie software che vengono utilizzate per costruire un'applicazione o un sistema. È composto da una combinazione di software di sviluppo, linguaggi di programmazione, framework, database e altri strumenti necessari per creare un'applicazione completa e funzionante. Un solution stack ben sviluppato può rendere molto più semplice e veloce lo sviluppo di un'applicazione, poiché fornisce tutti gli strumenti necessari in un pacchetto completo.

Esistono molti solution stack differenti, alcuni dei quali sono sviluppati per specifici sistemi operativi o piattaforme, mentre altri sono più generici e possono essere utilizzati su molte piattaforme diverse.

LAMP

Nel nostro caso, è stato utilizzato il solution stack denominato LAMP 4.5[1] (Linux, Apache [2], MySQL [3], PHP [4]), che è un popolare solution stack per lo sviluppo di applicazioni web su sistema operativo Linux. Questo perché il database e il server web Apache2 sono situati in un server Linux, che ha un indirizzo di rete fisso a cui chiunque sia collegato via internet può connettersi ed accedere al sito e ai suoi contenuti, ovviamente previa registrazione.

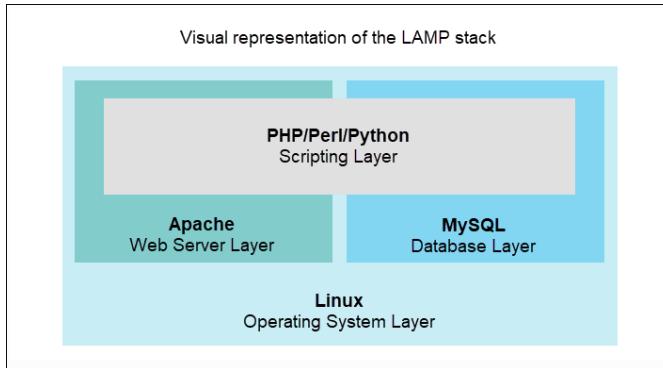


Figura 4.5: Stack LAMP, Fonte: [5]

PHP

Prendendo in considerazione il server, si ha un linguaggio di programmazione denominato PHP. Esso è open-source ed utilizzato per lo sviluppo di siti web dinamici e interattivi.

PHP è un linguaggio di scripting server-side, il che significa che il codice PHP viene eseguito sul server, non sul browser del client. Questo rende il processo di sviluppo più efficiente, poiché l'elaborazione avviene sul server e i dati vengono inviati al browser solo una volta elaborati. Perciò è possibile creare siti web dinamici e interattivi, che possono essere personalizzati in base alle esigenze dell'utente.

Esso supporta anche molte funzionalità avanzate, come l'elaborazione di form HTML [6], la connessione a database, l'elaborazione di cookie e sessioni e la gestione dei file. I siti web che vengono creati in questo modo possono offrire molte funzionalità avanzate, come ad esempio la personalizzazione dei contenuti in base alle preferenze dell'utente.

MYSQL

MySQL è un sistema di gestione del database relazionale open-source, utilizzato per gestire grandi quantità di informazioni. Il fatto che sia un sistema di gestione del database relazionale significa che i dati sono organizzati in tabelle e ogni tabella può avere relazioni con le altre. Questo rende il processo di gestione dei dati molto efficiente e semplice, poiché è possibile accedere rapidamente a informazioni specifiche utilizzando delle query.

La tecnologia descritta offre quindi un sistema di gestione del database altamente scalabile, il che significa che può gestire grandi quantità di informazioni senza compromettere le prestazioni. Inoltre, è molto flessibile e facile da usare, il che significa che gli sviluppatori possono facilmente creare e gestire grandi quantità di dati.

4.1.4 Fruizione del servizio

Il servizio, essendo di tipo centralizzato su un unico server, figura 4.6, è fruibile utilizzando le tre tecnologie che sono alla base del web.

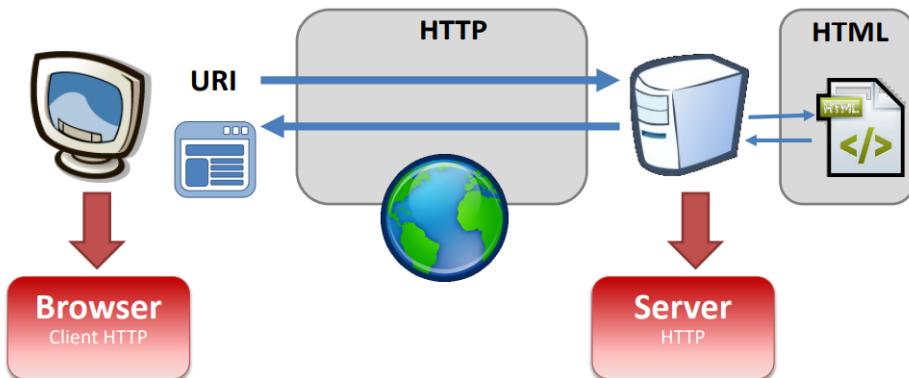


Figura 4.6: Tecnologie alla base del web.

URI

Lo Uniform Resource Identifier è utilizzato per identificare risorse in modo univoco e permanente. Questo è importante perché rende possibile accedere alle risorse su Internet in modo affidabile e coerente, indipendentemente dalla loro posizione. Nel nostro caso, si può utilizzare l'URL *isi-seawatch.csr.unibo.it* al fine di raggiungere il sito.

HTTPS

L'Hypertext Transfer Protocol Secure è un'estensione di HTTP, il protocollo di base utilizzato per la trasmissione di informazioni su internet. Il funzionamento di HTTPS si basa sulla crittografia. Quando un utente accede a un sito web che utilizza HTTPS, i dati inviati e ricevuti vengono crittografati utilizzando un certificato SSL (Secure Sockets Layer) o TLS (Transport Layer Security). Questo certificato garantisce che i dati siano protetti da eventuali attacchi esterni e che solo l'utente e il sito web abbiano accesso ai dati scambiati. L'utilizzo di HTTPS è molto importante per la sicurezza delle informazioni personali e dei dati sensibili degli utenti. Ad esempio, quando si effettua un login, i dati sensibili come le password vengono trasmessi attraverso internet. Se il sito web utilizza HTTPS, questi dati vengono protetti da eventuali attacchi esterni e non possono essere intercettati o utilizzati impropriamente.

Nel nostro caso, in particolare, una volta installato e configurato il server web, è stato utilizzato un software denominato Certbot [7] che facilita la creazione, l'installazione e l'utilizzo di certificati SSL per i siti web. Inoltre, Certbot si occupa automaticamente della gestione dei certificati, incluso il rinnovo automatico dei certificati prima della loro scadenza. Questo rende molto più semplice la gestione dei certificati SSL, poiché gli sviluppato-

ri non devono preoccuparsi di rinnovare manualmente i certificati. Certbot permette di ottenere certificati gratuiti tramite Let's Encrypt [8], un'organizzazione che fornisce certificati SSL a chiunque li richieda.

HTML

Una volta contattato il server attraverso l'identificatore e utilizzando il protocollo di file transfert in modalità sicura, il client e il server si scambiano le pagine in formato HTML.

HTML (Hypertext Markup Language) è un linguaggio di markup utilizzato per creare documenti web. HTML definisce la struttura e il contenuto della pagina web utilizzando un insieme di tag per identificare diversi elementi della pagina, come il testo, le immagini, i link e molto altro.

4.1.5 Stile

Per quanto riguarda lo stile del sito, sono state utilizzate diverse tecnologie combinate insieme per cercare di avere la massima resa possibile.

CSS

CSS (Cascading Style Sheets) [9] è un linguaggio utilizzato per descrivere l'aspetto di un documento HTML. Esso è un componente importante della creazione di siti web, poiché permette ai designer e agli sviluppatori di separare il contenuto di una pagina web dalla sua rappresentazione. In questo modo, il contenuto di una pagina web può essere modificato senza influire sulla sua presentazione, rendendo più semplice la manutenzione e

l’aggiornamento del sito. CSS viene utilizzato per la creazione di layout complessi, animazioni, transizioni ed effetti grafici.

Bootstrap

Bootstrap [10] è un framework di design per il web, che fornisce un insieme di componenti e strumenti per la creazione di pagine. Bootstrap è progettato per aiutare i designer e gli sviluppatori a creare progetti web più velocemente e con maggiore facilità, fornendo una serie di componenti predefiniti e stili per la creazione di interfacce utente. Esso è basato su HTML, CSS e JavaScript [11] e offre una vasta gamma di componenti, tra cui barre di navigazione, moduli, tabelle, pulsanti, icone, griglie e molto altro. Bootstrap rende inoltre molto più semplice la creazione di progetti web, senza dover scrivere da zero il codice CSS e HTML per ogni progetto. I designer e gli sviluppatori possono utilizzare i componenti predefiniti di Bootstrap, che sono già ben testati e ottimizzati per una buona esperienza utente. Un altro aspetto fondamentale è che esso è basato sul principio denominato Mobile first, che permette di creare pagine web adatte oltre che per PC anche per cellulari.

Grazie alla combinazione di righe e colonne che insieme formano una griglia, è possibile modificare agevolmente il layout della pagina in base alla grandezza del dispositivo che si utilizza.

4.1.6 Javascript e Librerie

Per l’applicazione è stato usato Javascript come linguaggio di programmazione lato client. Esso è dinamico, interpretato e viene utilizzato principalmente per creare interazioni ed effetti visivi sulle pagine web. Questo linguaggio di programmazione è completamente differente rispetto a HTML

e CSS, che sono utilizzati per la creazione della struttura e del design delle pagine web. JavaScript, d'altra parte, è utilizzato per creare interazioni e effetti dinamici all'interno della pagina, come ad esempio la validazione dei moduli e la creazione di animazioni.

JQUERY

Nel progetto, al fine di semplificare la creazione di determinate funzionalità, è stata utilizzata una libreria JavaScript open source così da rendere più semplice e veloce lo sviluppo di pagine interattive e applicazioni web. Essa è denominata JQUERY [12] e la sua principale caratteristica è la sua capacità di semplificare le attività di sviluppo web, come la manipolazione del DOM (Document Object Model), l'aggiunta di effetti e animazioni e la gestione degli eventi. JQUERY ha la caratteristica di essere totalmente compatibile con i browser più diffusi e di offrire un'ottima compatibilità anche con i dispositivi mobile.

AJAX

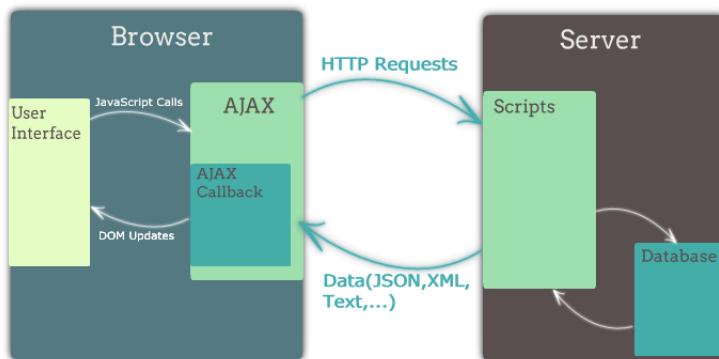


Figura 4.7: Funzionamento di AJAX, Fonte: [13]

Nell'applicazione, per richiedere dati dal database, vengono eseguite delle chiamate Ajax asincrone al server, figura 4.7. AJAX (Asynchronous JavaScript and XML) [14] è una tecnologia utilizzata per creare applicazioni web dinamiche e interattive. Essa si basa sull'utilizzo di JavaScript e permette di aggiornare parti di una pagina web senza doverla ricaricare. Invece di inviare una richiesta al server per ottenere l'intera pagina web, AJAX consente di inviare richieste al server per ottenere solo i dati necessari. Questo significa che, ad esempio, se un utente clicca su un link o modifica una form, solo i dati necessari vengono inviati al server, e solo la parte della pagina che deve essere aggiornata viene effettivamente modificata. Il processo rende l'interazione più veloce e fluida per l'utente, poiché non è necessario attendere il caricamento dell'intera pagina ogni volta che viene effettuata un'azione. Il server invierà i dati richiesti dal client in formato JSON [15].

PHPMailer

Nel sito, al fine di recuperare la password nel caso qualcuno la dimentichi, in fase di login, è stata implementata una funzione che permette di inviare email con all'interno la password temporanea dell'account. Per farlo è stata utilizzata la libreria PHP Mailer [16]. Essa è una libreria open-source per PHP che permette di inviare email tramite il protocollo SMTP (Simple Mail Transfer Protocol). Questa libreria è stata sviluppata per semplificare il processo di invio di email da un sito web e offre molte funzionalità avanzate, come ad esempio la gestione della codifica dei caratteri e l'invio di email con allegati. La libreria PHPMailer è molto semplice da utilizzare e offre un'interfaccia intuitiva per l'invio di email da un sito web.

LeafLet

Leaflet [17] è una libreria open-source per la visualizzazione di mappe interattive sul web. Leaflet consente di visualizzare mappe basate su tiles, marker, pop-up, forme e tanti altri elementi. La libreria supporta una grande quantità di formati di dati geografici e offre la possibilità di creare mappe personalizzate utilizzando fogli di stile. LeafLet è supportata da una vasta gamma di dispositivi, tra cui desktop, tablet e smartphone, il che la rende una soluzione versatile e scalabile. In particolare, è utile nella nostra piattaforma per evidenziare i punti in cui sono stati effettuati gli avvistamenti. Un esempio è riportato nell'immagine 4.8.

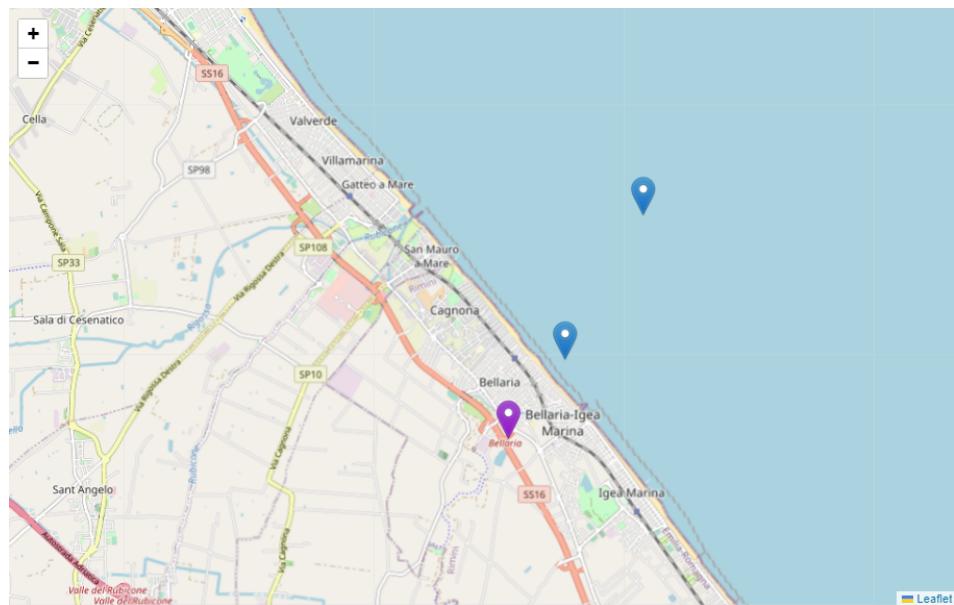


Figura 4.8: Mappa Leaflet interattiva.

Utilizzo delle librerie

L'utilizzo delle varie librerie nel progetto viene riportato in una tabella 4.1:

Libreria/Tool	Utilizzo
AJAX	Comunicazione asincrona col server remoto
PHPMailer	Recupero delle credenziali via mail
LeafLet	Visualizzazione di mappe con marker nei luoghi degli avvistamenti

Tabella 4.1: Tabella delle librerie utilizzate nella pagina web.

4.1.7 Cookies

Esistono diverse tecniche utilizzate per raccogliere i dati online e costituire, attraverso essi, un profilo dell'utente. Una di queste tecniche prevede l'utilizzo di cookies. Essi sono considerati delle stringhe di testo che il browser crea all'apertura di una pagina web sul computer dell'utente e che salvano dati dello stesso durante la navigazione di un sito web, agevolandone l'utilizzo. Dei possibili esempi possono essere:

- Memorizzare le preferenze linguistiche
- I dati di login

I dati vengono memorizzati per essere poi ritrasmessi ai medesimi siti alla visita successiva dello stesso utente. Mediante i cookies, è anche possibile monitorare la navigazione e raccogliere dati inerenti le abitudini e le scelte personali degli utenti, consentendo così la creazione di profili dettagliati dei visitatori delle pagine. Un esempio è la personalizzazione delle inserzioni pubblicitarie sul browser. Generalmente, un cookie contiene un attributo

che indica la durata di vita e un numero generato in modo casuale che consente il riconoscimento dell’utente.

Esiste una specifica normativa di derivazione europea che disciplina l’utilizzo dei cookies al fine di tutelare le persone da forme di profilazione definite occulte e di consentire loro un minimo controllo sulla circolazione dei dati inerenti la propria navigazione online.

Esistono tre tipologie di cookies:

1. **Cookies tecnici:** necessari per motivi tecnici e comportano una forma indispensabile (e molte volte temporanea) di memorizzazione di dati. Essi consentono la normale navigazione di un sito o la implementazione di un servizio, salvando solo le preferenze ed i criteri di navigazione di ogni utente
2. **Cookies di profilazione:** elementi che non sono tecnicamente necessari per il funzionamento del sito. Alcuni esempi sono i cookies di tracciamento, oppure i cookies che creano profili dell’utente per finalità pubblicitarie o che vengono utilizzati per finalità di marketing
3. **Cookies analitici:** consentono il monitoraggio dell’uso del sito da parte degli utenti e consentono il miglioramento del sito stesso. Talvolta anche i cookies analitici possono essere di terze parti

Nel nostro applicativo sono stati utilizzati solo cookies tecnici, al fine di memorizzare i dati di login. Secondo la normativa, i cookies tecnici possono essere usati anche senza chiedere il consenso dell’utente e questa è stata proprio la scelta fatta per lo sviluppo di questa applicazione. Si è dunque deciso di non utilizzare cookies di profilazione e nemmeno quelli analitici.

4.1.8 Sicurezza

Oltre all'utilizzo di un protocollo HTTPS, si è data una particolare attenzione alla sicurezza sia per quanto riguarda la trasmissione dei dati tecnici degli avvistamenti sia di quelli personali, come ad esempio le password.

Invio dati

Tutte le form tramite chiamate ajax inviano i dati in modalità POST. Il metodo "POST" è una delle due modalità principali per inviare dati tramite una richiesta HTTP. Quando si inviano dati utilizzando il metodo POST, essi sono inclusi nella corpo della richiesta HTTP, anziché nell'URL come con il metodo GET. Questo rende il metodo POST adatto per l'invio di dati sensibili o confidenziali, poiché questi non sono visibili nell'URL e non possono essere facilmente intercettati o modificati.

Criptaggio password

Per quanto riguarda le password, si è usata una tecnica leggermente complessa, figura 4.9:

1. In fase di registrazione l'utente inserisce due volte la password in modo da evitare errori di battitura
2. Quando conferma i dati inseriti, viene creata una chiave univoca basata sul timestamp del momento e criptata utilizzando una funzione hash SHA1 (SALT)
3. Una volta calcolata la chiave criptata, essa viene utilizzata come chiave di codifica per cifrare la password inserita dell'utente attraverso

l'algoritmo di crittografia a chiave simmetrica denominato HMAC-SHA512.

4. La password e la chiave così criptate, verranno salvate nel database insieme a tutti gli altri dati utente non criptati.
5. Una volta che l'utente esegue l'accesso, il client richiederà al server la chiave al fine di criptare la password e la invierà al server, a questo punto il server confronterà le due password e, in caso di corrispondenza perfetta, permetterà l'accesso al servizio.

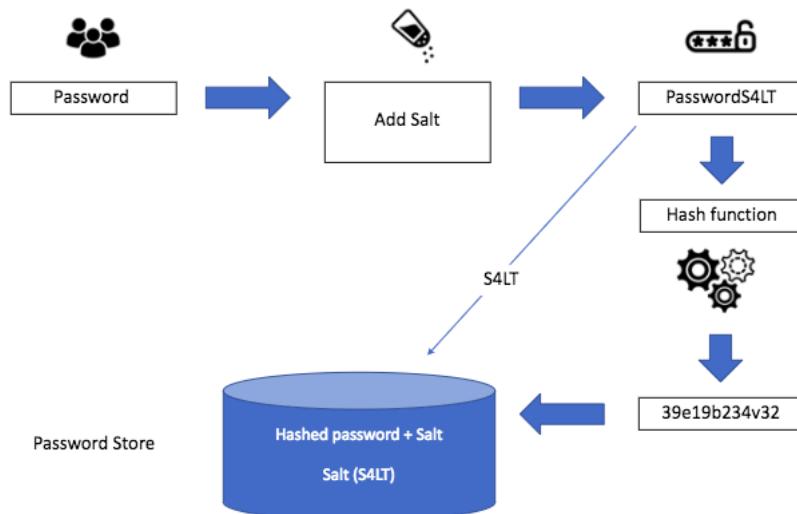


Figura 4.9: Struttura alla base della fase di criptaggio [18]

In questo modo la password dell'utente per l'accesso al sistema non verrà mai trasmessa in rete, permettendo maggiore sicurezza in caso di attacchi di tipo Sniffing, man in the middle o Spoofing.

4.2 Accessibilità

L'accessibilità informatica è la capacità per i sistemi informatici di fornire servizi e informazioni utilizzabili senza discriminazioni per tutti, compresi coloro che hanno bisogno di tecnologie assistive o configurazioni speciali a causa di disabilità. Questo viene realizzato nei limiti delle attuali conoscenze tecnologiche. Per quanto riguarda il sito, si sono presi in considerazione i seguenti tipi di disabilità:

- Visive
 - Daltonismo
 - Ipovisione (tool assistivi di ingrandimento)
 - Cecità (accesso attraverso screen reader e voice browser)
- Motorie (Utilizzo di strumenti di input diversi)

4.2.1 WCAG

L'intero sito internet segue le linee guida create dal W3C [19] per i diversi livelli di accessibilità, tali livelli sono racchiusi nell'immagine 4.10. In particolare, WAI è l'acronimo di Web Accessibility Initiative ed è un'iniziativa del W3C il cui obiettivo è di promuovere la creazione di contenuti web accessibili e di garantire che tutti gli utenti, indipendentemente dalla loro abilità, possano accedere e interagire con i contenuti del web in modo efficiente e significativo. Il WAI lavora in stretta collaborazione con organizzazioni governative, industriali ed enti di advocacy per garantire che le linee guida e le tecnologie sviluppate siano adatte alle esigenze di tutti gli utenti del web. Il suo impegno a supportare l'accessibilità del web ha contribuito a rendere questo ambiente un luogo più inclusivo e accessibile per

tutti gli utenti.

Le linee guida prese in considerazione sono le WCAG [20] che è l'acronimo di Web Content Accessibility Guidelines. Esse si basano su 4 principi, ovvero sulle caratteristiche necessarie perché un sito sia accessibile. Il sito deve risultare:

- Percepibile
- Utilizzabile
- Comprensibile
- Robusto

Dai 4 principi discendono le 12 linee guida che forniscono indicazioni per rendere il contenuto più accessibile. Per ciascuna di esse, sono stati identificati una serie di criteri di successo che possono essere facilmente verificati. Questi criteri sono suddivisi in tre livelli di conformità: A, AA e AAA (massimo).

Principles	Guidelines	Level A	Level AA	Level AAA
1. Perceivable	1.1 Text Alternatives	1.1.1		
	1.2 Time-based Media	1.2.1 – 1.2.3	1.2.4 – 1.2.5	1.2.6 – 1.2.9
	1.3 Adaptable	1.3.1 – 1.3.3		
	1.4 Distinguishable	1.4.1 – 1.4.2	1.4.3 – 1.4.5	1.4.6 – 1.4.9
2. Operable	2.1 Keyboard Accessible	2.1.1 – 2.1.2		2.1.3
	2.2 Enough Time	2.2.1 – 2.2.2		2.2.3 – 2.2.5
	2.3 Seizures	2.3.1		2.3.2
	2.4 Navigable	2.4.1 – 2.4.4	2.4.5 – 2.4.7	2.4.8 – 2.4.10
3. Understandable	3.1 Readable	3.1.1	3.1.2	3.1.3 – 3.1.6
	3.2 Predictable	3.2.1 – 3.2.2	3.2.3 – 3.2.4	3.2.5
	3.3 Input Assistance	3.3.1 – 3.3.2	3.3.3 – 3.3.4	3.3.5 – 3.3.6
4. Robust	4.1 Compatible	4.1.1 – 4.1.2		

Figura 4.10: Struttura delle linee guida coi rispettivi livelli, Fonte: [21]

Il WCAG è stato sviluppato con la collaborazione di una vasta comunità di esperti in accessibilità, tecnologia e advocacy e viene continuamente aggiornato per tener conto delle evoluzioni delle tecnologie e delle esigenze degli utenti. Esso è diventato un punto di riferimento per gli sviluppatori di siti web e per le organizzazioni che cercano di garantire l'accessibilità dei loro contenuti web.

Linee guida

Nel progetto, le linee guida a cui si è data maggiore importanza sono state:
Principio Percettibile:

- Linea guida 1.1 Alternative testuali: è necessario fornire alternative testuali per qualsiasi contenuto non di testo, in modo che questo possa essere trasformato in altre forme fruibili secondo le necessità degli utenti come Braille, stampa a caratteri ingranditi, sintesi vocale o in un linguaggio più semplice.
 - Criterio di successo 1.1.1 Contenuti non testuali: tutti i contenuti non testuali presentati all'utente hanno un'alternativa testuale equivalente. Questo viene fatto con l'utilizzo in modo opportuno dei tag figure, figcaption oppure con gli attributi alt e longdesc nei tag img.
- Linea guida 1.4 Distinguibile: rendere più semplice agli utenti la visione dei contenuti, separando i contenuti in primo piano dallo sfondo.
 - Criterio di successo 1.4.3 Contrasto (minimo): la rappresentazione visiva del testo e di immagini contenenti testo ha un rapporto di contrasto di almeno 4.5:1.

- Linea guida 1.3 Adattabile: creare contenuti che possano essere rappresentati in modalità differenti (ad esempio, con layout più semplici), senza perdere informazioni o struttura.
 - Questa linea guida tratta molte componenti che sono fondamentali nella costruzione della struttura di un documento, in particolare gli elementi della categoria Sectioning che hanno funzione strutturale. La struttura del documento è importante non solo per supportare la navigazione con l'ausilio di uno screen reader, ma anche per ottimizzare l'indicizzazione da parte dei motori di ricerca (SEO). La struttura del documento permette una navigazione più fluida tra sezioni, titoli e link.

In particolare, gli screen reader permettono all'utente non vedente di percorrere le pagine attraverso i tag di heading. Per questo motivo, i tag non devono assolutamente avere uno scopo presentazionale, ma la loro sequenza deve essere corretta al fine di aiutare gli utenti che utilizzano mezzi assistivi.

Un altro aspetto da tenere in considerazione sono le tabelle che sono presenti nel sito web. Queste, per essere accessibili, devono utilizzare l'attributo id nei tag di heading (th) e l'attributo headers nei tag (td) al fine di permettere l'associazione di ogni singola cella con il rispettivo titolo. In aggiunta, nei tag di intestazione, deve essere utilizzato l'attributo scope, che permette di identificare lo scopo della cella presa in considerazione. Per finire, al fine di far capire all'utente quali dati tale tabella rappresenti, è fondamentale utilizzare il tag caption, dove viene scritta la didascalia della tabella.

Principio Comprensibile:

- – Linea guida 3.3 Assistenza nell'inserimento: aiutare gli utenti ad evitare gli errori ed agevolarli nella loro correzione. In particolare, fornire etichette o istruzioni quando il contenuto richiede azioni di input da parte dell'utente.

La soluzione è l'utilizzo di label con attributo for e input con attributo id, oppure inserire il tag input all'interno della label.

4.3 Struttura del progetto

La struttura del progetto è concepita come un insieme di componenti distinti che vengono realizzati separatamente e successivamente integrati per creare il sistema finale. Come è visibile dall'immagine 4.11.

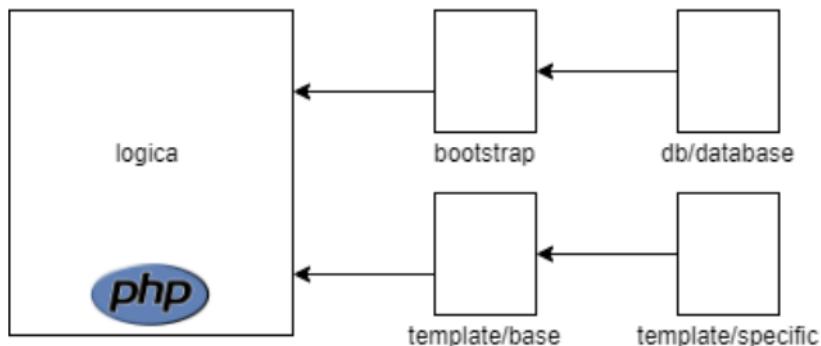


Figura 4.11: Struttura del progetto.

- La base della pagina web è un file che al suo interno contiene tutti i riferimenti a file di stile esterni, file Javascript comuni a tutte le pagine e riferimenti a librerie.

- Ogni pagina è composta da più componenti, ed ognuno è composto a sua volta da un template al cui interno è presente solo il codice html con la struttura di quella sezione, un file javascript che permette di eseguire chiamate asincrone a un terzo file php che, a seconda dell'operazione richiesta, richiede un'interrogazione al database e restituisce i dati richiesti in formato JSON.
- Al fine di eseguire le interrogazioni al database, esiste un file Bootstrap nel quale viene creata una connessione al DB. Tutte le funzioni di interrogazione sono racchiuse nel file denominato database che è utilizzato da tutti i file PHP citati prima.
- Esistono anche delle funzioni di utilità sia in PHP che in JS, che sono file separati e sono utilizzabili da tutti i file, col fine di non avere del codice ridondante o troppo pesante.

Un esempio:

- Per JS le funzioni di criptaggio, riempimento delle select, creazione di alert.
- Per PHP funzioni di invio mail, conversione date, orario, registrazione delle sessioni dell'utente.

4.3.1 Pagina di login

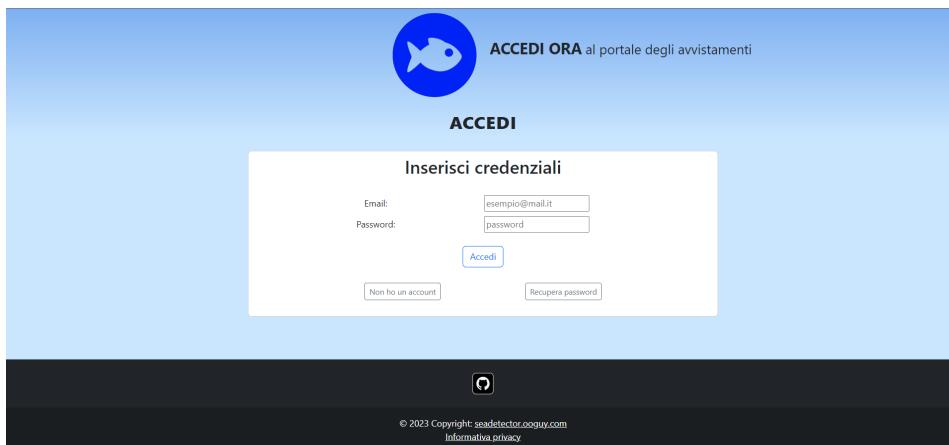


Figura 4.12: Struttura della pagina di Login.

La pagina, figura 4.12, è composta da due sezioni: la principale è la sezione di inserimento dati, dove l'utente inserisce le proprie informazioni per accedere al portale. Ci sono altri due pulsanti oltre a quello di accesso: uno per la pagina di registrazione, se l'utente non ha un account, e l'altro per il recupero della password in caso di dimenticanza. Al click, appare un modale da cui, inserendo l'email, vedi 4.13, l'utente riceverà le informazioni necessarie per accedere al portale, rispettando le regole di sicurezza.

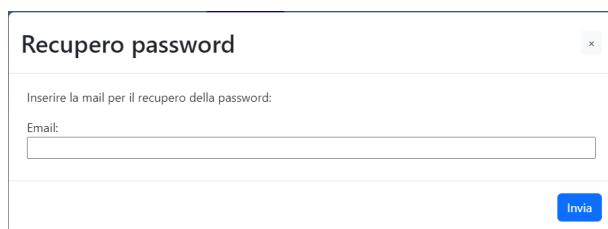


Figura 4.13: Modale per il recupero della password.

Il secondo componente, presente in tutte le pagine dell'applicazione, è il footer che contiene un link al repository di Github e un link alla sezione dedicata alla privacy.

4.3.2 Pagina di Sign Up

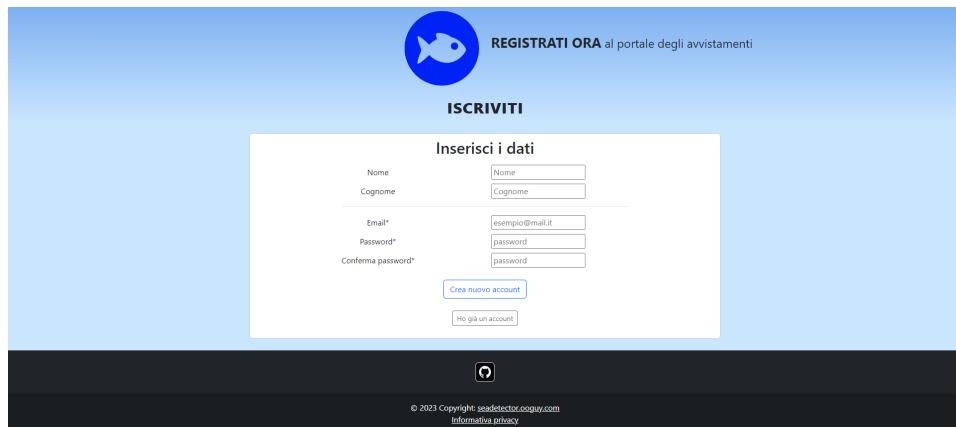


Figura 4.14: Struttura della pagina di Sign-Up.

La pagina, vedi figura 4.14, è composta da una form che contiene diversi campi obbligatori e alcuni facoltativi. Durante la registrazione, l'utente inserisce i propri dati che vengono successivamente convalidati e registrati nel database. La pagina presenta uno sfondo con un gradiente lineare che crea un forte contrasto con lo sfondo bianco della form, per garantire una maggiore accessibilità come descritto dalle specifiche del WCAG.

4.3.3 Pagina della privacy



Figura 4.15: Struttura della pagina dell’informatica sulla privacy.

In questa pagina, figura 4.15, sono elencate tutte le politiche sulla privacy. Ogni sezione presenta un elenco numerato che descrive i provvedimenti attuati per migliorare la protezione dei dati.

Questa è l’unica pagina che dispone di due menù di navigazione differenti. Il primo è visibile a tutti gli utenti che non hanno effettuato l’accesso al portale, con i soli link per il login e la registrazione. Il secondo è invece disponibile per gli utenti loggati e permette di navigare all’interno del sito.

4.3.4 Pagina principale

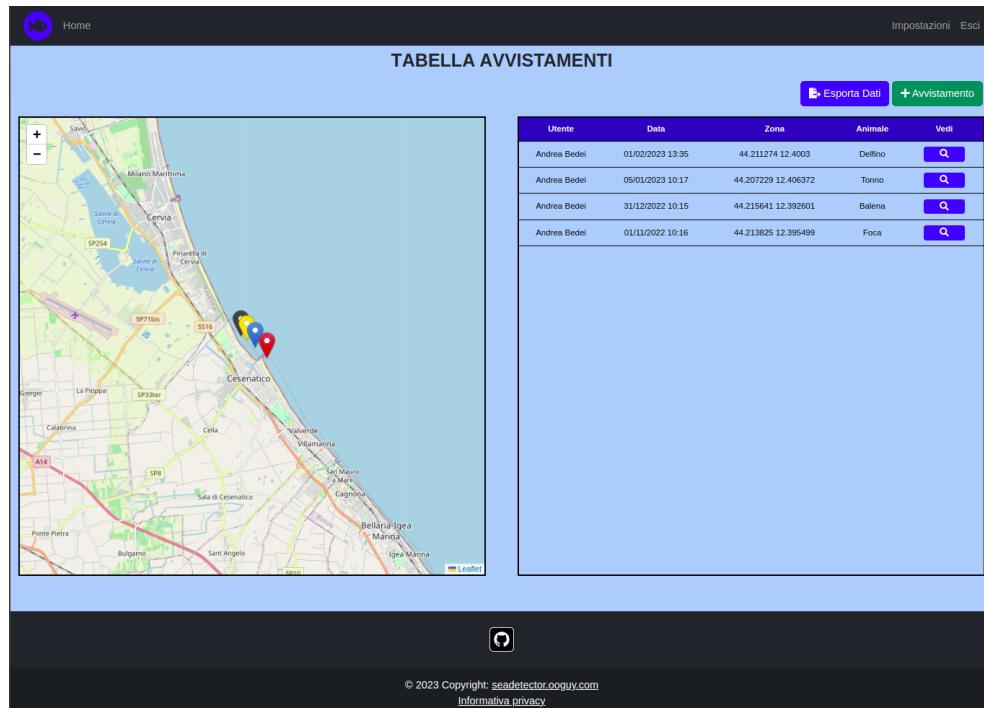


Figura 4.16: Struttura della pagina principale del sito.

La struttura della pagina, immagine 4.16, è molto semplice ed è composta da un menù di navigazione che viene condiviso tra tutte le pagine interne del sito. Questo menù permette agli utenti di navigare facilmente il sito e di effettuare il logout.

La parte principale della pagina è divisa in due sezioni costituite da una mappa e una tabella. La mappa mostra tutti gli avvistamenti segnati con dei marker colorati in base all'animale che rappresentano. Gli utenti possono interagire con la mappa spostandosi all'interno della sezione e visualizzando le informazioni principali sugli avvistamenti attraverso i popup che

appaiono quando si passa il mouse sui marker. Se uno di essi viene cliccato, verrà aperta la pagina dell'avvistamento specifico.

La sezione di destra presenta invece una tabella che include tutti gli avvistamenti con le informazioni più importanti. Questa tabella è stata realizzata seguendo tutte le direttive sull'accessibilità. Cliccando sul bottone "Vedi", si aprirà la pagina relativa all'avvistamento specifico. Sia la mappa che la tabella cambiano aspetto quando si passa il mouse sopra di esse, per motivi di stile.

In alto a destra è presente un pulsante che permette di aggiungere un nuovo avvistamento. Questo viene fatto attraverso un modale, figura 4.17, che appare in sovrapposizione quando si clicca sul pulsante.

Aggiungi avvistamento

Data*: <input type="text" value="gg/mm/aaaa --:--"/>	Num. esemplari*: <input type="text"/>
Latitudine*: <input type="text"/>	Longitudine*: <input type="text"/>
Animale: <input type="text" value="Sconosciuto"/>	Specie: <input type="text"/>
Mare (nodi): <input type="text"/>	Vento (km/h): <input type="text"/>
Note: <input type="text"/>	

Aggiungi **Annulla**

Figura 4.17: Modale per l'inserimento di un nuovo avvistamento.

Il modale è composto da una form in cui ci sono 4 campi obbligatori al fine dell'aggiunta dell'avvistamento e dei dati opzionali che potranno essere completati o modificati anche in un secondo momento. Per quanto riguarda l'animale e la specie sono presenti due menù a cascata in cui verranno vi-

sualizzati i nomi dei relativi elenchi estrapolati dal database. È necessario innanzitutto selezionare l'animale, e solo dopo si abiliterà il menù a discesa della specie al fine di poter scegliere la specie corretta dell'individuo. Infine, è possibile anche estrarre i dati di tutti gli avvistamenti presenti nella tabella e anche quelli eliminati attraverso il pulsante esporta, che avvia il download di un file .csv, leggibile col programma excel.

4.3.5 Pagina delle impostazioni

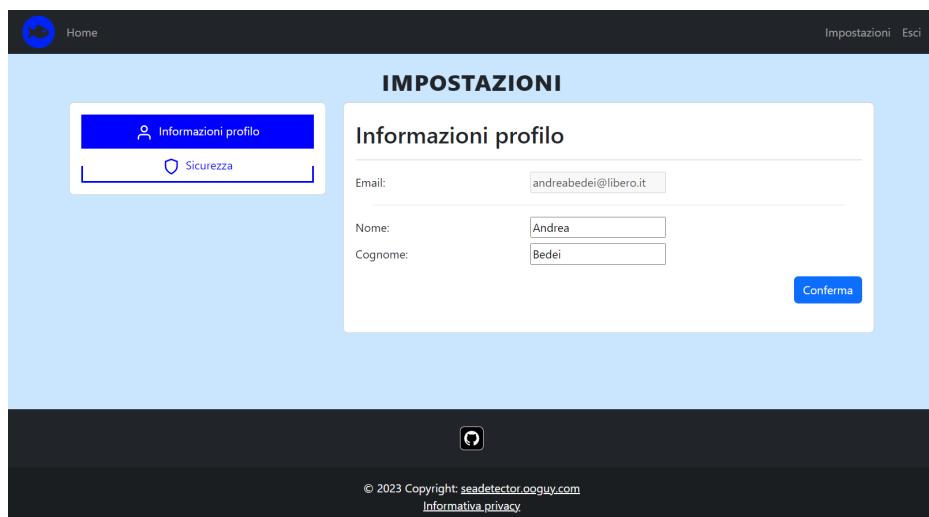


Figura 4.18: Pagina delle impostazioni del profilo utente.

Questa pagina, vedi 4.18, presenta un menù leggermente diverso rispetto agli altri, che cambia aspetto in base alle dimensioni dello schermo. Se viene utilizzato uno schermo più piccolo, il menù viene integrato nella sezione di destra. La sezione comprende una form, dove l'utente può modificare le proprie informazioni inserite durante il login, ad eccezione dell'indirizzo email, la cui modifica è disabilitata per ragioni di sicurezza. Se invece si

entra nella sezione sicurezza, si ha la possibilità di modificare la password. In primo luogo, verrà chiesta la vecchia password, poi la nuova e una conferma ulteriore della nuova password. Tutto questo è pensato per evitare problemi legati alla sicurezza ed errori di battitura.

Per di più, in tutte le pagine del sito, al fine di migliorare l'accessibilità e la facilità di utilizzo, sono stati aggiunti attributi "placeholder" ai tag di input, in modo da indicare all'utente le informazioni richieste. Inoltre, ai pulsanti il cui scopo non è chiaramente indicato è stato aggiunto l'attributo "aria-label" per specificarne lo scopo.

Infine, viene mostrata come viene modificata la pagina in modalità tablet o smartphone, nell'immagine 4.19.

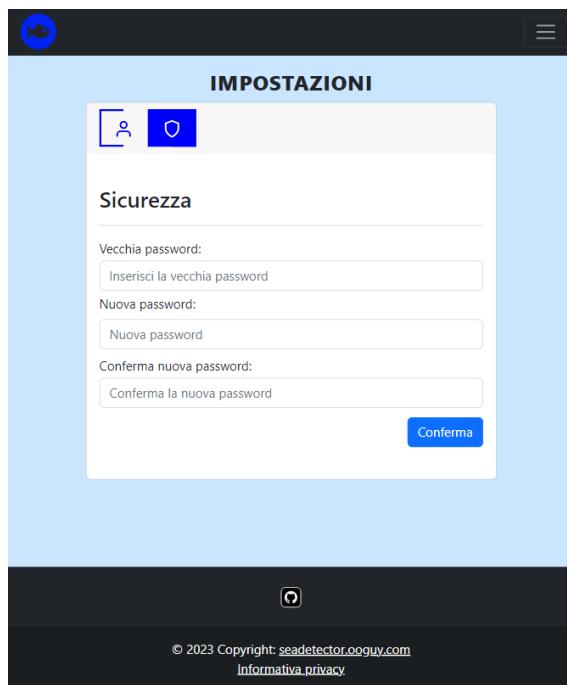


Figura 4.19: Pagina delle impostazioni sulla sicurezza in modalità mobile.

4.3.6 Pagina degli avvistamenti

La pagina degli avvistamenti, figura 4.20, consente di modificare e aggiungere informazioni sugli avvistamenti precedentemente caricati. Offre anche la possibilità di soddisfare tutti i requisiti specificati dal committente, per questo verrà descritta in dettaglio.

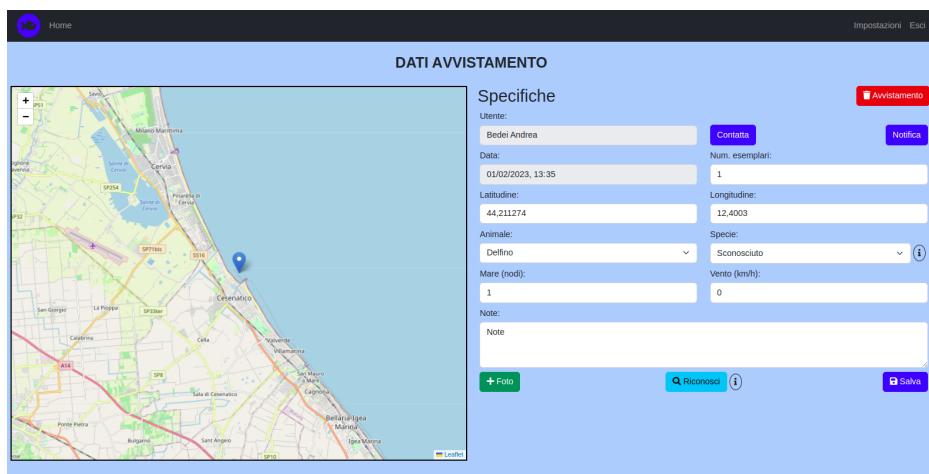


Figura 4.20: Prima parte della pagina dell'avvistamento, modifica dei dati.

In questa prima parte, ci sono due sezioni: la prima comprende una mappa nella quale viene indicato il punto dell'avvistamento attraverso un marker, col quale se l'utente clicca o ci passa sopra col mouse compaiono i dati principali dell'avvistamento. Nella parte di destra, invece, c'è la possibilità di modificare i dati inseriti in precedenza. Per quanto riguarda le interazioni, in alto a sinistra il bottone rosso permette di eliminare la registrazione dell'avvistamento. Esso non elimina definitivamente i dati, ma li mantiene nel database e li rende non visibili agli utenti, in questo modo, in caso di eliminazioni non volute, c'è la possibilità di recuperarli. Vicino all'indicazione della specie, è presente un pulsante a forma di punto interrogativo

che aiuta l’utente a individuare la specie, fornendo informazioni dettagliate sull’elemento scelto, figura 4.22. Il pulsante ”salva”, invece, permette di salvare le informazioni inserite negli input, mentre il bottone ”+Immagine” permette di aggiungere un’immagine all’avvistamento. Per evitare errori involontari, viene sempre visualizzato un modale di conferma prima di eliminare un avvistamento, figura 4.21. Sono presenti anche due bottoni: ”contatta” permetterà di contattare l’utente che ha eseguito l’avvistamento e ”notifica” invece permetterà di inviare una notifica all’utente qualora qualcuno avesse modificato i dati del suo avvistamento. Infine, è presente un bottone ”riconoscimento” che permetterà di ricevere dei suggerimenti sulla specie attraverso un modale. In aiuto all’utente, di fianco ad esso è presente un pulsante che spiega all’utente stesso le modalità di utilizzo.

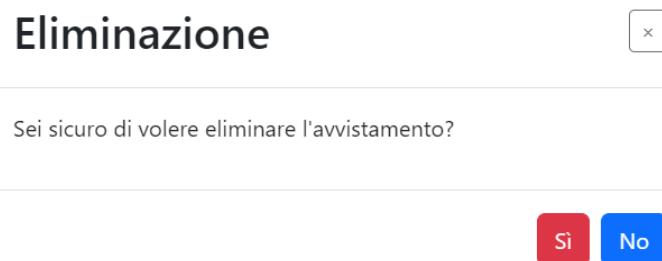


Figura 4.21: Modale di eliminazione.

Riconoscimento esemplari



Dalle immagini, la visione artificiale ha riscontrato che si potrebbe trattare di:

tursiope

OK

Figura 4.22: Modale per l'auto-riconoscimento della specie.

Viene mostrato un esempio del modale che si apre premendo il pulsante di informazione sulla specie, figura 4.23:

Informazioni



Nomenclatura Binomiale

Tursiops truncatus

Dimensione

Circa 3m

Descrizione

Il tursiope comune o tursiope dell'Atlantico è un mammifero marino ad ampio raggio della famiglia Delphinidae. Il tursiope comune è un delfino molto familiare per l'ampia esposizione che ottiene in cattività nei parchi marini e nei delfinari, e nei film e nei programmi televisivi.

Curiosità

Essi comunicano tra loro modulando dei fischi.

OK

Figura 4.23: Modale di informazione sulla specie.

Per quanto riguarda la seconda parte della pagina, immagine 4.24, essa è incentrata unicamente sulle immagini caricate.

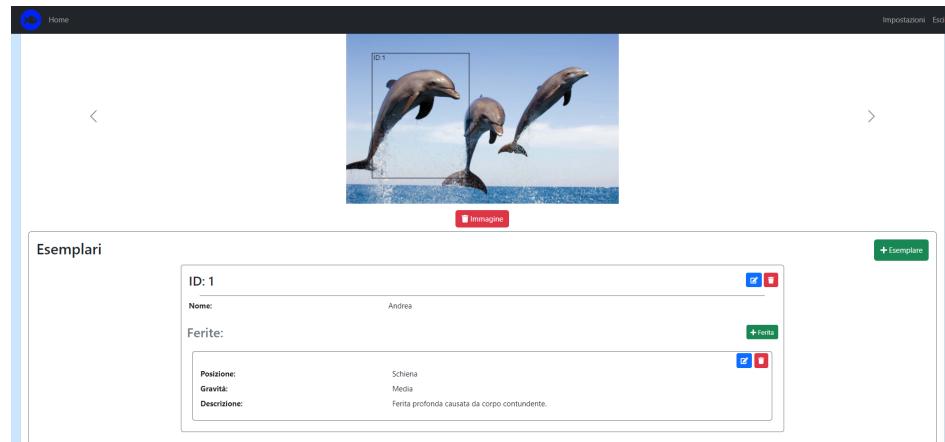


Figura 4.24: Seconda parte della pagina dell'avvistamento con caratteristiche degli esemplari.

Nella pagina è presente un carosello che permette di visualizzare tutte le foto dell'avvistamento. In caso siano state caricate più immagini, a destra e a sinistra della foto compariranno due bottoni per navigare la galleria. E' anche presente sotto all'immagine un bottone "elimina" che eliminerà non solo la foto, ma anche tutti i riferimenti ad essa associati. Per ogni immagine è presente la possibilità di aggiungere un esemplare attraverso il relativo bottone che, una volta premuto, permetterà attraverso un modale di disegnare dei rettangoli per selezionare l'individuo. Tali rettangoli vengono chiamati sottoimmagini e vengono salvati nella banca dati attraverso le coordinate dei vertici. Affinché il sistema sia portatile nei vari dispositivi, tutte le coordinate devono essere convertite da assolute in relative e, ogni volta che si richiede l'utilizzo, esse devono essere ripristinate. L'id dell'individuo, cioè della sottoimmagine, è univoco all'interno della stessa foto.

A ciascun individuo si ha la possibilità di associare un nome oppure, se si riconosce che si tratta di un esemplare già incontrato, si ha la possibilità di selezionarlo dai salvataggi precedenti. Come negli altri casi, anche qui è presente la funzione di eliminazione dei dati di un esemplare.

A ciascuna sottoimmagine sono associate una o più indicazioni ferite, che possono essere aggiunte, modificate o cancellate a seconda delle condizioni dell'animale. Un esempio di aggiunta di un individuo può essere il seguente, figura 4.25:

Aggiunta



Seleziona la parte di immagine che rappresenta un esemplare

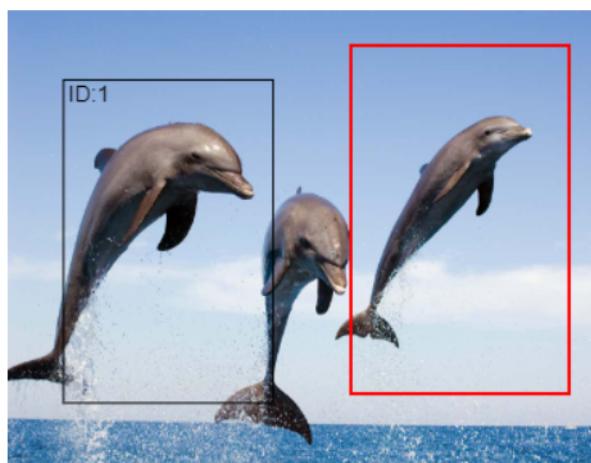


Figura 4.25: Modale con canvas per disegnare l'individuo.

Invece per quanto riguarda la modifica dell'individuo si ha il seguente modello:

Modifica

Nome corrente: **Andrea**

Scegliere l'esemplare:

1-Andrea

Nuovo Esemplare

Salva

Figura 4.26: Modifica dell'esemplare.

Come si vede dall'immagine 4.26, si hanno due possibilità: o attraverso una select viene scelto un esemplare già visto in precedenza oppure, cliccando sul pulsante "nuovo esemplare", si ha la possibilità di specificare il nuovo nome e di inserirlo nel database. Oltre al nome, nel menù a tendina si ha anche l'id univoco dell'esemplare, perché due esemplari possono avere lo stesso nome.

Per quanto riguarda le segnalazioni di ferite, c'è la possibilità di aggiungerle o modificarle attraverso questo semplice modale, figura 4.27:

The screenshot shows a modal window with the title "Aggiungi Ferita" at the top left. At the top right is a small square button with a white "X". Below the title is a horizontal line. The first section contains the label "Posizione:" followed by a light blue input field containing the text "Schiena". To the right of the input field is a small gray "X" icon. The second section contains the label "Gravità:" followed by a dropdown menu with the option "Lieve" selected. To the right of the dropdown is a small gray downward arrow icon. The third section contains the label "Descrizione:" followed by a text area containing the text "Ferita profonda causata da corpo contundente.". At the bottom right of the modal is a green rectangular button with the white text "Inserisci".

Figura 4.27: Modale per l'aggiunta e modifica della ferita.

Capitolo 5

Documentazione Applicazione Mobile

Si è deciso di realizzare l'applicazione mobile per il sistema operativo Android [22]. In particolare, l'applicazione è supportata da tutte le versioni dalla 8.0 a quella più recente.

5.1 Tecnologie utilizzate

L'app mobile è stata creata utilizzando un set di strumenti moderni tra cui: Kotlin [23], Jetpack Compose [24] e Material3 [25]. Grazie ad essi si è potuto scrivere codice UI in modo semplice e dichiarativo, ottenendo un'interfaccia utente reattiva con le linee guida di Material3.

5.1.1 Kotlin

Kotlin è definito come un linguaggio di programmazione ad alto livello, funzionale e orientato agli oggetti, sviluppato principalmente da JetBrains.

È stato progettato per essere interoperabile con Java [26], il che significa che gli sviluppatori possono utilizzare nel loro codice librerie Java esistenti. In aggiunta, Kotlin offre molte nuove funzionalità, ad esempio la null safety e una sintassi più concisa e leggibile rispetto agli altri linguaggi. Per null safety si intende la sicurezza rispetto al valore nullo: è un concetto che riguarda i linguaggi di programmazione e la gestione dei valori nulli o non definiti. L'obiettivo è di prevenire errori comuni legati alla gestione dei valori nulli che possono portare a crash del programma o a comportamenti imprevisti.

Esso è stato creato per semplificare la scrittura di codice e ridurre gli errori comuni, come la NullPointerException. Inoltre, offre espressività e flessibilità rispetto a Java, il che lo rende una scelta popolare per gli sviluppatori Android.

5.1.2 Jetpack Compose

Jetpack Compose è un framework UI per lo sviluppo di applicazioni Android. Compose permette agli sviluppatori di creare codice UI in modo dichiarativo e reattivo, semplificando la creazione di interfacce utente belle e performanti.

Il framework è basato sul linguaggio di programmazione Kotlin e offre un insieme di componenti UI predefiniti e personalizzabili, chiamati composable. Essi possono essere combinati tra loro per creare interfacce utente complesse.

Inoltre, Jetpack Compose permette agli sviluppatori di scrivere codice UI con un livello di astrazione più alto. Un esempio: gli sviluppatori possono

scrivere il codice per definire l’aspetto visivo dell’applicazione in un unico file, senza dover gestire la separazione tra codice XML e codice Kotlin. Grazie a queste funzionalità, il framework semplifica il processo di sviluppo e manutenzione, offrendo una soluzione moderna e performante per la creazione di interfacce utente reattive e coerenti con le linee guida di Material Design.

5.1.3 Material Design 3

Material Design 3 è l’ultima versione della libreria di design sviluppata da Google per creare interfacce utente su piattaforme diverse, tra cui Android, iOS e il web.

La libreria si concentra sulla proposta di elementi tangibili, cioè elementi di design che sembrano essere fatti di carta, inchiostro e superfici. Essa utilizza colori vivaci, forme geometriche chiare e transizioni fluide per creare un’esperienza utente coinvolgente e intuitiva.

Material Design 3 offre una vasta gamma di componenti UI predefiniti e personalizzabili, ad esempio: buttoni, icone, card e modali per aiutare gli sviluppatori a creare interfacce utente di alta qualità.

Inoltre, Material Design 3 fornisce un sistema di tipografia flessibile, che consente di personalizzare l’aspetto dei testi dell’interfaccia utente.

Il risultato finale è visto come un’interfaccia utente moderna, pulita e coerente, che migliora la user-experience e favorisce l’usability dell’applicazione.

5.2 Schema di navigazione

In questa sezione si riporta lo schema di navigazione dell'applicazione. Nello schema 5.1 è possibile vedere come avviene la navigazione dell'ap-

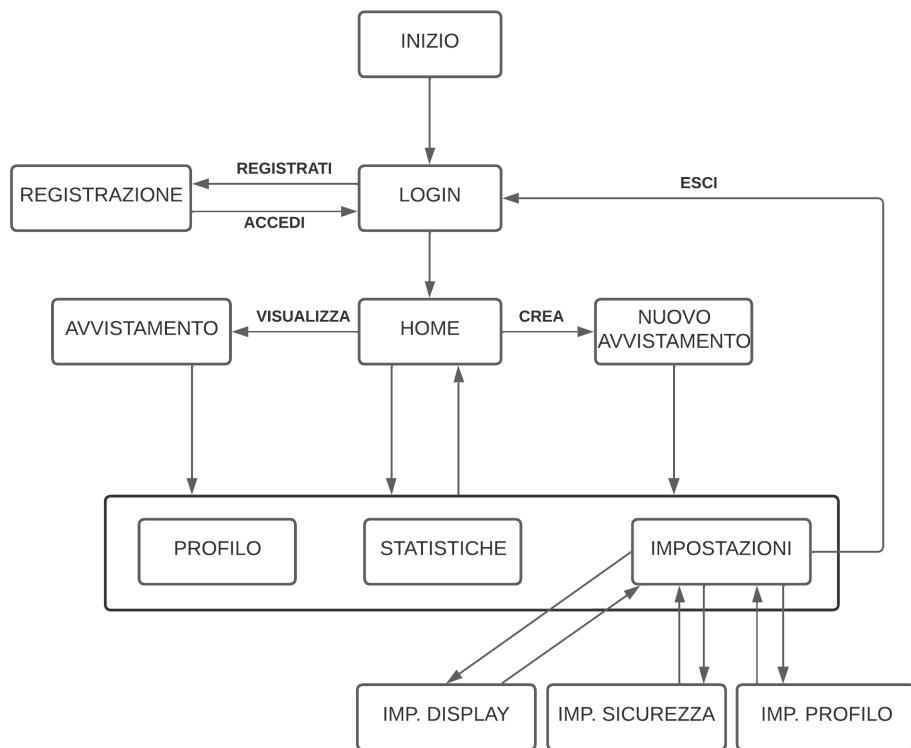


Figura 5.1: Schema di navigazione dell'applicazione mobile.

plicazione mobile. In particolare si inizia sempre dalla pagina di login, dalla quale o si effettua l'accesso o si passa all'activity di registrazione. Una volta effettuato l'accesso si entra nella Homepage del sito, dalla quale si ha accesso a tutte le altre schermate dell'applicazione. In particolare si può:

- Creare un nuovo avvistamento
- Visualizzare vecchi avvistamenti
- Entrare nella pagina delle statistiche
- Visualizzare il profilo utente
- Entrare in una delle tre pagine delle impostazioni, dalle quali si può uscire dall'applicazione e ritornare nella pagina di login

5.3 Mockup

Il primo passo nella realizzazione dell'applicazione è stato quello della creazione dei mockup. Per disegnarli, si è utilizzato il software di progettazione Balsamiq. Sono stati realizzati i modelli e si è richiesta l'approvazione del committente. Di seguito si riportano i principali prototipi di interfacce utente.

5.3.1 Login e Registrazione

Nella fase di registrazione, l'utente deve inserire i dati obbligatori ed eventuali dati facoltativi. Una volta registrato, si potrà effettuare l'accesso mediante credenziali oppure mediante riconoscimento delle impronte digitali. Vedi figura 5.2 e 5.3.

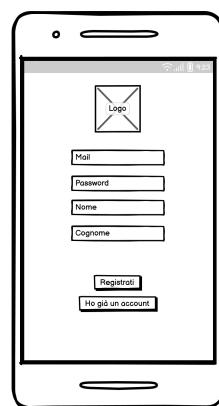


Figura 5.2: Prototipo activity di registrazione.

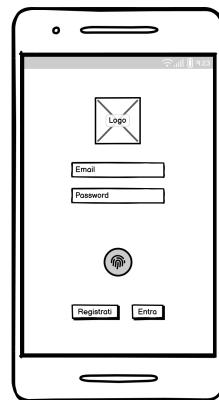


Figura 5.3: Prototipo activity di login.

5.3.2 Homepage

La schermata principale dell'applicativo, figura 5.4, si compone di una mappa in cui vengono raffigurati tutti gli avvistamenti con dei marker di colori diversi a seconda dell'animale che è stato individuato. A ciascun avvistamento corrisponde una card, nella quale sono presenti i dati più importanti della rilevazione, la foto dell'utente che l'ha eseguita e la possibilità di mettere nei preferiti l'avvistamento. In aggiunta, nella successiva schermata sono stati inseriti due menù: uno superiore e uno inferiore. Nel primo sono presenti il nome della pagina e la possibilità di filtrare i risultati per preferiti oppure per il tipo di animale avvistato. Invece il secondo permette la navigazione all'interno dell'applicazione (nello specifico: nella schermata delle impostazioni, statistiche e profilo) o tornare alla homepage. Infine, nella pagina è presente un FAB col simbolo del + per aggiungere un nuovo avvistamento.

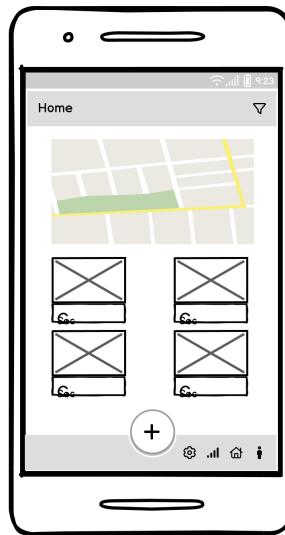


Figura 5.4: Prototipo activity Homepage.

5.3.3 Avvistamento

Per quanto riguarda l'avvistamento, figura 5.5, si avrà la mappa con la posizione dell'avvistamento segnalata da un marker, tutte le informazioni e immagini relative all'avvistamento saranno modificabili solamente dall'utente che ha eseguito l'avvistamento, per tutti gli altri saranno visibili in sola lettura. È data la possibilità di avere maggiori informazioni sul tipo di specie a cui è riferito l'avvistamento grazie a un popup che comparirà in rilievo se l'utente preme il pulsante informazioni. Infine nel menù superiore è presente un pulsante per tornare alla schermata precedente.

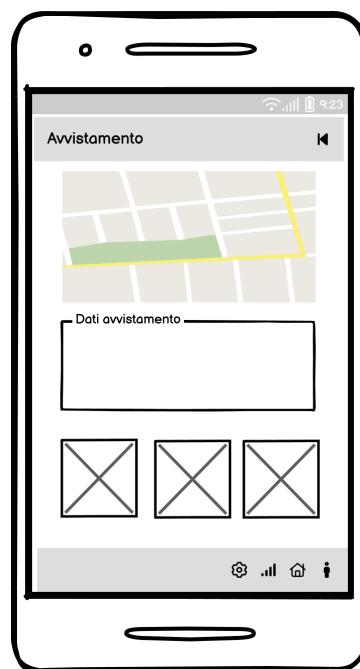


Figura 5.5: Prototipo activity di visualizzazione dell'avvistamento per altri utenti non creatori.

5.3.4 Aggiungi avvistamento

Questa activity, immagine 5.6, è uguale a quella della visione dell'avvistamento per chi ha creato l'avvistamento. Nella successiva schermata, sono presenti i campi per inserire i dati dell'avvistamento e un bottone per caricare foto fatte sul momento. In aggiunta, premendo il pulsante per le coordinate, esse vengono prese automaticamente e rese visibili all'utente utilizzando il GPS del dispositivo. Una volta completati i campi, in particolare quelli obbligatori, l'utente può caricare l'avvistamento attraverso l'apposito FAB situato a fianco.



Figura 5.6: Prototipo activity di aggiunta di un avvistamento.

5.3.5 Statistiche

Nell'applicativo mobile si è deciso di aggiungere un'ulteriore funzionalità rispetto all'applicativo web, cioè una Activity per le statistiche. Nella seguente pagina, figura 5.7, si vedono diverse statistiche, con relativi grafici degli avvistamenti, in particolare il numero degli avvistamenti, il numero di avvistamenti per ciascun animale o specie e il numero di avvistamenti in un determinato lasso temporale.

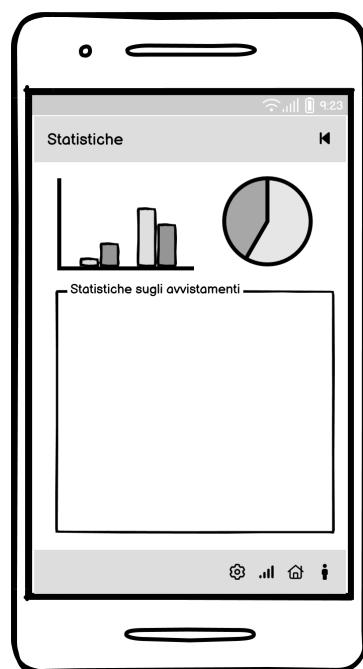


Figura 5.7: Prototipo activity di statistiche della piattaforma.

5.3.6 Impostazioni

Nella pagina delle impostazioni, figura 5.8, si possono modificare le informazioni del proprio profilo, modificare la password e anche decidere se il tema dell'applicazione deve essere chiaro o scuro, 5.9: tale preferenza viene salvata e memorizzata dall'app. Infine, nella barra superiore, compare il pulsante per effettuare il Logout o tornare alla schermata precedente.

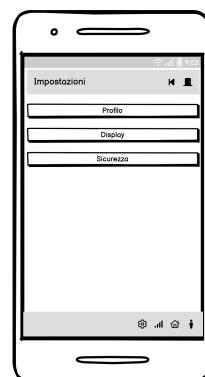


Figura 5.8: Prototipo activity delle impostazioni.

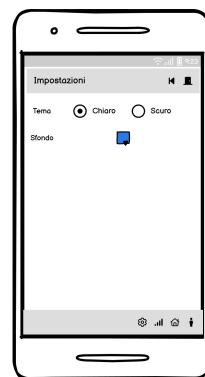


Figura 5.9: Prototipo activity delle impostazioni sul tema, con salvataggio.

5.4 Librerie esterne utilizzate

Nel progetto sono state utilizzate le seguenti librerie esterne, tabella 5.1:

Nome Libreria	Utilizzo
Datastore-Preferences	Implementazione di DataStore per la memorizzazione dei dati di preferenza
Leaflet	Libreria JavaScript per la visualizzazione di mappe interattive
Biometric [27]	Supporto per l'autenticazione biometrica su Android
Play-services-location	API di localizzazione di Google Play Services
Room [28]	Libreria di persistenza di dati locali basata su SQLite
Okhttp3 [29]	Libreria per richieste HTTPS su Android
Castle [30]	Libreria crittografica per Java
Gson [31]	Libreria per la conversione di oggetti Java in formato JSON e viceversa
Coil [32]	Libreria per il caricamento e la visualizzazione di immagini su Android
Tehras-charts [33]	Libreria di grafici per Android

Tabella 5.1: Librerie esterne utilizzate nell'applicazione mobile.

5.5 Activity

In questa sezione, vengono riportate e commentate le activity dell'applicazione finale. Per ciascuna di esse è stata creata sia la versione landscape sia quella portrait al fine di ottimizzare al meglio l'esperienza utente.

5.5.1 Login

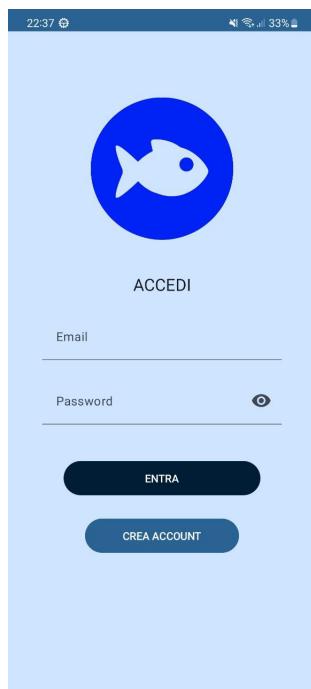


Figura 5.10: Activity di login.

Nella pagina di login, figura 5.10, è possibile inserire l'email e la password negli appositi campi ed effettuare l'accesso attraverso l'apposito pulsante. L'utente ha la possibilità di rendere visibile la password inserita, oppure di

creare un nuovo account. In aggiunta, al fine di facilitare il login all’utente, dopo il primo accesso viene memorizzata l’email e, se il dispositivo è provvisto del sensore di impronte digitali, si avrà la possibilità di accedere senza l’utilizzo della password.

5.5.2 Registrazione

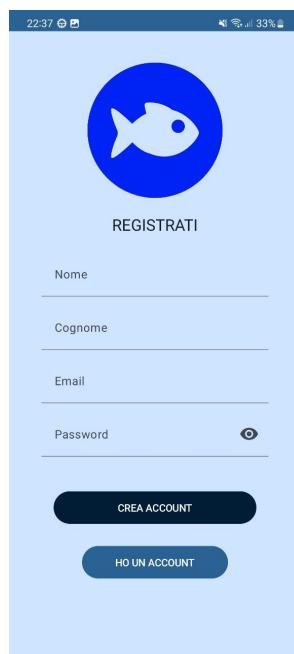


Figura 5.11: Activity di registrazione.

Nella pagina di registrazione, figura 5.11, è possibile inserire le informazioni negli appositi campi ed effettuare la registrazione attraverso l’apposito pulsante. L’utente ha la possibilità di rendere visibile la password inserita in modo tale che possa controllarne la correttezza. Una volta che l’utente si è registrato, verrà portato nella pagina della login per effettuare l’acces-

so. Si ricorda che l'utente, prima di essere operativo, deve essere abilitato manualmente dagli amministratori, nel rispetto delle politiche di sicurezza sui dati.

5.5.3 Profilo



Figura 5.12: Activity del profilo.

Nella successiva schermata, figura 5.12, si vede la foto dell'utente con le relative informazioni personali. In aggiunta, premendo il link della mail, si apre l'app di posta elettronica che consente di far comunicare i due utenti. Il FAB permette di andare nelle impostazioni per modificare le informazioni del profilo.

5.5.4 Homepage

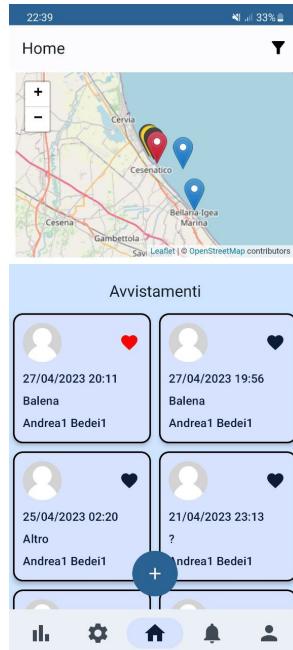


Figura 5.13: Activity della Homepage.

La pagina principale dell'applicazione è la più complessa, figura 5.13. Nella parte centrale, è presente una mappa, con all'interno dei Marker che servono a segnalare la posizione degli avvistamenti. Si ha anche la possibilità di interagire con essa, ad esempio per modificare lo zoom su una determinata area oppure vedere i dettagli di uno specifico avvistamento premendo il relativo Marker. Subito sotto è presente una griglia di avvistamenti: essa comprende tutti gli avvistamenti, sia quelli presenti sul server remoto che quelli locali che verranno caricati in un secondo momento. Per differenziarli, si è deciso di rappresentare le Card con colori diversi tra loro, in modo tale che l'utente possa capire se sono presenti ancora degli avvistamenti lo-

cali. Per ciascun avvistamento, si ha anche la possibilità di aggiungerlo ai preferiti e di visualizzare il profilo dell’utente che lo ha eseguito premendo sull’immagine. Nella barra in alto, oltre a visualizzare il titolo della pagina, si ha la possibilità di aprire una casella di dialogo al fine di selezionare dei filtri: esiste il filtro per selezionare solo gli avvistamenti preferiti oppure filtrare la lista di avvistamenti per animale.

Nella parte bassa della schermata, è presente la barra di navigazione, che porta alle altre pagine dell’applicazione. Il pulsante centrale permette invece di aggiungere un nuovo avvistamento.

5.5.5 Aggiunta avvistamento



Figura 5.14: Activity di aggiunta di un avvistamento.

Nella seguente schermata 5.14 si ha la possibilità di aggiungere tutti i dati dell'avvistamento effettuato. Il campo posizione viene riempito in modo automatico una volta che si è premuto il pulsante del GPS, in questo modo viene utilizzato il sensore GPS del telefono per richiedere le coordinate. Una volta scelti l'animale e la specie, si ha la possibilità di richiedere delle informazioni dettagliate al fine di individuare al meglio la specie di appartenenza. In fondo alla schermata, è presente un pulsante per aggiungere delle immagini (massimo 5) all'avvistamento corrente. Ogni immagine verrà salvata nella galleria del dispositivo e l'utente avrà la possibilità di visualizzarla o di eliminarla. Una volta completato l'inserimento, si può premere il pulsante per caricare l'avvistamento: se è presente una connessione a internet viene caricato direttamente online, altrimenti viene salvato localmente e, quando sarà presente connessione, i dati verranno caricati online. Rappresentiamo il processo con un diagramma di attività 5.15.

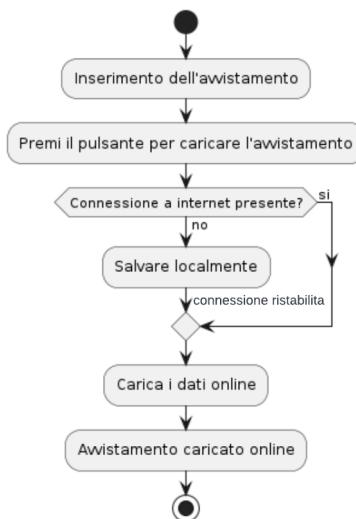


Figura 5.15: Diagramma di attività per il caricamento di un avvistamento.

5.5.6 Visualizzazione avvistamento



Figura 5.16: Activity di visualizzazione di un avvistamento.

La seguente schermata 5.16 è differente a seconda dell'identità del creatore dell'avvistamento. Se il post è stato creato da un utente diverso da quello che lo sta visualizzando, tutti i campi compariranno in sola lettura come nell'immagine. Altrimenti compare una schermata simile a quella dell'aggiunta di un avvistamento, in cui si ha la possibilità di aggiungere foto o modificare i dati. Questo è possibile sia per avvistamenti locali che per avvistamenti online, ma per poter modificare gli avvistamenti online è necessaria una connessione a internet. Oltre alla mappa che fornisce la posizione dell'avvistamento, si ha la possibilità per ogni utente di richiede-

re informazioni sulla specie, se essa è presente. Nella parte sottostante, si ha la possibilità di visualizzare tutte le immagini del relativo avvistamento. Attraverso un intent è possibile visualizzarle anche nella galleria.

5.5.7 Impostazioni



Figura 5.17: Activity di impostazioni.

La schermata 5.17 è composta da tre buttoni, ognuno dei quali porta a una schermata diversa. Attraverso il bottone del profilo si possono modificare tutti i campi relativi alle informazioni personali dell'avvistatore, tra cui nome e cognome, ma anche la foto profilo. Col pulsante display, si ha possibilità di modificare la configurazione del display, ossia utilizzare il te-

ma chiaro oppure scuro. All'inizio, l'app viene avviata nella modalità di configurazione del dispositivo, se l'utente invece cambia questa impostazione l'app mantiene in memoria la scelta fino a un successivo cambiamento. Infine, il pulsante sicurezza permette all'utente di modificare la propria password. In tutte le schermate delle impostazioni si ha la possibilità di eseguire il logout, che disconnette l'utente e lo riporta alla pagina di login.

5.5.8 Notifiche

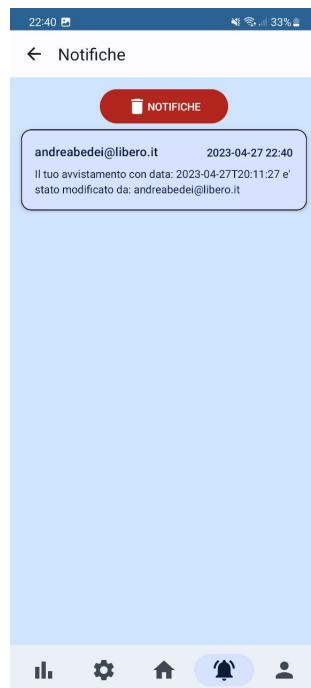


Figura 5.18: Activity delle notifiche.

La schermata 5.18 permette di visualizzare tutte le notifiche che vengono inviate da un apposito bottone nel sito web. Nella notifica vengono comuni-

cate informazioni su cambiamenti avvenuti nei dati degli avvistamenti che l’utente ha caricato. Premendo la notifica, l’utente viene indirizzato al relativo avvistamento. Il pulsante rosso in alto permette di cancellare tutte le notifiche presenti. Quando ne arrivano delle nuove, nella barra sottostante si può notare il cambio dell’icona (da disattiva passa ad attiva).

5.5.9 Statistiche



Figura 5.19: Activity delle Statistiche.

Nella schermata 5.19, sono presenti diversi dati statistici, tra cui:

- contatore del numero di avvistamenti

- contatore del numero di avvistatori.
- grafico con la distribuzione degli animali negli avvistamenti.
- classifica di avvistatori, ordinati per il numero di avvistamenti.
- grafico con la distribuzione dei delfini negli avvistamenti.
- grafico con il numero di avvistamenti negli ultimi giorni.

5.6 Accessibilità e sicurezza

In tutto il progetto si è data estrema importanza ad aspetti di sicurezza dell'app. Per ogni operazione considerata pericolosa da Android vengono richiesti i permessi all'utente con apposite schermate di dialogo. In aggiunta si è cercato di aiutare nel modo migliore possibile l'utente in fatto di user experience e si è dato molto peso all'accessibilità dell'applicazione anche per persone con disabilità, attraverso appositi attributi delle componenti delle schermate.

5.7 Connessione ad internet

Visto che l'applicazione verrà spesso utilizzata quando il dispositivo non possiede una connessione ad internet, si è deciso di richiedere all'utente di effettuare almeno il primo accesso con una connessione. Dopo il primo accesso, i dati più importanti verranno salvati localmente in memoria in modo tale da poter eseguire l'accesso al portale e di poter aggiungere nuovi avvistamenti quando ci si trova in modalità offline. I nuovi avvistamenti verranno caricati appena si avrà la possibilità di contattare il server.

Capitolo 6

Riconoscimento della specie

Nelle specifiche di progetto è stata chiesta la possibilità di riconoscere in modo automatico la specie di delfino rappresentata nelle immagini caricate dagli operatori. Per fare questo, si è utilizzato un algoritmo di visione artificiale che, attraverso la segmentazione delle pinne degli individui, permette di aiutare l'utente a individuare la specie.

6.1 Il problema

I ricercatori utilizzano tecniche come le impronte digitali e il riconoscimento facciale per identificare le persone, ma stanno anche esplorando metodi simili per gli animali marini. Ad esempio, gli scienziati identificano manualmente questi animali basandosi sulla forma e sui segni presenti sulla coda, sulle pinne dorsali, sulla testa e in altre parti del loro corpo. Questa tecnica, nota come "foto-identificazione", consistente nell'identificazione dei segni naturali tramite fotografie, è un importante strumento per la scienza dei mammiferi marini. Essa permette di monitorare singoli individui nel

tempo e di valutare lo stato e le tendenze della popolazione.

Grazie all'utilizzo di un algoritmo di foto-identificazione, i ricercatori possono ridurre i tempi di identificazione delle immagini. Ciò può aprire la strada a nuovi studi su larga scala, altrimenti impossibili o difficili da realizzare. Attualmente la maggior parte degli istituti di ricerca si affida alla corrispondenza manuale con l'occhio umano, un processo che richiede molto tempo e che a volte può essere impreciso.

L'obiettivo principale di questo algoritmo è quello di aumentare la comprensione e la cura degli ambienti marini a livello globale attraverso la scienza della conservazione e l'educazione di alta qualità. Essa sviluppa un modello per abbinare i singoli delfini alla loro specie, basandosi sulle loro uniche e spesso sottili marcature naturali, con particolare attenzione alle pinne dorsali e alle viste laterali del loro corpo, utilizzando un vasto set di dati creato da 28 istituti di ricerca.

6.2 Modello

In questa sezione parleremo del modello utilizzato per la classificazione delle immagini con l'attribuzione della relativa etichetta.

6.2.1 Identificazione della specie

Per identificare la specie di delfino da un'immagine si è presa in considerazione la pinna dell'esemplare. L'algoritmo prima scansiona l'immagine e poi segmenta la parte della pinna, la quale gli permette di identificare la

specie di appartenenza. Un esempio del funzionamento è descritto dall'immagine 6.1:

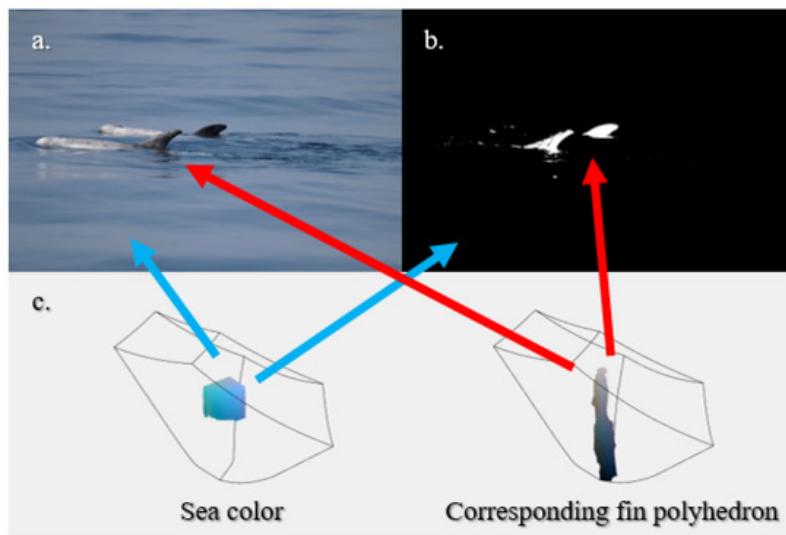


Figura 6.1: Segmentazione pinna dell'esemplare. [34]

Per eseguire questa operazione si è utilizzato il modello preaddestrato MobileNetV2.

6.2.2 MobileNetV2

MobileNetV2 [35] è un modello di rete neurale convoluzionale progettato per l'elaborazione delle immagini. È stato sviluppato da Google ed è una versione migliorata del modello originale MobileNet.

MobileNetV2 è noto per la sua efficienza computazionale e la sua capacità di eseguire l'elaborazione delle immagini in tempo reale su dispositivi con risorse limitate. Il modello sfrutta l'architettura di base dei blocchi di con-

voluzione profonda.

Una delle caratteristiche chiave di MobileNetV2 è l'uso di un blocco di convoluzione chiamato *bottleneck* che utilizza un'operazione di separazione dei canali. Questo significa che la convoluzione viene suddivisa in due fasi: una convoluzione 1x1 per ridurre la dimensionalità e una convoluzione 3x3 per acquisire le informazioni spaziali. Questa tecnica riduce il numero di parametri e la quantità di calcoli richiesti.

Esso utilizza anche una tecnica chiamata *skip connection* per facilitare il flusso di informazioni attraverso la rete. Le skip connection connettono alcune delle uscite dei livelli precedenti direttamente ai livelli successivi, consentendo il passaggio di informazioni a basso livello che altrimenti potrebbero essere perse durante la propagazione all'interno della rete.

Il modello è stato addestrato su grandi set di dati di immagini, come ImageNet [36], per apprendere una vasta gamma di caratteristiche visive. Può essere utilizzato per una varietà di compiti legati all'elaborazione delle immagini, come il riconoscimento degli oggetti, la classificazione delle immagini e la segmentazione semantica.

6.3 Dataset

Il dataset [37] comprende immagini di balene e di delfini che possono essere identificati attraverso le loro forme, le caratteristiche e le segnalazioni (alcune naturali, altre acquisite) delle loro pinne dorsali, dei dorsi, delle teste e dei fianchi. Alcune specie e alcuni individui hanno caratteristiche molto distinte, mentre altre sono molto meno individuabili. Inoltre, le caratteristiche individuali possono variare nel tempo. I dati che compongono il dataset contengono immagini di oltre 15.000 singoli mammiferi marini di 30 specie diverse, raccolte dalle 28 diverse organizzazioni di ricerca citate

poc'anzi. Gli individui sono stati identificati manualmente e a loro è stato assegnato un ID-individuo dai ricercatori marini.

6.3.1 Specie presenti e quantità di immagini

Il dataset è stato suddiviso in cartelle in base alla specie a cui ogni animale appartiene con le seguenti quantità di immagini:

Tabella 6.1: Numero di immagini per ogni specie.

Specie	Numero immagini
Tursiope	10781
Beluga	7443
Megattera	7392
Balena Blu	4830
Pseudorca	3326
Delfino Scuro	3139
Orca	2455
Stenella Dal Lungo Rostro	1700
Peponcefalo	1689
Balenottera Minore	1608
Balenottera Comune	1324
Balena Grigia	1123
Balena Franca Australe	866
Stenella Maculata	635
Balenottera Boreale	428
Globicefalo di Gray	367
Delfino Comune	347

Continua nella pagina successiva

Tabella 6.1 – Continuazione dalla pagina precedente

Specie	Numero immagini
Zifio	320
Globicephala	262
Globicefalo	238
Delfino Bianco Atlantico	229
Balaenoptera Edeni	154
Globis	116
Cefalorinco di Commerson	90
Feresa	76
Steno	60
Lagenodelfino	14
Zifio4	8
Zifio3	7
Zifio2	6

Come si può notare dalla tabella 6.1, per ciascuna specie c’è un diverso numero di immagini, quindi ci sarà un grado di accuratezza diverso per ciascuna categoria.

6.4 Algoritmo di riconoscimento

Per quanto riguarda l’algoritmo di riconoscimento, esso è diviso in due parti:

- Fase di training
- Fase di riconoscimento dell’immagine

Entrambe le fasi sono realizzate attraverso degli script in codice python. Tramite comandi bash, esse sono integrate nel codice php del server web.

6.4.1 Fase di training

La fase di training dell'algoritmo di visione artificiale è un processo cruciale che ha l'obiettivo di formare il modello sulla base dei dati forniti. Questo processo consente all'algoritmo di riconoscere ed elaborare correttamente i segnali visivi, in modo da eseguire compiti specifici come la classificazione, la rilevazione e la localizzazione di oggetti all'interno di un'immagine. Durante questa fase, l'algoritmo viene alimentato con un gran numero di dati di formazione, che consistono in immagini etichettate con informazioni sul loro contenuto. Questi dati vengono utilizzati per formare un modello matematico che descrive la relazione tra gli input visivi e le relative etichette. Il modello viene quindi testato su un insieme di dati di prova, che non sono stati utilizzati durante il processo di formazione. Questi dati servono per valutare la precisione del modello, ovvero la sua capacità di prevedere correttamente le etichette associate agli input visivi. In base ai risultati ottenuti, il modello può essere raffinato e ripetere il processo di formazione fino a quando non si ottiene una precisione sufficientemente alta.

L'algoritmo di visione artificiale può essere addestrato, utilizzando diverse tecniche di apprendimento automatico, come l'apprendimento profondo (deep learning), l'apprendimento supervisionato e l'apprendimento non supervisionato. Queste tecniche sono basate su diverse architetture di rete neurale, che utilizzano algoritmi matematici per elaborare i dati visivi e produrre output di alta qualità.

Il codice che è stato utilizzato per realizzare questa funzionalità verrà scomposto in più parti in modo da descriverlo nel dettaglio.

Librerie utilizzate

Listing 6.1: Le librerie utilizzate

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
4 from pathlib import Path
5 import os.path
6 import seaborn as sns
7 from sklearn.model_selection import train_test_split
8 import tensorflow as tf
9 from sklearn.metrics import confusion_matrix,
    classification_report
```

Descrizione delle librerie usate nel codice 6.1:

- **numpy [38]**: è una libreria di alto livello per il calcolo scientifico in Python. Fornisce supporto per array multi-dimensionale, funzioni matematiche e altri strumenti utili per il calcolo numerico.
- **matplotlib [39]**: è una libreria per la creazione di visualizzazioni dati. Può essere utilizzata per creare grafici e altre forme di rappresentazione visiva dei dati.
- **pathlib [40]**: è una libreria per lavorare con file system e percorsi. Fornisce un modo semplice per creare, manipolare e lavorare con percorsi di file e directory.
- **os.path**: è un modulo che fornisce funzioni per lavorare con i percorsi dei file in un sistema operativo.

- **pandas [41]**: è una libreria per l'analisi dei dati in Python. Fornisce strumenti per la manipolazione e l'elaborazione di dati strutturati in forma di tabelle.
- **seaborn [42]**: è una libreria di visualizzazione dei dati che si basa su matplotlib.
- **tensorflow [43]**: è una libreria open-source per il machine learning e il deep learning sviluppata da Google. Può essere utilizzata per costruire, addestrare e utilizzare modelli di machine learning e deep learning.
- **scikit-learn [44]**: è una libreria di machine learning che fornisce un insieme di algoritmi di apprendimento automatico. Le funzioni che vengono utilizzate della libreria sono:
 - **train_test_split**: viene utilizzata per suddividere i dati in un insieme di addestramento e di test.
 - **confusion_matrix**: permette di creare una matrice di confusione che mostra le previsioni del modello confrontate con i valori effettivi.
 - **classification_report**: fornisce una relazione di classificazione che riassume le prestazioni del modello.

Caricamento dei dati

Listing 6.2: Caricamento dei dati di training.

```
1 df = pd.read_csv('../dolphin/train.csv')
2 df.head()
```

Il codice 6.2 permette di caricare il file CSV denominato "train.csv" in un DataFrame, una struttura di dati tabulare a 2 dimensioni utilizzata per l'analisi dei dati.

La funzione pd.read_csv() legge i dati dal file CSV e li carica in un oggetto. La seconda riga, df.head(), mostra le prime cinque righe del DataFrame al fine di assicurarsi che tutti i dati siano stati caricati correttamente. Il risultato è la tabella 6.2:

	Immagine	Specie	ID immagine
0	00021adfb725ed.jpg	Tursiope	cadddb1636b9
1	000562241d384d.jpg	Orca	1a71fbb72250
2	0007c33415ce37.jpg	Delfino_comune	60008f293a2b
3	0007d9bca26a99.jpg	Balenottera	4b00fe572063
4	00087baf5cef7a.jpg	Balena_blu	8e5253662392

Tabella 6.2: Struttura del dataframe caricato per l'analisi dei dati.

Controllo caricamento e distribuzione delle categorie

Listing 6.3: Controllo del corretto caricamento dei dati.

```
1 df.shape
2 df.species.nunique()
3 df.species.value_counts().sort_index()
4 plt.figure(figsize=(20, 10))
5 df['species'].value_counts().sort_values(ascending=
    True).plot(kind='barh');
```

Nel codice 6.3, la prima riga permette di verificare che siano stati caricati nel DataFrame tutti i dati del file, nella seconda si verifica che il numero di specie (nomi unici) sia corretto. Invece la terza permette di visualizzare tutti i nomi delle specie col relativo numero di immagini ad esse associate. Le ultime due righe permettono di disegnare un grafico a barre orizzontali che mostra la distribuzione delle categorie della colonna "species" del DataFrame. In particolare:

- `df['species']` seleziona la colonna "species" dal DataFrame.
- `df.value_counts()` conta il numero di elementi univoci per ogni valore nella colonna species.
- `sort_values(ascending=True)` ordina i valori in ordine crescente.
- `plot()` disegna il grafico.

Questo grafico aiuta a visualizzare la distribuzione delle categorie nella colonna species, che è utile per comprendere la quantità e la distribuzione delle diverse categorie presenti nei dati. Il risultato è rappresentato nella figura 6.2:

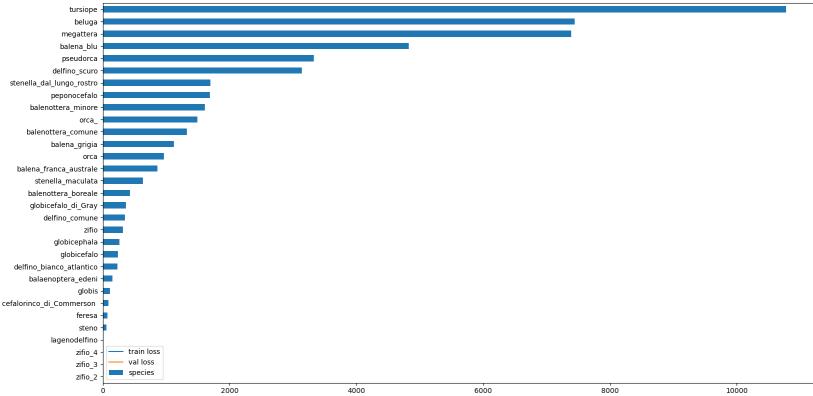


Figura 6.2: Distribuzione delle categorie nella colonna species.

Associazione immagine etichetta

Listing 6.4: Associazione immagini alle relative etichette.

```

1 image_dir = Path('../imagessortedbyspecies/
                  train_species_list')
2 filepaths = list(image_dir.glob(r'**/*.jpg'))
3 labels = list(map(lambda x: os.path.split(os.path.
                  split(x)[0])[1], filepaths))
4 filepaths = pd.Series(filepaths, name='Filepath').
                  astype(str)
5 labels = pd.Series(labels, name='Label')
6 image_df = pd.concat([filepaths, labels], axis=1)
7 image_df

```

In queste righe di codice 6.4, si sta creando un DataFrame Pandas che associa a ciascun file immagine un'etichetta. In particolare:

- **image_dir**: è un oggetto che rappresenta il percorso della directory che contiene le immagini classificate per specie.
- **filepaths**: è una lista che contiene i percorsi completi di ogni file immagine nella directory. La funzione `list(image_dir.glob(r'**/*.jpg'))` utilizza la funzione `glob` per trovare tutti i file con estensione ".jpg" nella directory.
- **labels**: è una lista che contiene le etichette associate a ciascuna immagine. La funzione `map()` utilizza la funzione `os.path.split()` per estrarre il nome della cartella che contiene l'immagine, che viene utilizzato come etichetta per l'immagine.
- **filepaths e labels**: sono entrambi convertiti in oggetti Series di Pandas, che sono array unidimensionali etichettati.
- Infine, gli oggetti sono concatenati insieme per creare un DataFrame.

Il risultato è un DataFrame che ha due colonne: una contiene i percorsi dei file immagine e l'altra contiene le etichette associate a ciascuna immagine. Questo DataFrame può essere utilizzato per lavorare con i dati delle immagini. Il risultato finale è quello riportato in tabella 6.3:

	Filepath	Label
0	./train/img0.jpg	globicefalo_di_Gray
1	./train/img1.jpg	globicefalo_di_Gray
...
51032	./train/img51032.jpg	tursiope

Tabella 6.3: DataFrame con associazione immagine-etichetta.

Divisione del dataframe

Listing 6.5: Divisione del dataframe.

```
1 train_df, test_df = train_test_split(image_df,  
                                         train_size=0.8, shuffle=True, random_state=1)
```

Nel codice sopra riportato 6.5, viene utilizzata la funzione `train_test_split` del pacchetto scikit-learn per dividere un dataframe `image_df` in due parti, una chiamata `train_df` e l'altra `test_df`.

La funzione `train_test_split` divide un insieme di dati in due parti, una parte di addestramento (train) e una parte di test (test). La dimensione della parte di addestramento è specificata dall'argomento `train_size` e viene impostata su 0,8 in questo caso, il che significa che l'80% dei dati verrà utilizzato per l'addestramento e il restante 20% verrà utilizzato per i test.

L'argomento `shuffle` viene impostato su `True`, il che significa che i dati verranno mescolati casualmente prima della divisione. Questo è importante per garantire una distribuzione casuale dei dati tra la parte di addestramento e quella di test.

L'argomento `random_state` viene impostato su 1, il che significa che verrà utilizzato un seme specifico per la generazione di numeri casuali. Questo garantisce che i dati vengano divisi allo stesso modo ogni volta che il codice viene eseguito con lo stesso seme.

Istanziamento dei generatori

Listing 6.6: Istanziamento dei generatori.

```
1 train_generator = tf.keras.preprocessing.image.  
    ImageDataGenerator(  
2     preprocessing_function=tf.keras.applications.  
        mobilenet_v2.preprocess_input,  
3 )  
4 test_generator = tf.keras.preprocessing.image.  
    ImageDataGenerator(  
5     preprocessing_function=tf.keras.applications.  
        mobilenet_v2.preprocess_input  
6 )
```

Il codice 6.6 crea due generatori di immagini, uno chiamato `train_generator` e l'altro chiamato `test_generator`. Entrambi i generatori sono istanziati dalla classe `ImageDataGenerator` del pacchetto `tf.keras.preprocessing.image`.

La classe `ImageDataGenerator` viene utilizzata per pre-elaborare i dati delle immagini in modo che possano essere utilizzati come input per una rete neurale. In questo caso, la pre-elaborazione viene effettuata utilizzando la funzione `preprocess_input` della libreria `tf.keras.applications.mobilenet_v2`. Questa funzione si occupa di normalizzare i valori dei pixel delle immagini in modo che siano adatti per l'utilizzo con la rete neurale MobileNetV2.

Da notare che la funzione `preprocess_input` viene passata come argomento `preprocessing_function` a entrambe le istanze di `ImageDataGenerator`. Questo garantisce che sia il generator di addestramento che quello di test vengano pre-elaborati allo stesso modo.

In particolare, per il caso di MobileNetV2 i dati vengono normalizzati col seguente procedimento:

1. **Regolazione della gamma dei valori dei pixel:** La funzione preprocess_input si assicura che i valori dei pixel siano nella gamma corretta, ossia tra 0 e 255, che è il range comune per le immagini RGB con 8 bit per canale.
2. **Sottrazione della media dei pixel delle immagini di addestramento:** Calcola la media dei valori dei pixel per le immagini di addestramento utilizzate per allenare il modello. Successivamente, sottrae questa media da ogni valore del pixel dall'immagine in input. Con lo scopo di centrare i dati intorno a zero, in questo modo la rete neurale può apprendere i modelli in modo più efficace.
3. **Divisione per la deviazione standard dei pixel delle immagini di addestramento:** Calcola la deviazione standard dei valori dei pixel delle immagini di addestramento utilizzate per addestrare il modello. Quindi, ogni valore del pixel nell'immagine di input viene diviso per questa deviazione standard. Questo passaggio scala i dati in modo che abbiano una varianza unitaria, aiutando a ridurre le differenze di scala tra le diverse feature.

Creazione generatore di immagini da un dataframe

Listing 6.7: Creazione del generatore di immagini da un dataframe di addestramento.

```
1 train_images = train_generator.flow_from_dataframe(  
2     dataframe=train_df,  
3     x_col='Filepath',  
4     y_col='Label',  
5     target_size=(224, 224),  
6     color_mode='rgb',  
7     class_mode='categorical',  
8     batch_size=32,  
9     shuffle=True,  
10    seed=42,  
11    subset='training'  
12 )
```

Il codice 6.7 utilizza la funzione `flow_from_dataframe` del generatore di addestramento (`train_generator`) per creare un generatore di immagini da un dataframe di addestramento (`train_df`). La funzione `flow_from_dataframe` legge le immagini dal file system e le trasforma in un formato utilizzabile per l'addestramento del modello di apprendimento automatico. Gli argomenti passati sono:

- **dataframe**: specifica il dataframe di origine per il generatore di immagini.
- **x_col**: specifica la colonna che contiene il percorso del file per ciascuna immagine.

- **y_col**: specifica la colonna che contiene l'etichetta per ciascuna immagine.
- **target_size**: specifica le dimensioni delle immagini di destinazione e viene impostato su (224, 224).
- **color_mode** viene impostato su `rgb`, il che significa che le immagini verranno lette come immagini a colori.
- **class_mode**: viene impostato su `categorical`, il che significa che le etichette verranno restituite come matrici one-hot codificate. Esse sono una rappresentazione numerica delle categorie, ovvero un vettore di lunghezza pari al numero di categorie possibili, in cui solo un elemento è impostato su 1 e gli altri sono impostati su 0.
- **batch_size**: specifica il numero di immagini da restituire in un singolo batch (sottoinsieme di dati) e viene impostato in questo caso su 32.
- **seed**: viene impostato su 42, il che significa che verrà utilizzato un seme specifico per la generazione di numeri casuali durante la mescolatura delle immagini.
- **subset**: viene impostato su `training`, il che significa che il generatore di immagini verrà utilizzato solo per l'addestramento del modello.

Creazione generatore di immagini di test

Listing 6.8: Creazione del generatore di immagini di test.

```
1 test_images = test_generator.flow_from_dataframe(  
2     dataframe=test_df,  
3     x_col='Filepath',  
4     y_col='Label',  
5     target_size=(224, 224),  
6     color_mode='rgb',  
7     class_mode='categorical',  
8     batch_size=32,  
9     shuffle=False  
10 )
```

Il codice 6.8 è quasi identico a quello descritto sopra e crea un generatore di immagini di test per un modello di apprendimento automatico, settando i relativi parametri.

Creazione modello pre-addestrato di MobileNetV2

Listing 6.9: Modello pre-addestrato di MobileNetV2t.

```
1 pretrained_model = tf.keras.applications.MobileNetV2(  
2     input_shape=(224, 224, 3),  
3     include_top=False,  
4     weights='imagenet',  
5     pooling='avg'  
6 )  
7 pretrained_model.trainable = False
```

Il codice 6.9 crea un modello pre-addestrato di MobileNetV2 con Tensor-Flow. Il modello viene creato utilizzando la funzione `tf.keras.applications.MobileNetV2` e viene configurato con una forma di input di `(224, 224, 3)`, che corrisponde alla dimensione delle immagini che verranno utilizzate. Per il modello vengono settate le seguenti proprietà:

- **include_top**: viene impostata su `False`, il che significa che la parte superiore del modello, che include i livelli di classificazione, verrà esclusa.
- **weights**: viene impostata su `"imagenet"`, il che significa che il modello verrà inizializzato con i pesi pre-addestrati sulla base di ImageNet(vasto insieme di dati di immagini).
- **pooling**: viene impostata su `"avg"`, il che significa che verrà utilizzata la media pooling per la riduzione delle dimensioni delle feature.

Infine, la proprietà `trainable` del modello viene impostata su `False`, il che significa che i pesi del modello non verranno adattati durante l'addestramento del modello finale. Questo è utile perché il modello MobileNetV2 è già stato addestrato su un vasto insieme di immagini e le feature estratte sono molto utili per molte diverse attività di classificazione delle immagini. Pertanto si desidera utilizzare queste feature pre-addestrate come caratteristiche fisse nel modello finale.

Definizione di un nuovo modello

Listing 6.10: Aggiunta di due strati densi alla rete.

```
1 inputs = pretrained_model.input
2 x = tf.keras.layers.Dense(128, activation='relu')(
    pretrained_model.output)
3 x = tf.keras.layers.Dense(128, activation='relu')(x)
4 outputs = tf.keras.layers.Dense(30, activation='
    softmax')(x)
5 model = tf.keras.Model(inputs, outputs)
6 print(model.summary())
```

Il codice 6.10 definisce un nuovo modello ”model” che utilizza come input il modello pre-addestrato. Quindi vengono aggiunti due strati densi alla rete, ognuno con 128 unità e un’attivazione ”relu”. Gli strati densi sono strati di una rete neurale artificiale che hanno una grande quantità di pesi che sono connessi a ogni neurone dello strato. Questi strati sono utilizzati per elaborare le informazioni provenienti dai livelli precedenti della rete e per produrre un’uscita che rappresenta le caratteristiche più complesse delle immagini. In questo codice, i due strati densi con 128 unità e attivazione ”relu” sono stati aggiunti alla rete per estrarre ulteriori informazioni sulle immagini. L’attivazione ”relu” è un’attivazione non lineare utilizzata comunemente nei modelli di rete neurale. Questa attivazione restituisce il valore di ingresso se è positivo e restituisce zero se è negativo. Questo rende la rete più efficiente e capace di apprendere relazioni più complesse tra gli input e le relative uscite.

Infine, viene aggiunto un ultimo strato denso con 30 unità e un’attivazione ”softmax”. Essa produce una distribuzione di probabilità delle classi. Questa attivazione è spesso utilizzata come ultimo strato di una rete neurale,

poiché trasforma l'uscita della rete in una probabilità per ogni classe. In questo caso, l'attivazione softmax con 30 unità viene utilizzata come ultimo strato del modello per classificare le immagini in 30 categorie diverse. Infine, il codice stampa un riepilogo del modello, che fornisce informazioni sul numero di parametri del modello e sulle sue dimensioni.

Compilazione modello di apprendimento automatico

Listing 6.11: Compilazione del modello di apprendimento automatico.

```
1 model.compile(  
2     optimizer='adam',  
3     loss='categorical_crossentropy',  
4     metrics=['accuracy'])  
5 )
```

Il codice 6.11 è relativo alla compilazione del modello di apprendimento automatico. La compilazione è l'ultimo passo prima di addestrare il modello. Durante la compilazione, si specificano i seguenti parametri:

1. **Ottimizzatore:** 'adam' è un tipo di ottimizzatore che viene utilizzato per la formazione del modello. L'ottimizzatore "Adam" è un ottimizzatore di gradiente estensione della discesa del gradiente stocastico. La discesa del gradiente stocastico (SGD) è un algoritmo di addestramento della rete. Il processo consiste nel trovare i pesi ottimali del modello che minimizzano la loss function, che misura la differenza tra le previsioni del modello e i valori desiderati. Nella discesa del gradiente stocastico, viene calcolato il gradiente (una misura della pendenza) della loss function rispetto ai pesi del modello

su un singolo esempio o su un piccolo gruppo di esempi (batch) alla volta. Questo gradiente viene quindi utilizzato per aggiornare i pesi del modello nella direzione opposta, in modo da minimizzare la loss function. Questo processo viene ripetuto più volte su diverse iterazioni fino a quando i pesi non convergono verso un valore ottimale. La discesa del gradiente stocastico è chiamata stocastica perché il gradiente viene calcolato su un singolo esempio o un piccolo gruppo di esempi, anziché sull'intero set di dati, il che rende il processo più efficiente rispetto alla discesa del gradiente batch. Inoltre, l'utilizzo di un singolo esempio o di un piccolo gruppo di esempi a ogni passo può aiutare a evitare di rimanere bloccati in un minimo locale durante il processo di addestramento.

2. **Loss function:** La loss function utilizzata è categorical_crossentropy. Essa è una loss function utilizzata per la classificazione multi-classe che calcola la distanza tra la distribuzione di probabilità prevista e quella reale.
3. **Metriche:** La metrica utilizzata per valutare il modello è 'accuracy', che misura la percentuale di previsioni corrette fatte dal modello.

Possiamo dire che la compilazione del modello specifica come il modello dovrebbe essere addestrato, come la perdita dovrebbe essere calcolata e come dovrebbe essere valutato il modello.

Fase di training del modello

Listing 6.12: Fase di training del modello.

```
1 r = model.fit(  
2     train_images,  
3     validation_data=test_images,  
4     epochs=5,  
5 )
```

Il codice 6.12 rappresenta il training del modello. Il metodo fit accetta i dati di training:

- **train_images**: le immagini che si utilizzano per addestrare il modello.
- **validation_data**: fornisce i dati di convalida al modello durante il training.
- **epochs**: specifica il numero di volte che i dati di training verranno presentati al modello durante l’addestramento.

Il risultato dell’addestramento viene salvato nella variabile r. L’output dell’addestramento del modello, le informazioni sul suo progresso e la sua precisione nelle varie epoche vengono mostrate nella tabella 6.4:

Epoch	Time	Train. loss	Train. accuracy	Valid. loss	Valid. accuracy
1	155s	0.7081	0.7903	0.5015	0.8443
2	150s	0.4338	0.8634	0.5241	0.8386
3	148s	0.3507	0.8870	0.4261	0.8665
4	147s	0.2905	0.9040	0.4281	0.8700
5	151s	0.2480	0.9164	0.4433	0.8715

Tabella 6.4: Risultati sul training e sul validation set.

I valori che vengono rappresentati nella tabella sono:

- **Epoch:** Il numero di epoca a cui è arrivato. Un'epoca rappresenta l'elaborazione dell'intero dataset di addestramento. In genere, si eseguono molte epoche di addestramento su un modello di apprendimento automatico per aumentare la sua precisione.
- **Time:** Tempo totale che ci vuole per completare un singolo step di training. Gli step di training sono il numero di passaggi che il modello compie sull'intero set di dati di addestramento prima di terminare un'epoca. Ogni step di addestramento è costituito da un gruppo di esempi di addestramento chiamati batch. Il numero di step di addestramento per un'epoca è calcolato dividendo il numero totale di esempi di addestramento per il numero di esempi in un singolo batch. Questo significa che, per ogni epoca, il modello si eserciterà su ogni esempio di addestramento una volta e solo una.
- **Training loss:** è una misura della quantità di errore nel modello durante il training. In altre parole, rappresenta la differenza tra le previsioni del modello e i valori effettivi per le osservazioni di training. Si utilizza come funzione di costo nell'ottimizzazione del modello,

dove l’obiettivo è minimizzare questo valore. Una riduzione della Training Loss indica un miglioramento del modello e un adattamento più preciso alle osservazioni di training.

- **Training accuracy:** è una metrica che misura l’accuratezza della previsione del modello sui dati di training. Viene calcolata come la percentuale di previsioni esatte rispetto al numero totale di previsioni effettuate dal modello sui dati di training. Esso è utilizzato per valutare la performance del modello durante il training e per verificare che esso stia effettivamente imparando dai dati di training. Tuttavia, essa non è un buon indicatore della performance del modello sui dati di test o sui dati nuovi, poiché c’è il rischio che il modello abbia imparato a memoria i dati di training invece che generalizzare davvero le informazioni presenti in essi.
- **Validation loss:** è una metrica utilizzata per valutare la performance del modello durante la validazione. Questa metrica rappresenta la quantità di errore che il modello commette quando prevede le classificazioni per i dati di validazione. Il valore di Validation loss viene calcolato come la media della funzione di costo tra le previsioni effettuate dal modello e le verità di etichetta per i dati di validazione. Più basso è il suo valore, migliore è la capacità del modello di generalizzare su dati nuovi.
- **Validation accuracy:** è una metrica utilizzata per valutare la precisione del modello. Si riferisce alla percentuale di previsioni corrette effettuate dal modello su un set di dati di validazione, che è separato dal set di dati di addestramento. Essa fornisce una misura dell’accuratezza del modello sui dati che non ha ancora visto e quindi fornisce

una valutazione più equa della sua capacità di generalizzare su dati nuovi.

Rappresentazione delle curve di perdita

Listing 6.13: Codice per il grafico delle curve di perdita.

```
1 plt.plot(r.history['loss'], label='train loss')
2 plt.plot(r.history['val_loss'], label='val loss')
3 plt.legend()
4 plt.show()
```

Il codice 6.13 traccia due curve: una rappresenta la perdita (loss) ottenuta dal modello durante l’addestramento sui dati di training, l’altra rappresenta la perdita ottenuta dal modello durante la validazione su un insieme di dati di validazione. Il grafico quindi mostra come la perdita del modello vari durante l’addestramento e la validazione, fornendo informazioni utili per valutare l’andamento del processo di apprendimento e l’efficacia del modello. La funzione plt.plot viene utilizzata per tracciare il grafico, mentre i dati sono forniti dal dizionario r.history che contiene la storia dell’addestramento e della validazione. Il risultato è rappresentato nella figura 6.3.

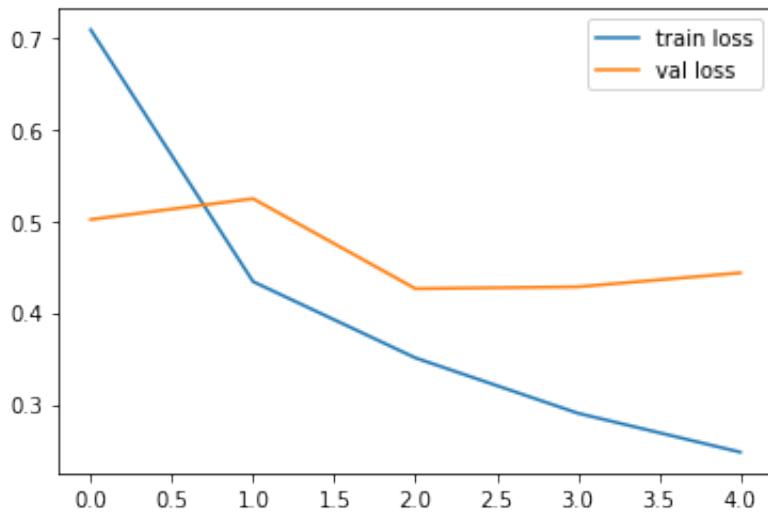


Figura 6.3: Perdita (loss) ottenuta dal modello durante l’addestramento (blu) e perdita ottenuta dal modello durante la validazione (arancione).

Rappresentazione delle curve di accuratezza

Listing 6.14: Codice per il grafico delle curve di accuratezza.

```

1 plt.plot(r.history['accuracy'], label='train acc')
2 plt.plot(r.history['val_accuracy'], label='val acc')
3 plt.legend()
4 plt.show()
```

Il codice 6.14 traccia due curve: una rappresenta l’accuratezza (accuracy) ottenuta dal modello durante l’addestramento sui dati di training, l’altra rappresenta l’accuratezza ottenuta dal modello durante la validazione su un insieme di dati di validazione. Quindi il grafico, figura 6.4, mostra come l’accuratezza del modello vari durante l’addestramento e la valida-

zione, fornendo informazioni utili per valutare l'andamento del processo di apprendimento e l'efficacia del modello.

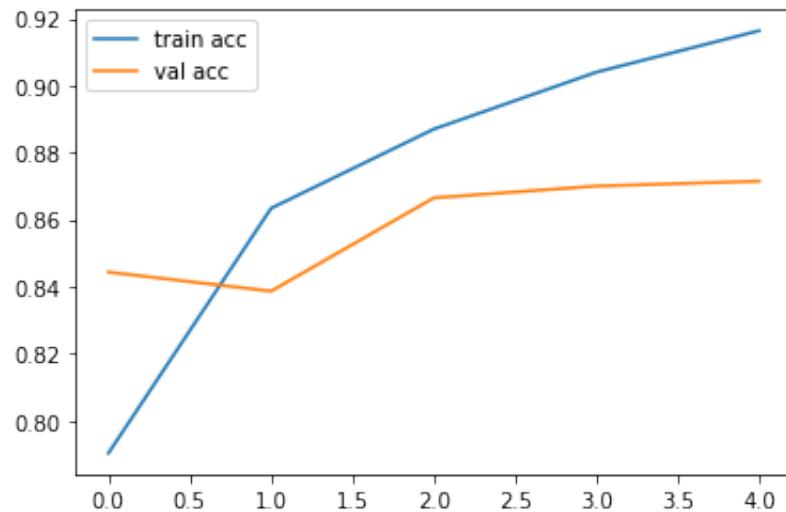


Figura 6.4: L'accuratezza ottenuta dal modello durante l'addestramento (blu) e l'accuratezza ottenuta dal modello durante la validazione (arancione).

Significato delle rappresentazioni

Come è possibile vedere dai grafici delle due sezioni precedenti, si nota che il grafico che rappresenta la perdita, ovvero la misura della discrepanza tra i valori predetti dal modello e i valori reali attraverso l'addestramento, porta a minimizzare la perdita in modo tale che i valori predetti si avvicinino sempre di più a quelli reali. Invece, il grafico dell'accuratezza ovvero la percentuale di predizioni corrette effettuate dal modello rispetto al numero totale di predizioni effettuate, con l'addestramento si alza.

Determinazione dell'accuratezza media

Listing 6.15: Determinazione dell'accuratezza.

```
1 results = model.evaluate(test_images, verbose=0)
2 print("Test Accuracy: {:.2f}%".format(results[1] *
100))
```

Il codice 6.15 valuta le prestazioni del modello sui dati di test, utilizzando il metodo *evaluate* del modello. Esso viene invocato passando come input le immagini di test(*test_images*) e restituisce una lista di valori, tra cui la perdita (*loss*) e l'accuratezza (*accuracy*) ottenute dal modello sui dati di test. Successivamente, il codice stampa a video l'accuratezza ottenuta sul test set. Nel nostro caso l'accuratezza ottenuta è del 86.95%.

Matrice di confusione e report di classificazione

Listing 6.16: Codice per la creazione della matrice di confusione e del report di classificazione.

```
1 predictions = np.argmax(model.predict(test_images),
axis=1)
2 cm = confusion_matrix(test_images.labels, predictions)
3 clr = classification_report(test_images.labels,
predictions, target_names=test_images.class_indices
, zero_division=0)
```

Il codice 6.16 prende in input le immagini di test e utilizza il modello addestrato per effettuare le predizioni. In particolare, utilizza il metodo *predict* del modello per ottenere le predizioni in forma di probabilità, e la funzione *argmax* di Numpy per ottenere la classe con la probabilità maggiore.

Successivamente, il codice utilizza la matrice di confusione per valutare le prestazioni del modello. La matrice di confusione è una tabella che mostra il numero di predizioni corrette e di predizioni errate del modello per ciascuna classe di appartenenza. In particolare, la variabile `cm` rappresenta la matrice ottenuta utilizzando le etichette di classe reali e le etichette di classe predette dal modello. La matrice di confusione è rappresentata come una matrice quadrata, in cui ogni riga rappresenta la classe reale e ogni colonna rappresenta la classe predetta. Gli elementi sulla diagonale principale rappresentano il numero di predizioni corrette, mentre gli elementi fuori dalla diagonale principale rappresentano il numero di predizioni errate.

L'ultimo passo del codice utilizza la funzione `classification_report` della libreria Scikit-learn per generare un report di classificazione. Il report di classificazione, è un resoconto dettagliato delle prestazioni del modello. In particolare, il report fornisce informazioni come l'accuratezza, la precisione, la recall e l'F1-score per ogni classe di appartenenza. L'accuratezza è il rapporto tra il numero di predizioni corrette e il numero totale di predizioni, mentre la precisione è il rapporto tra il numero di predizioni corrette per una classe e il numero totale di predizioni per quella classe. La recall invece è il rapporto tra il numero di predizioni corrette per una classe e il numero totale di esempi di quella classe nel dataset di test. Mentre l'F1-score è una media armonica di precisione e recall. Il report di classificazione è un'importante fonte di informazioni per valutare le prestazioni del modello e per identificare le classi con cui il modello presenta maggiori difficoltà.

Visualizzazione della matrice di confusione

Listing 6.17: Codice per la visualizzazione della matrice di confusione.

```
1 plt.figure(figsize=(15, 15))
2 sns.heatmap(cm, annot=True, fmt='g', vmin=0, cmap='
    Blues', cbar=False)
3 plt.xticks(ticks=np.arange(30) + 0.5, labels=
    test_images.class_indices, rotation=90)
4 plt.yticks(ticks=np.arange(30) + 0.5, labels=
    test_images.class_indices, rotation=0)
5 plt.xlabel("Predicted")
6 plt.ylabel("Actual")
7 plt.title("Confusion Matrix")
8 plt.show()
```

Il codice 6.17 disegna una heatmap (mappa di calore) della matrice di confusione, che visualizza graficamente le informazioni contenute nella matrice di confusione in modo più chiaro. In particolare, con l'utilizzo della libreria Seaborn per creare la heatmap, vengono specificati i seguenti parametri: il parametro `annot=True` per mostrare i numeri all'interno della heatmap, il parametro `fmt='g'` per formattare i numeri come interi, e il parametro `vmin=0` per impostare il valore minimo del range di colori.

Inoltre, il codice utilizza le funzioni `xticks` e `yticks` di Matplotlib per impostare le etichette degli assi x e y con i nomi delle classi di appartenenza, utilizzando il parametro `labels` e il dizionario `test_images.class_indices` per ottenere i nomi delle classi.

La matrice di confusione così ottenuta è quella riportata nella figura 6.5.

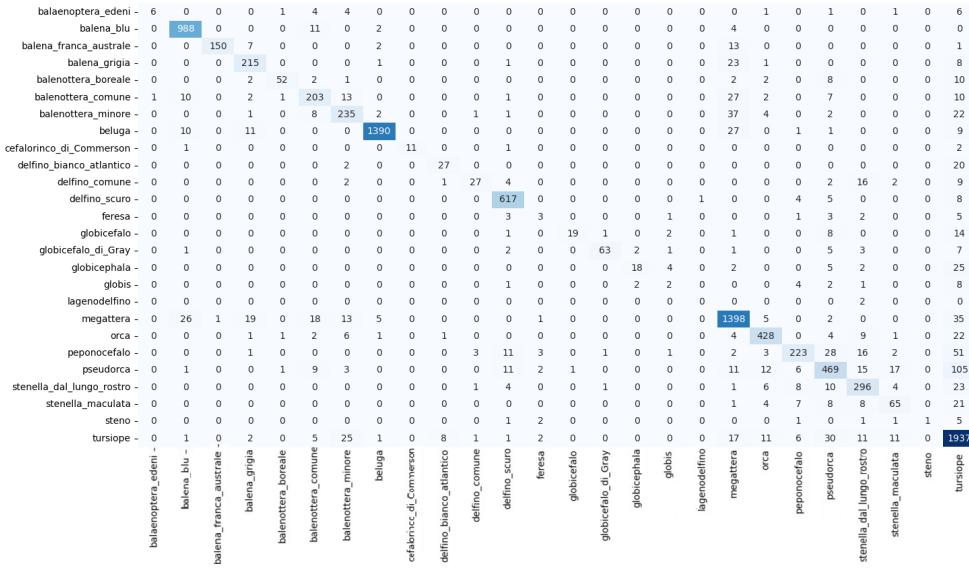


Figura 6.5: Matrice di confusione.

Visualizzazione della precisione per ogni specie

La precisione di ogni singola specie è riportata nella tabella 6.5, in aggiunta sono presenti anche altri indici spiegati in precedenza:

Tabella 6.5: Risultati delle misure di precisione, recall, F1-Score e support per ogni specie.

Specie	Precisione	Recall	F1-Score	Support
balaenoptera edeni	0.86	0.25	0.39	24
balena blu	0.95	0.98	0.97	1005
balena franca australis	0.99	0.87	0.93	173
balena grigia	0.82	0.86	0.84	251

Continua nella pagina successiva

Tabella 6.5 – Continuazione dalla pagina precedente

Specie	Precisione	Recall	F1-Score	Support
balenottera boreale	0.93	0.66	0.77	79
balenottera comune	0.76	0.72	0.74	281
balenottera minore	0.75	0.75	0.75	315
beluga	0.99	0.96	0.97	1449
cefalorinco Commer.	1.00	0.73	0.85	15
delfino b. atlant.	0.73	0.55	0.63	49
delfino comune	0.82	0.42	0.56	64
delfino scuro	0.93	0.97	0.95	635
feresa	0.23	0.17	0.19	18
globicefalo	0.95	0.41	0.58	46
globicefalo di Gray	0.95	0.74	0.83	85
globicephala	0.82	0.32	0.46	56
globis	0.18	0.10	0.13	20
lagenodelfino	0.00	0.00	0.00	2
megattera	0.88	0.92	0.90	1527
orca	0.89	0.89	0.89	480
peponocefalo	0.85	0.65	0.74	345
pseudorca	0.78	0.71	0.74	663
stenella lungo rostro	0.77	0.84	0.80	354
stenella maculata	0.62	0.57	0.59	114
steno	1.00	0.08	0.15	12
tursiope	0.82	0.94	0.87	2071
zifio	0.65	0.46	0.54	70
zifio2	0.00	0.00	0.00	1

Come si può vedere dalla tabella 6.5 ci sono valori molto contrastanti tra loro. In particolare si può notare che quando si hanno pochi esempi di training di una determinata specie la precisione risulta estremamente bassa. Oppure potrebbe capitare che alcuni individui molto simili tra loro vengano scambiati, questo è possibile vederlo dalla matrice di confusione . Un ulteriore aspetto che bisogna tenere in considerazione è la qualità delle immagini, la quale non sempre è ottimale, e questo porta l'algoritmo ad una maggiore possibilità di errore.

6.4.2 Fase di riconoscimento dell'immagine

In questa fase esamineremo il codice per identificare una nuova immagine che passeremo all'algoritmo addestrato al fine di riconoscere la specie. Per come è stata implementata, la rete neurale viene addestrata periodicamente in caso di aggiunta di nuove immagini, invece ogni volta che si vuole utilizzare l'algoritmo bisogna caricare un file nel quale è presente l'addestramento della rete.

Librerie utilizzate

Listing 6.18: Librerie per il riconoscimento dell'immagine.

```
1 from keras.models import load_model  
2 from keras.preprocessing.image import load_img,  
    img_to_array
```

Le librerie del codice 6.18 sono utilizzate per caricare un modello di rete neurale già addestrato (`load_model`) e per caricare un'immagine come input per la rete neurale (`load_img, img_to_array`). La libreria `load_img` consente di caricare un'immagine in memoria come oggetto PIL (Python Imaging

Library), mentre la libreria *img_to_array* converte l’oggetto PIL in un array NumPy che può essere utilizzato come input per la rete neurale.

Caricamento della rete neurale

Listing 6.19: Codice di caricamento della rete neurale.

```
1 model1 = load_model('./WCvF.h5', compile=False)
2 lab = train_images.class_indices
3 lab={k:v for v,k in lab.items() }
```

Il codice 6.19 carica il modello salvato precedentemente nel file WCvF.h5 (un file di dati binari che contiene un modello di rete neurale addestrato) e lo assegna ad una variabile `model1`. Successivamente, crea un dizionario `lab` che associa i valori numerici alle rispettive classi (in ordine alfabetico) estratte dal modello caricato. Il dizionario è stato creato invertendo la posizione di chiave e valore rispetto al dizionario originale `train_images.class_indices`. Col parametro `compile` uguale a `False` il modello non verrà compilato.

Funzione di predizione

Listing 6.20: Codice della funzione di predizione.

```
1 def output(location):
2     img=load_img(location,target_size=(224, 224, 3))
3     img=img_to_array(img)
4     img=img/255
5     img=np.expand_dims(img, [0])
6     answer=model1.predict(img)
7     y_class = answer.argmax(axis=-1)
```

```
8     y = " ".join(str(x) for x in y_class)
9     y = int(y)
10    res = lab[y]
11    return res
```

La funzione del codice 6.20 prende in input la posizione di un’immagine, la carica, la converte in un array di numpy, normalizza i valori dei pixel dell’immagine, aggiunge una dimensione per il batch e fa una previsione sulla classe dell’immagine usando un modello preaddestrato *modell*. Attraverso l’utilizzo del dizionario *lab* per mappare l’indice della classe al nome corrispondente viene restituita la sigla della specie predetta dell’immagine.

Predizione delle immagini

Listing 6.21: Codice della predizione delle immagini.

```
1 import os
2 folder_path = "./foto"
3 file_list = []
4 # Scandisce tutti i file nella cartella
5 for filename in os.listdir(folder_path):
6     file_path = os.path.join(folder_path, filename)
7     if os.path.isfile(file_path):
8         file_list.append(file_path)
9 for arg in file_list:
10    img=arg
11    pic=load_img(img, target_size=(224,224,3))
12    ris = ris + output(pic) + "-"
13 print(ris)
```

Dal codice 6.21 si può notare che le foto vengono passate dentro al contenitore Docker e inserite nella cartella relativa. Attraverso le funzioni di una libreria di sistema vengono catturate le immagini presenti e per ciascuna si elabora l'immagine utilizzando la funzione "output" che restituisce una stringa con il nome della classe predetta per quell'immagine. Alla fine, la funzione restituisce una stringa che contiene i nomi delle classi predette per tutte le immagini passate come argomenti, separate da "-".

6.5 Integrazione algoritmo di riconoscimento

Il sistema che permette di identificare la specie in un avvistamento è stato integrato con un bottone nella pagina di ogni singolo avvistamento. Quando questo viene premuto, viene aperto un modale e inviate delle sottoimmagini dell'avvistamento. Per la precisione, se sono presenti solo delle foto senza nessuna sottoimmagine, vengono inviate le foto intere, ma questo non permetterà un'alta precisione visto che il sistema è addestrato solo con singole immagini. Se invece l'utente ha creato delle sottoimmagini dell'avvistamento, queste sottoimmagini verranno ritagliate e inviate singolarmente all'algoritmo. Esso processerà tutte le immagini che gli sono state inviate e risponderà con le possibili soluzioni. Se invece non sono presenti immagini il sistema darà un errore.

Per utilizzare le librerie di machine learning citate sopra, il sistema è stato spostato su un server web dell'università con sistema operativo Linux, nel quale è stato installato Docker. Esso è una piattaforma open source che consente di creare e gestire applicazioni in modo indipendente dal sistema operativo sottostante. In pratica, Docker consente di creare degli ambienti virtuali isolati, che includono tutti i componenti necessari per far funzionare un'applicazione come librerie, dipendenze e configurazioni. Nel nostro

caso, è stato scelto di utilizzare un container con all'interno un sistema operativo Linux Ubuntu 18.04. In esso sono stati installati la versione specifica di Python 3.8, e i pacchetti necessari per eseguire un particolare progetto, senza influire sulle altre installazioni presenti sul sistema operativo.

Quindi, all'apertura del modale attraverso l'esecuzione del comando *exec* nel file PHP, viene aperto un terminale a cui viene passato il nome della cartella contenente le immagini. Esso, a sua volta, attiverà il contenitore Docker passandogli i file e farà partire lo script 6.22.

Listing 6.22: Codice script bash.

```
1 #!/bin/bash
2 cd ../../../../ric
3 percorso="/var/www/html/Sito/img/temp/"$@
4 echo $percorso
5 docker run -v ${percorso}:/Ric/foto --rm ric_sea
```

Capitolo 7

Test dell'applicazione

7.1 Test con l'utente

Al fine di effettuare operazioni di test direttamente con l'utente, si è deciso di creare un prototipo per far interagire direttamente lui sull'intero sistema in beta-release. Gli aspetti testati sono:

- Il modello concettuale è sufficientemente rappresentato.
- Rispetto al progetto, l'interfaccia è adatta e sono stati rispettati tutti gli standard.
- Possibilità di uso alternativo tra mouse e tastiera.
- Livello di interazione tra l'utente e l'interfaccia.
- Adeguato bilanciamento tra flusso predefinito e flessibilità.

7.1.1 Verifica del software

Questa fase ha lo scopo di verificare che il sistema contenga le specifiche di progetto. Tale operazione non viene eseguita solo sul prodotto finale, ma segue il progetto ad ogni suo passo. Le tecniche di verifica che sono state utilizzate sono:

- **Di testing:** che, attraverso delle prove sperimentali, su un insieme rappresentativo di situazioni, verificano il corretto funzionamento del sistema.
- **Di analisi:** attraverso l'analisi della struttura dei moduli e del codice che li realizza viene verificato il corretto funzionamento.

Testing

”Le operazioni di testing possono individuare la presenza di errori nel software ma non possono dimostrarne la correttezza.”[45]

Come primo passo, si sono individuati i casi significativi in cui applicare tale processo.

Le operazioni di testing si suddividono in:

1. **Testing in the small:** Riguardano porzioni specifiche di codice a cui è stata attribuita una particolare importanza.
2. **Testing in the large:** riguardano tutto il sistema.

Nel primo caso si è cercato di seguire il seguente criterio di copertura:

Criterio di copertura delle decisioni e delle condizioni: selezionare un insieme di test concreti tali che, a seguito dell'esecuzione del programma su tutti i casi di test, ogni arco del grafo di controllo sia attraversato e tutti

i possibili valori delle condizioni composte siano valutati almeno una volta. Nel secondo caso, il sistema viene considerato come una black-box. L'insieme di test che verranno usati vengono selezionati sulla base delle specifiche di progetto, e grazie ad esse definiti i valori di input a cui corrisponderanno determinati valori di output. I casi di testing in the large presi in considerazione sono:

- **Test di modulo:** controlla se l'implementazione di un modulo è corretta in base al comportamento esterno.
- **Test d'integrazione:** sottoparti del sistema vengono verificate sulla base del loro comportamento esterno.
- **Test di sistema:** controllo il comportamento dell'intero sistema sulla base del suo comportamento esterno.

7.1.2 Analisi

Analizzare il software consiste nel capire le caratteristiche e le funzionalità. L'approccio per l'analisi del software è stata la Code Inspection. In particolare si è posta molta attenzione all'analisi di flusso dei dati. Infatti l'evoluzione del valore associato alle variabili durante l'esecuzione del programma è estremamente dinamica. Quindi si è usata la tecnica di associare staticamente ad ogni comando il tipo di operazioni eseguite sulle variabili. In questo modo, si sono ottenute delle sequenze di comandi corrispondenti a possibili esecuzioni, che sono riconducibili staticamente a sequenze di tali operazioni. Queste ultime verranno analizzate al fine di individuare delle possibili anomalie nel programma.

7.1.3 Test

È stato eseguito sia l'alpha testing (cioè una procedura tramite la quale un software viene testato internamente dallo sviluppatore o dal team di sviluppo), sia il beta testing che invece si riferisce alla fase in cui il software viene testato esternamente dagli utenti prima della sua distribuzione pubblica. Nel primo caso, si è cercato di migliorare al massimo le funzionalità del programma e cercato di risolvere tutti gli eventuali problemi che si sono presentati. Invece nel secondo, il sistema, una volta installato, è stato testato direttamente dagli utenti finali, in modo da poter verificare che tutte le funzionalità fossero presenti e che fossero state implementate correttamente.

Capitolo 8

Conclusioni

Alla fine del mio percorso di studi, ritengo che il progetto che ho appena completato sia stato uno dei più complessi e soddisfacenti tra quelli avviati negli anni. Inizialmente, si trattava solo di un piccolo progetto basato su un mini-server contenente il database e le pagine web. Tuttavia, ho avuto l'opportunità di espandere il progetto in modo significativo, fino a includere un sistema di riconoscimento tramite visione artificiale, librerie e codice per il server web, oltre ad una solida infrastruttura software. Alla fine, il sistema è stato spostato nei server web dell'università dove potrà essere utilizzato da tutti i ricercatori e dai pescatori al fine di documentare gli avvistamenti delle varie specie marine.

Questo progetto mi ha dato l'opportunità di crescere e migliorare le mie abilità: sono riuscito a sviluppare un sistema altamente ampliabile, con numerose funzionalità e applicazioni mobili e web. Ad esempio, l'applicazione mobile permette di caricare gli avvistamenti delle varie specie marine, mentre la pagina web offre la possibilità di caricare, modificare ed elimina-

re gli avvistamenti. Inoltre, grazie all'integrazione di un sistema di machine learning, il progetto permette di riconoscere con una discreta precisione la specie dei delfini.

Inoltre, l'infrastruttura è altamente ampliabile e può essere utilizzata per eseguire ricerche in vari ambiti. Ad esempio, si pensava di estendere l'utilizzo della visione artificiale sulla funzione di riconoscimento di individui già avvistati in precedenza oppure di estendere l'applicazione mobile attraverso delle nuove features e aggiungere nuovi dati statistici. Invece, nell'applicazione web si potrebbe implementare un servizio di messaggistica in modo che gli utenti possano scambiarsi ulteriori informazioni oppure consultarsi reciprocamente.

Il mio percorso universitario mi ha dato l'opportunità di imparare molte cose e di crescere come persona e come professionista. Sono grato per tutte le esperienze che ho avuto durante gli anni di studio, che mi hanno permesso di diventare la persona che sono oggi. Credo che il progetto che ho appena completato sia una testimonianza del mio impegno e della mia passione per la tecnologia e la ricerca. Spero che possa essere utilizzato e apprezzato da molti ricercatori e pescatori in tutto il mondo.

Bibliografia

- [1] LAMP Applications, Ubuntu server, ubuntu.com
- [2] Apache, Apache HTTP Server Documentation,
httpd.apache.org/docs
- [3] MySQL, dev.mysql.com/doc/
- [4] PHP, www.php.net/docs.php
- [5] Jason Potter, Featured Articles, www.liquidweb.com, March 3 2021
- [6] HTML, HTML Tutorial, www.w3schools.com/html/default.asp
- [7] Certbot, eff-certbot.readthedocs.io/en/stable/
- [8] Let's encrypt, letsencrypt.org/docs/
- [9] CSS, CSS Tutorial, www.w3schools.com/css/default.asp
- [10] Bootstrap, Introduction, getbootstrap.com
- [11] JavaScript, Introduction, getbootstrap.com
- [12] jQuery, jQuery API, api.jquery.com

- [13] javascript-coder, Reintroduction To AJAX,
www.javascript-coder.com.
- [14] AJAX, AJAX Introduction, www.w3schools.com/xml/ajax_intro.asp
- [15] JSON, Introducing JSON, www.json.org/json-en.html
- [16] PHPmailer, github.com/PHPMailer/PHPMailer
- [17] Leaflet, leafletjs.com/reference.html
- [18] Jason Jung, Password Hashing and Salting, www.okta.com, May 7
2021
- [19] W3C, W3C standards and drafts, www.w3.org/TR/
- [20] WCAG, The WCAG 2 Documents, www.w3.org/WAI/standards-guidelines/wcag/docs/
- [21] Antonio Giovanni Schiavone, Accessibilità,
www.antoniogiovannischiavone.it, 1 Settembre 2016.
- [22] Android, Guide per gli sviluppatori,
developer.android.com/docs?hl=it
- [23] Kotlin, Kotlin docs, kotlinlang.org/docs/home.html
- [24] Jetpack Compose, m3.material.io/develop/android/jetpack-compose
- [25] Material Design, m3.material.io
- [26] Java, Java documentation, docs.oracle.com/en/java/
- [27] Biometric, developer.android.com/jetpack/androidx/releases/biometric

- [28] Room, developer.android.com/jetpack/androidx/releases/room
- [29] Okhttp3, Package Okhttp3, java-doc.io/doc/com.squareup.okhttp3(okhttp/3.14.9(okhttp3/package-summary.html)
- [30] Castle, An Introduction to Castle, docs.castle.io/docs
- [31] Gson, Class GSON, www.javadoc.io/doc/com.google.code.gson/gson/2.8.0/com/google/gson/Gson.html
- [32] Coil, coil-kt.github.io/coil/
- [33] Tehras-charts, github.com/tehras/charts
- [34] Renò, V.; Losapio, G.; Forenza, F.; Politi, T.; Stella, E.; Fanizza, C.; Hartman, K.; Carlucci, R.; Dimauro, G.; Maglietta, R. Combined Color Semantics and Deep Learning for the Automatic Detection of Dolphin Dorsal Fins. *Electronics* 2020, 9, 758. <https://doi.org/10.3390/electronics9050758>
- [35] MobileNetV2, Google, TensorFlow, gitHub.com/tensorflow/models/tree/master/research/slim/nets/mobilenet
- [36] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. Stanford University. image-net.org/
- [37] Happywhale. (2022). Happywhale - Whale and Dolphin Identification. Kaggle. www.kaggle.com/competitions/happy-whale-and-dolphin/data
- [38] NumPy, NumPy documentation, numpy.org/doc/

- [39] Matplotlib, Matplotlib 3.7.1 documentation, matplotlib.org/stable/index.html
- [40] Pathlib, Pathlib 3.4 documentation, docs.python.org/3/library/pathlib.html
- [41] pandas, pandas documentation, pandas.pydata.org/docs/
- [42] seaborn, seaborn: statistical data visualization, seaborn.pydata.org
- [43] TensorFlow, API Documentation, www.tensorflow.org/api_docs
- [44] scikit-learn, scikit-learn.org/stable
- [45] Khan, K. & Yadav, A Literature Review on Software Testing Techniques, S. 2022.