

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

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

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

Il dipartimento di Acquacoltura sta cercando di monitorare gli avvistamenti di diverse specie marine nel nostro territorio, in particolare necessita di un'applicazione che permetta di controllare tutti gli avvistamenti che i ricercatori hanno effettuato durante le loro uscite in mare.

L'obiettivo di questo studio è quello di poter aiutare i ricercatori nel loro lavoro, in particolare nel loro compito di monitoraggio al fine di comprendere al meglio il comportamento degli individui con lo scopo di individuare delle soluzioni ambientali affinché questi esemplari non si sentano minacciati nei nostri mari.

Per riuscire in questo compito si ha la necessità di un'intera infrastruttura, in primis una banca dati al fine del mantenimento degli stessi, e poi un portale, il quale permetta la registrazione di nuovi avvistamenti, ma anche la modifica e la cancellazione. Per ogni avvistamento si ha anche la possibilità di caricare immagini e individuare esemplari, i quali possono avere diverse ferite. Durante la modifica o l'inserimento dei dati ai ricercatori è stata proposta la possibilità dell'utilizzo di una tecnica di riconoscimento automatico al fine di individuare più velocemente la specie di appartenenza.

Tale applicazione può essere utilizzata non solo per scopi di ricerca, ma anche al fine di creare una grande comunità, la quale abbia la possibilità di scambiarsi informazioni e immagini vicendevolmente.

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 la quale mi ha sempre aiutato con un sorriso durante la stesura del seguente elaborato. Tutto è partito dalle sue lezioni in cui grazie alla sua felicità e tranquillità è riuscita a creare molta empatia con gli studenti. Per questo ho deciso di scegliere lei come 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 voi.

Ringrazio tutti i miei amici che mi sono stati vicino, che mi hanno supportato e anche chi mi è stato affianco durante le mie idee, molte volte irrealizzabili.

In particolare ringrazio Fabio, amico di studi, amico di vita, amico di avventure: senza di lui non sarei qui a scrivere queste frasi. Abbiamo percorso tutto questo percorso insieme, sostenendoci, supportandoci e sopportandoci a vicenda. Credo sia stata una meravigliosa esperienza e questo credo sia grazie anche a te.

Infine, dedico questa tesi a me stesso, perché non pensavo minimamente di riuscire a diventare quello che sono oggi e grazie a tutto questo percorso riesco ad avere più fiducia in me stesso. Posso dire di essere estremamente soddisfatto di questa avventura.

Indice

1	Introduzione	8
1.1	Diagramma dei casi d'uso	9
2	Documentazione Base di dati	10
2.1	Analisi dei requisiti	10
2.1.1	Intervista	10
2.1.2	Rilevamento delle ambiguità e correzioni proposte	12
2.1.3	Estrazione dei concetti principali	14
2.2	Progettazione Concettuale	15
2.2.1	Schema scheletro	15
2.2.2	Raffinamenti attuati	18
2.2.3	Schema concettuale finale	18
2.3	Progettazione Logica	19
2.3.1	Stima del volume dei dati	20
2.3.2	Descrizione delle operazioni principali e stima della loro frequenza	20
2.3.3	Schemi di navigazione e tabelle degli accessi . . .	21
2.3.4	Raffinamento dello schema	25
2.3.5	Traduzione di entità e associazioni in relazioni . .	25

2.3.6	Schema relazionale finale	27
3	Documentazione Applicazione Mobile	28
4	Documentazione Interfaccia Web	29
4.1	Progettazione	29
4.1.1	Usabilità	30
4.1.2	Design	31
4.1.3	Solution stack	36
4.1.4	Fruizione del servizio	38
4.1.5	Stile	40
4.1.6	Javascript e Librerire	41
4.1.7	Cookies	44
4.1.8	Sicurezza	46
4.2	Accessibilità	48
4.2.1	WCAG	48
4.3	Struttura del progetto	52
4.3.1	Pagina di login	54
4.3.2	Pagina di sign Up	55
4.3.3	Pagina della privacy	56
4.3.4	Pagina principale	57
4.3.5	Pagina delle impostazioni	59
4.3.6	Pagina degli avvistamenti	60
5	Documentazione Algoritmo Di Riconoscimento	66
5.1	Il problema	66
5.2	Dataset	67
5.2.1	Specie presenti e quantità di immagini	68
5.3	Algoritmo di riconoscimento	69

5.3.1	Fase di training	70
5.3.2	Fase di riconoscimento dell'immagine	96
5.4	Integrazione dell'algoritmo col sistema	98
6	Documentazione Test	99
6.1	Test con l'utente	99
7	Conclusioni	100
8	Utilità	101
8.1	Ingegneria del software	101
8.1.1	Verifica del software	101
8.1.2	Analisi	103

Elenco delle figure

1.1	Schema dei casi d'uso per la piattaforma.	9
2.1	Schema concettuale per la modellazione degli avvistamenti e delle proprietà dell'utente.	15
2.2	Modellazione dell'attribuzione dell'animale e della specie all' avvistamento.	16
2.3	Schema concettuale raffigurante le immagini con relative sottoimmagini e la gestione delle ferite.	17
2.4	Schema concettuale finale	19
2.5	Schema concettuale raffigurante gli accessi alle tabelle per l'aggiunta di un nuovo avvistamento.	23
2.6	Schema logico finale.	27
4.1	Mockup homepage del sito in versione mobile.	32
4.2	Mockup homepage del sito in versione desktop.	33
4.3	Mockup avvistamento del sito in versione mobile.	34
4.4	Mockup avvistamento del sito in versione desktop.	35
4.5	Stack LAMP. Fonte: phoenixnap.com	37
4.6	Tecnologie alla base del web.	38
4.7	Funzionamento di AJAX. Fonte: www.javascript-coder.com	42
4.8	Struttura alla base della fase di criptaggio. Fonte: www.okta.com	47

4.9	Struttura delle linee guide coi rispettivi livelli. Fonte: www.antoniogiovannischiavone.it	49
4.10	Struttura del progetto.	52
4.11	Struttura della pagina di Login.	54
4.12	Modale per il recupero della password.	54
4.13	Struttura della pagina di Sign-Up.	55
4.14	Struttura della pagina dell'informativa sulla privacy.	56
4.15	Struttura della pagina principale del sito.	57
4.16	Modale per l'inserimento di un nuovo avvistamento.	58
4.17	Pagina delle impostazioni del profilo utente.	59
4.18	Pagina delle impostazioni sulla sicurezza in modalità mobile.	60
4.19	Prima parte della pagina dell'avvistamento, modifica dei dati.	61
4.20	Modale di eliminazione.	62
4.21	Modale di informazione sulla specie.	62
4.22	Seconda parte della pagina dell'avvistamento con caratteristiche degli esemplari.	63
4.23	Modale con canvas per disegnare l'individuo.	64
4.24	Modifica dell'esemplare.	65
4.25	Modale per l'aggiunta e modifica della ferita.	65
5.1	Distribuzione delle categorie nella colonna species.	75
5.2	Perdita(loss) ottenuta dal modello durante l'addestramento(blu) e perdita ottenuta dal modello durante la validazione(arancio).	89
5.3	L'accuratezza ottenuta dal modello durante l'addestramento(blu) e l'accuratezza ottenuta dal modello durante la validazione(arancio).	90
5.4	Matrice di confusione.	94

Capitolo 1

Introduzione

Durante il percorso accademico mi è stata proposta la creazione di una piattaforma per la gestione degli avvistamenti di particolari specie marine. Tale proposta è stata presa in considerazione al fine 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 corso universitario con lo scopo di creare un'intera infrastruttura centralizzata.

Il sistema si suddivide in diverse parti, tra cui:

- 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 riconoscere il singolo individuo per poterne eventualmente tracciare gli spostamenti.

1.1 Diagramma dei casi d'uso

E' stato riportato un diagramma dei casi d'uso il quale descrive le principali operazioni del sistema:

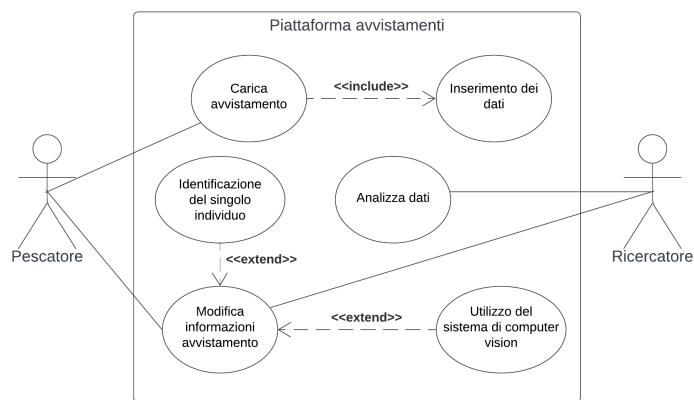


Figura 1.1: Schema dei casi d'uso per la piattaforma.

Capitolo 2

Documentazione Base di dati

2.1 Analisi dei requisiti

In questa sezione analizzeremo tutte le specifiche che il committente richiede per la piattaforma.

2.1.1 Intervista

Di seguito si riporta la prima intervista col committente: "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 durante le uscite a largo dalla costa.

Ogni utente, il quale ha a disposizione uno smartphone, una volta effettuato l'accesso con login e password, ha la possibilità di inserire un nuovo avvistamento, per il quale gli elementi da memorizzare sono la data e ora corrente (inserite automaticamente dal sistema), il numero di esemplari che sono stati avvistati e opzionalmente anche il vento in km/h, le condizioni del mare, eventuali ferite visibili. In particolare, si ponga attenzione alla

descrizione della ferita, la sua posizione e la relativa gravità. Deve anche essere possibile scrivere note che possano aiutare ad individuare la specie e/o elementi che aggiungano informazioni all'avvistamento.

Un avvistamento può far riferimento ad un particolare animale, a una specie specifica o anche a nessuna delle due: il terzo caso viene preso in considerazione quando chi sta eseguendo la rilevazione non ha la possibilità di riconoscere la specie esatta, inseribile comunque in un secondo momento.

Oltre all'applicazione smartphone deve essere realizzato un applicativo web il quale, una volta effettuato l'accesso, deve poter visualizzare tutti gli avvistamenti che sono stati effettuati dagli utenti e poterne 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 può 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 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 vedere la sua evoluzione nel tempo e i suoi aspettamenti.

Per aiutare l'utente nella decisione della specie da assegnare a un particolare animale, il sistema deve 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 in auto-

nomia di identificare la relativa specie attraverso la foto scattata durante il rilevamento. Nello specifico deve cercare di aiutare l’utente nella scelta della specie, stilando una classifica di compatibilità con le specie e gli individui che ha presenti in database.

Un aspetto importante da tenere in considerazione è il fatto che il sistema in autonomia, siccome se ci si trova a largo della costa non è presente connessione, 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. 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, i quali saranno considerati al fine della progettazione concettuale:

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

2.1.3 Estrazione dei concetti principali

Dall'intervista si ricavano anche le operazioni principali richieste:

- Creazione di un nuovo utente.
- Registrazione di un nuovo avvistamento attraverso l'applicazione mobile.
- Inserimento 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, e ciascuna delle quali a seconda dell'animale analizzarla attraverso un sistema di riconoscimento della specie.
- Monitorare gli avvistamenti di particolari individui al fine di 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, compreso 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 parzialmente nella formalizzazione dei requisiti, in particolare sui termini e sulle transazioni, i quali sono stati raccolti e analizzati nella fase precedente.

2.2.1 Schema scheletro

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

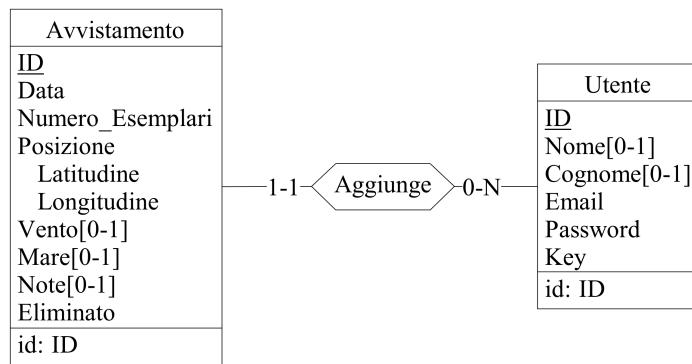


Figura 2.1: Schema concettuale per la modellazione degli avvistamenti e delle proprietà dell’utente.

Come mostrato in figura sono presenti uno o più utenti, ognuno dei quali accedono alla piattaforma attraverso la propria email, password e grazie a una chiave personale univoca all’interno del database, la quale viene utilizzata al fine di criptare la password. Di tale utente possiamo tenere anche in considerazione il nome e cognome. Ogni utente aggiunge degli avvistamenti, ognuno dei quali contiene tutti i dati richiesti del committente tra cui la data di avvistamento, il numero di esemplari presenti e opzionalmente anche il

vento espresso in km/h, lo stato del mare ed eventuali note informative al fine di ricerca e in particolare per riuscire ad identificare meglio la specie. In aggiunta per ogni avvistamento è associata una posizione la quale tiene in considerazione la longitudine e la latitudine al momento dell'inserimento.

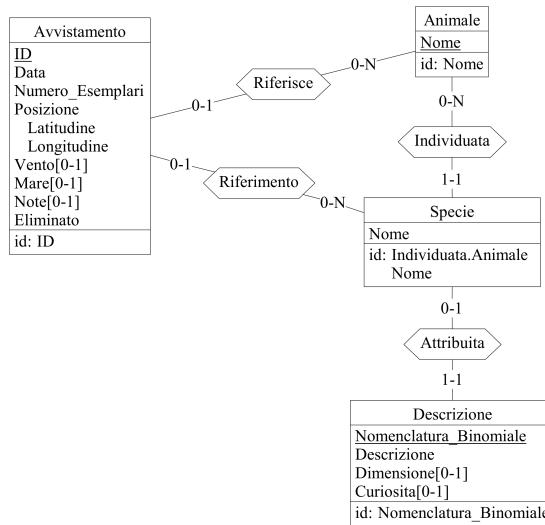


Figura 2.2: Modellazione dell'attribuzione dell'animale e della specie all'avvistamento.

Dalla figura 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. Al fine di individuare al meglio la specie ad ognuna 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 che se dichiaro una determinata specie essa deve fare riferimento al giusto animale. Un ulteriore vincolo da rispettare è il fatto che in un avvistamento, anche se esso include più esemplari, questi devono far riferimento allo stesso animale ed eventualmente alla relativa specie, per segnalare la presenza di specie diverse sarà quindi necessario inserire più avvistamenti.

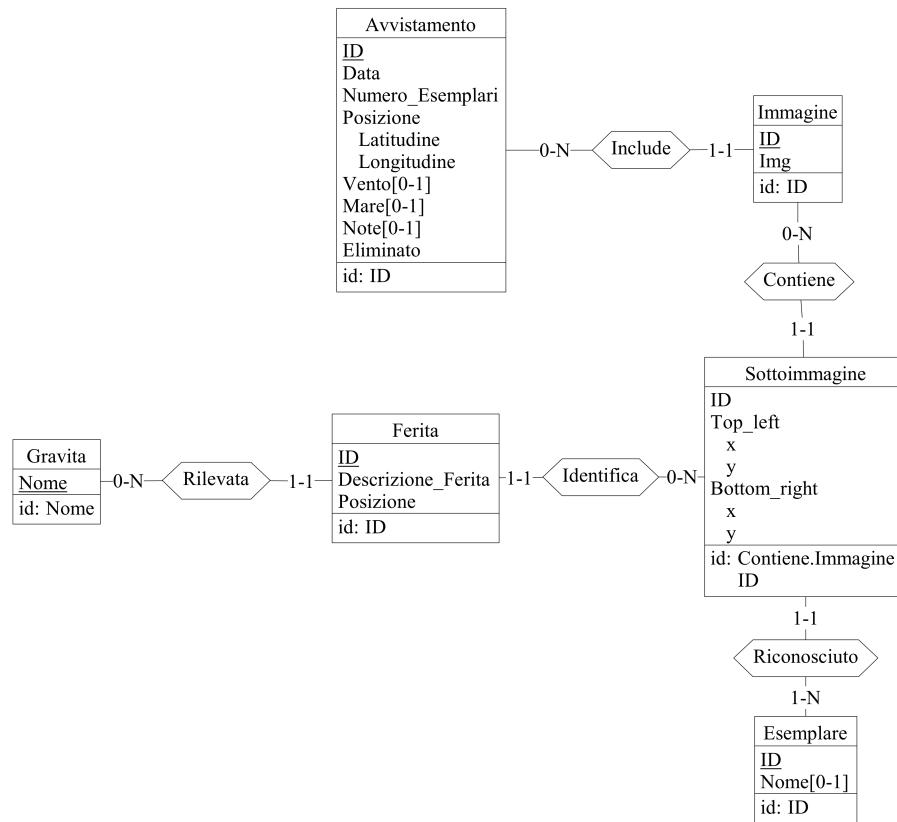


Figura 2.3: Schema concettuale raffigurante le immagini con relative sotimmagini e la gestione delle ferite.

Navigando lo schema partendo dall'entità Avvistamento, si nota che per

ogni avvistamento si possono includere una o più immagini le quali possono 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 esso presenta delle ferite queste possono essere inserite e ad ognuna attribuita una gravità. In aggiunta il singolo individuo viene catalogato, con un id e optionalmente 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 al fine di non appesantire troppo l'entità Specie con attributi optionali, i quali come si vede dallo schema rappresentano alla fine una sezione a parte del dominio.

2.2.3 Schema concettuale finale

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

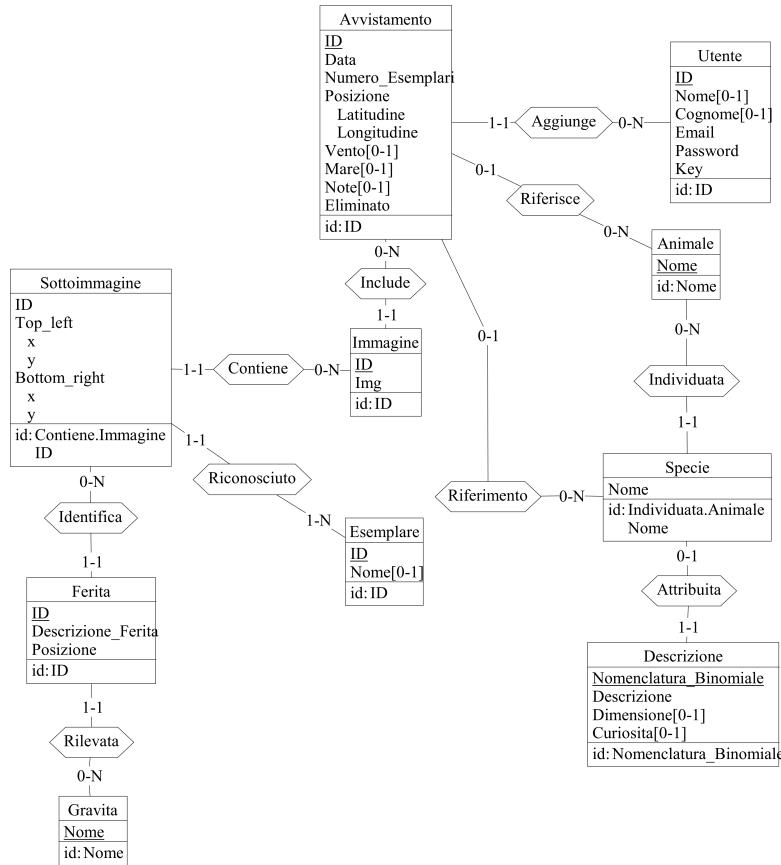


Figura 2.4: Schema concettuale finale

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 di seguito è riportato il volume atteso per ciascun costrutto presente nello schema concettuale. Inoltre, 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	z
Specie	E	x
Descrizione	E	x-y

2.3.2 Descrizione delle operazioni principali e stima della loro frequenza

Di seguito si riporta una tabella 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

2.3.3 Schemi di navigazione e tabelle degli accessi

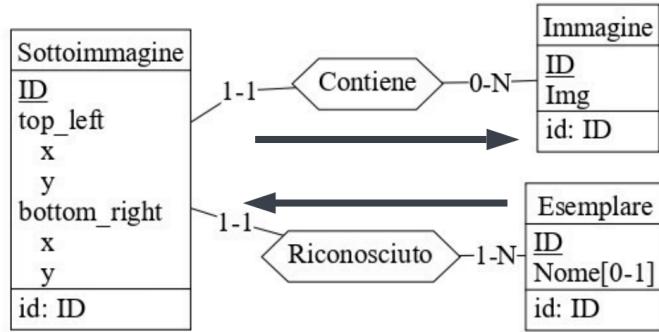
Dopo aver stimato i volumi dei principali costrutti presenti nella base di dati e la frequenza delle principali operazioni, si può procedere a disegnare i relativi schemi di navigazione e scrivere le tabelle degli accessi.

1. Creazione di un nuovo utente:

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

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

2. Monitoraggio di specifici individui al fine di determinate verifiche:



Per svolgere questa operazione 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 dovranno 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			

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

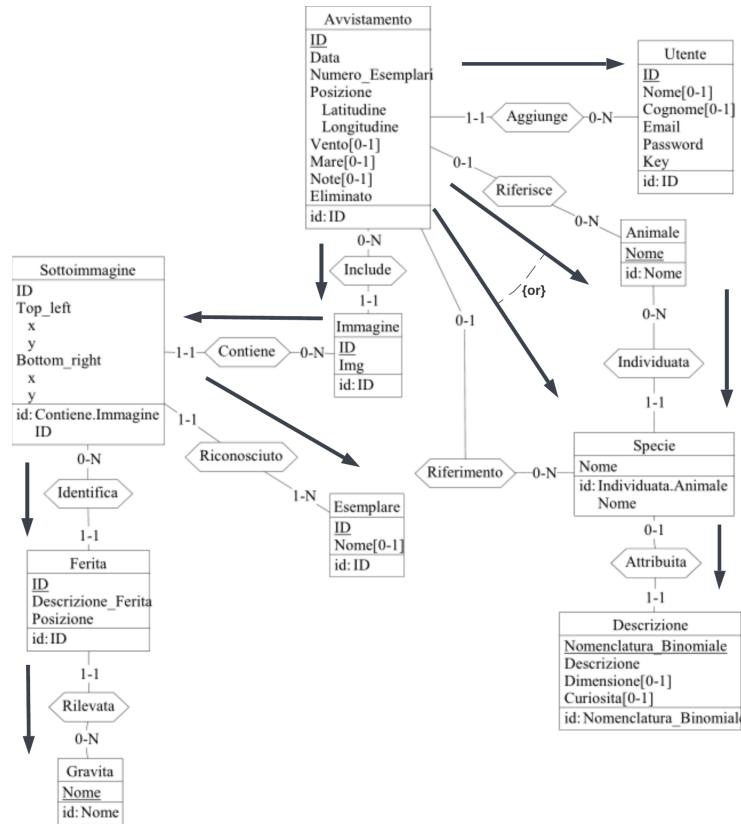


Figura 2.5: Schema concettuale raffigurante gli accessi alle tabelle per l'aggiunta di un nuovo avvistamento.

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

ferita è obbligatorio associargli una gravità. Nella tabella degli accessi sottostante 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		

Accessi totali

Si riporta di seguito la tabella 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

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 singole 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:Sottoimmagini, Img_rif:Immagini)

Sottoimmagini(ID, tl_x, tl_y, br_x, br_y, Immag_ID:Immagini,
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

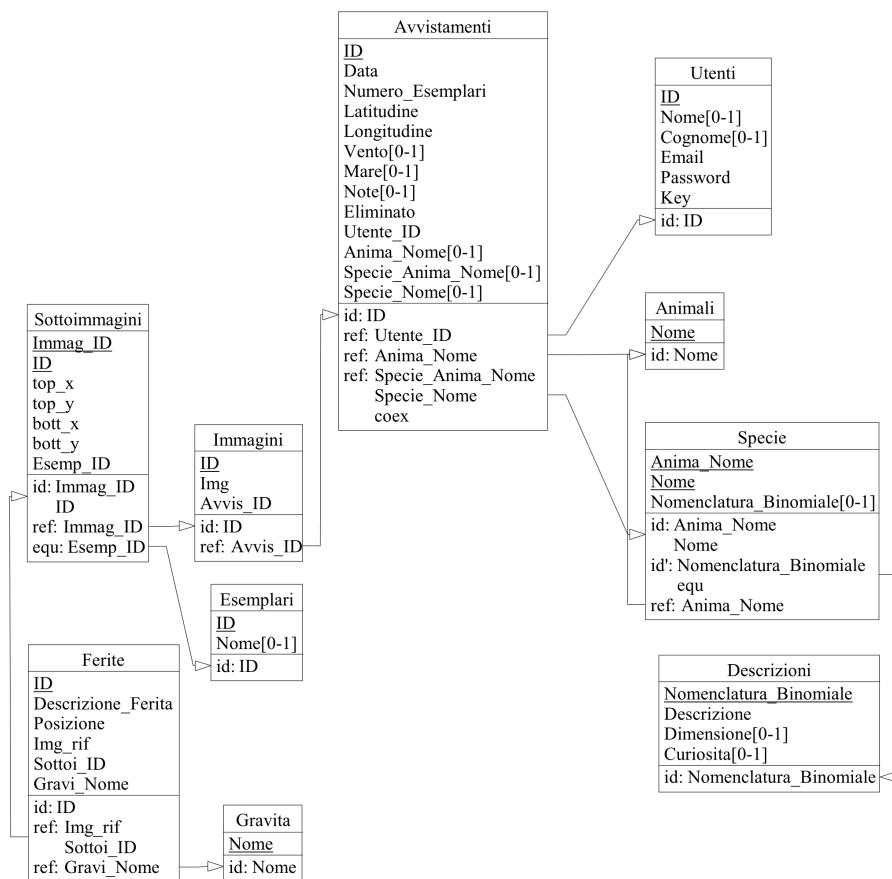


Figura 2.6: Schema logico finale.

Capitolo 3

Documentazione Applicazione Mobile

Capitolo 4

Documentazione 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 puntamento.
- **Metafora:** una parola, una frase o una figura la quale 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, i quali saranno gli utilizzatori del programma ed a ognuno di loro è stato associato uno scenario, il quale rappresenta le operazioni, il modo di muoversi nel sito e le loro modalità di utilizzo della piattaforma. Le personas che sono state riscontrate sono:

- Responsabile avvistamenti: persona addetta alla gestione delle informazioni al fine di modificare, aggiungere o eliminare i dati.

- Pescatore: persona che carica nuove informazioni, di un nuovo avvistamento direttamente dal portale.

A ciascuna di esse è associato uno scenarios:

- 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 è stata scelta la migliore strategia.
- Mockup: sulla base delle informazioni sono stati creati degli schemi base i quali 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 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 operazioni chiave. Il primo mockup che viene mostrato è quello della pagina principale in modalità mobile:

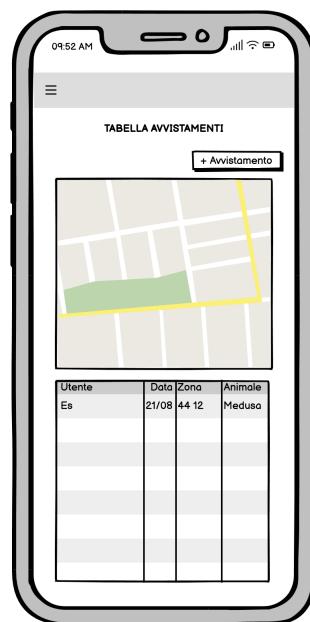


Figura 4.1: Mockup homepage del sito in versione mobile.

Da come si può vedere dall'immagine, si tratta di un design molto semplice

in cui è presente 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 viene invece mostrata la stessa pagina web in versione desktop:

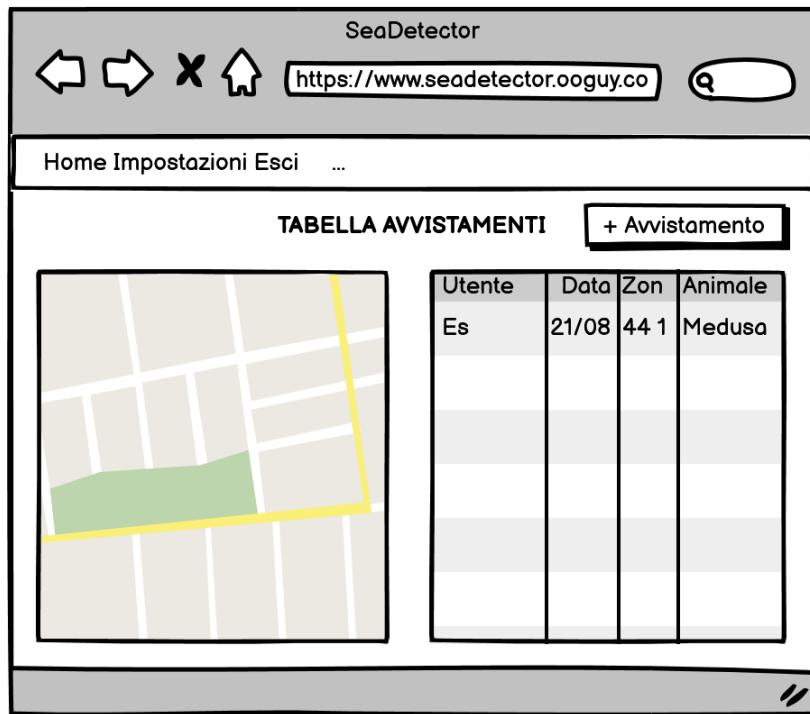


Figura 4.2: Mockup homepage del sito in versione desktop.

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

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

La pagina web contiene una semplice mappa in cui viene raffigurato il lu-

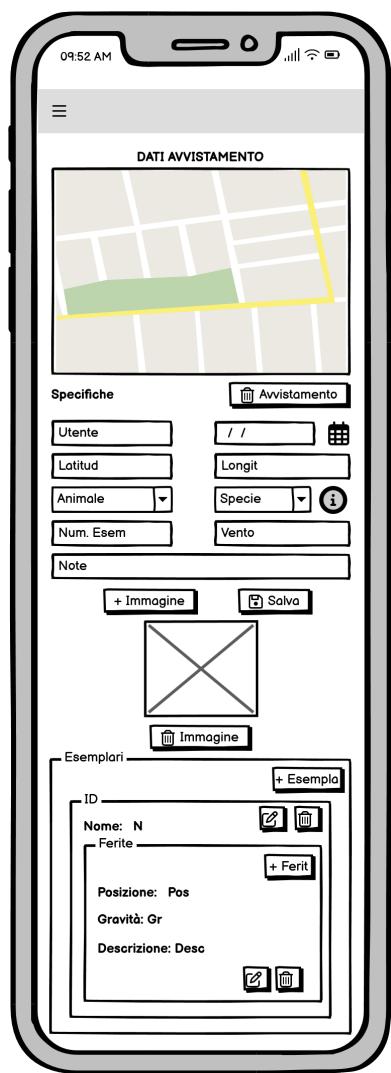


Figura 4.3: Mockup avvistamento del sito in versione mobile.

go dell'avvistamento e per ognuno vengono mostrati i relativi dati, i quali 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 aggiunte o eliminate le ferite.

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

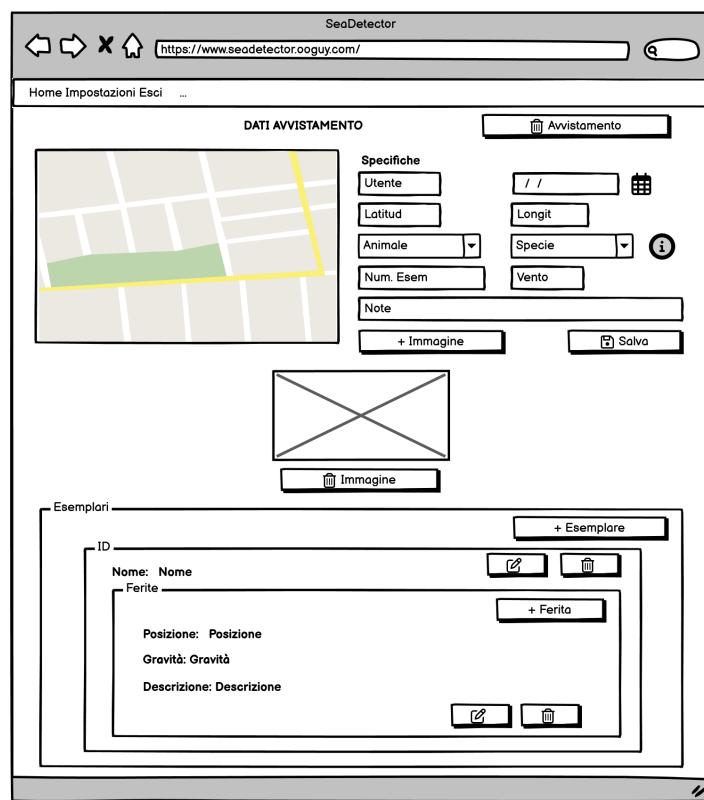


Figura 4.4: Mockup avvistamento del sito in versione desktop.

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 (Linux, Apache, MySQL, PHP), il quale è un popolare solution stack per lo sviluppo di applicazioni web su sistema operativo Linux. Questo perchè il database e il server web Apache2 sono situati in una Raspberry pi 4, la quale ha un indirizzo di rete fisso a cui chiunque collegato via internet può connettersi ed accedere al sito e ai suoi contenuti previa registrazione.

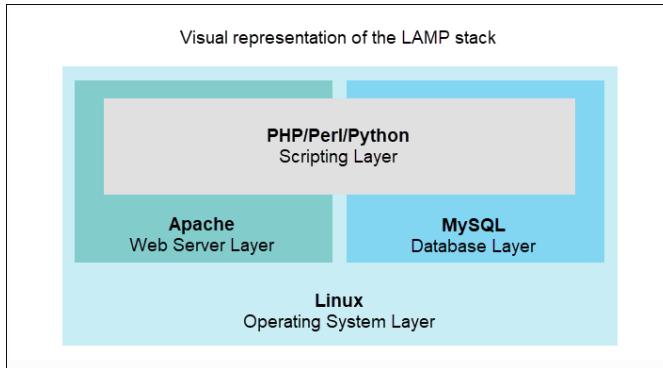


Figura 4.5: Stack LAMP. Fonte: phoenixnap.com

PHP

Lato 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. Questo rende 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, la connessione a database, l'elaborazione di cookie e sessioni e la gestione dei file. Questo rende possibile creare siti web che 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.

Questa tecnologia 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, è fruibile utilizzando le tre tecnologie che sono alla base del web.

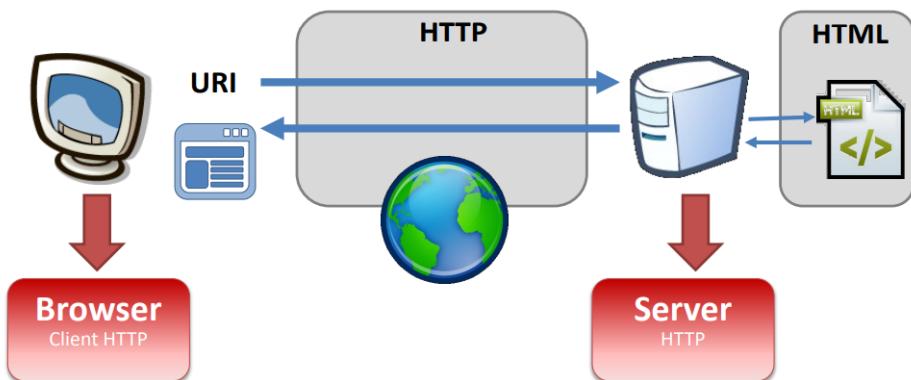


Figura 4.6: Tecnologie alla base del web.

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 www.seadetector.ooguy.com 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 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 sulla Raspberry pi4 è stato utilizzato un software denominato Certbot il quale 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 sviluppatori non devono preoccuparsi di rinnovare manualmente i certificati. Certbot permette di ottenere certificati gratuiti tramite Let's Encrypt, 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) 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.

namento del sito. CSS viene utilizzato per la creazione di layout complessi, animazioni, transizioni e effetti grafici.

Bootstrap

Bootstrap 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 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 e questo 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 Librerire

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, i quali 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 crezione di determinate funzionalità, è stata utilizzata una libreria JavaScript open source utilizzata per rendere più semplice e veloce lo sviluppo di pagine interattive e applicazioni web. Essa è denominata JQUERY 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

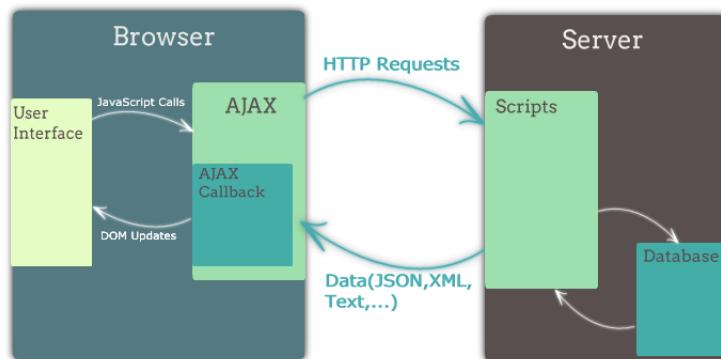


Figura 4.7: Funzionamento di AJAX. Fonte: www.javascript-coder.com

Nell'applicazione per richiedere dati dal database vengono eseguite delle chiamate Ajax asincrone al server. AJAX (Asynchronous JavaScript and XML) è una tecnologia utilizzata per creare applicazioni web dinamiche e interattive. Esso si basa sull'utilizzo di JavaScript e permette di aggiornare parti di una pagina web senza dover ricaricarla. 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. Questo 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.

PHPMailer

Nel sito, al fine di recuperare la password nel caso qualcuno se 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. 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.

Al fine di non dover comprare un dominio per le mail e dover gestire il server mail è stato utilizzato il servizio di posta elettronica di Libero.

LeafLet

Leaflet 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 altri elementi geografici. La libreria supporta una vasta gamma di formati di dati geografici. Leaflet offre la possibilità di creare mappe personalizzate utilizzando fogli di stile. La libreria è supportata da una vasta gamma di dispositivi, tra cui desktop, tablet e smartphone, rendendola una soluzione versatile e scalabile. In particolare è utile nella nostra piattaforma per evidenziare i punti in cui sono stati effettuati gli avvistamenti.

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 degli utenti. 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 consentirgli 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. Quindi secondo la normativa, i cookies tecnici possono essere usati anche senza chiedere il consenso dell’utente. Al fine dell’applicazione si è 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 che 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:

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. Avendo criptato la password e la chiave, insieme a tutti gli altri dati utente (non criptati), essi verranno salvati nel database.
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 uguaglianza permetterà l’accesso al servizio.

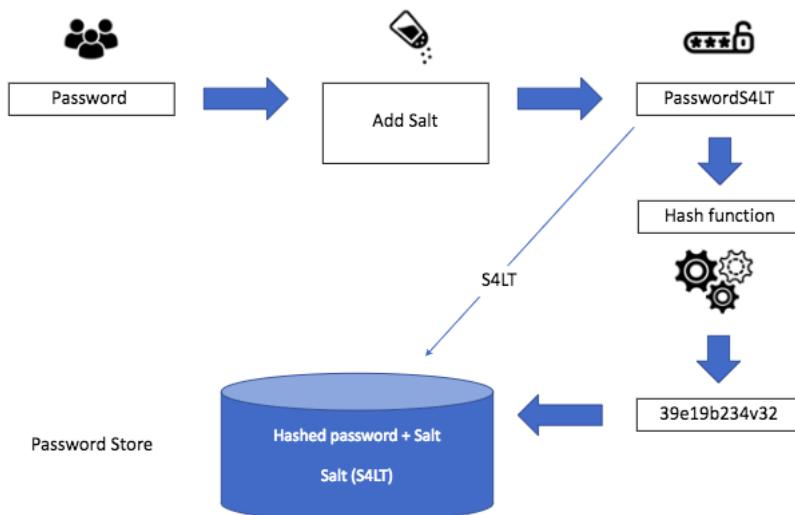


Figura 4.8: Struttura alla base della fase di criptaggio. Fonte: www.okta.com

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 per i diversi livelli di accessibilità. In particolare il 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 e 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 che è l'acronimo di Web Content Accessibility Guidelines. Esse si basano su 4 principi:

- 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 (minimo), AA e AAA (massimo). Il livello A rappresenta il livello minimo di conformità, mentre il livello AAA rappresenta il massimo livello di conformità possibile.

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 4.9: Struttura delle linee guide coi rispettivi livelli. Fonte: www.antoniogiovannischiafone.it

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: si necessita di 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 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 quella cella con il rispettivo titolo. In aggiunta nei tag di intestazione deve essere utilizzato l'attributo scope, che permette di identificare lo scopo di quella cella. Per finire, al fine di far capire all'utente quali dati tale tabella rappresenta 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.

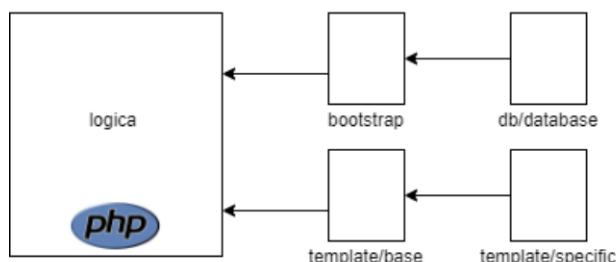


Figura 4.10: Struttura del progetto.

- La base della pagina web è un file il quale al suo interno contiene tutti i riferimenti a file di stile esterni, file js comuni a tutte le pagine e riferimenti a librerie.
- Ogni pagina è composta da più componenti, ed ognuno è composto da un template al cui interno è presente solo codice html con la strut-

tura di quella sezione, un file javascript il quale permette di eseguire chiamate asincrone a un terzo file php il quale, a seconda dell'operazione richiesta richiede un'interrogazione al database e ritorna 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, il quale 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

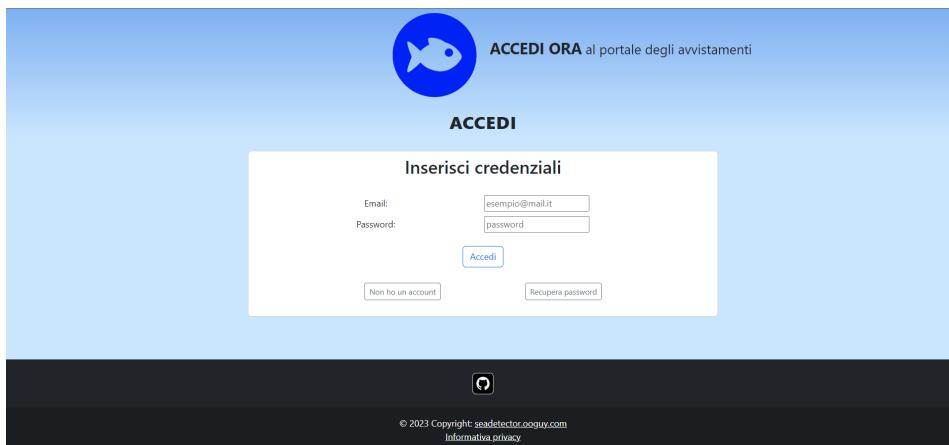


Figura 4.11: Struttura della pagina di Login.

La pagina 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 su quest'ultimo, appare un modale dove, inserendo la propria email, l'utente riceverà le informazioni necessarie per accedere al portale, rispettando le regole di sicurezza.

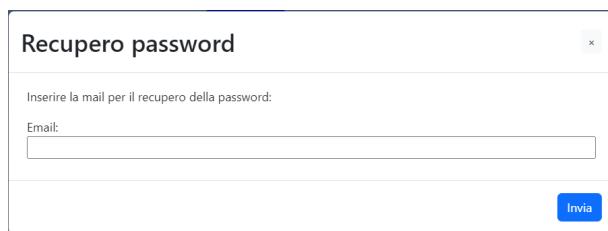
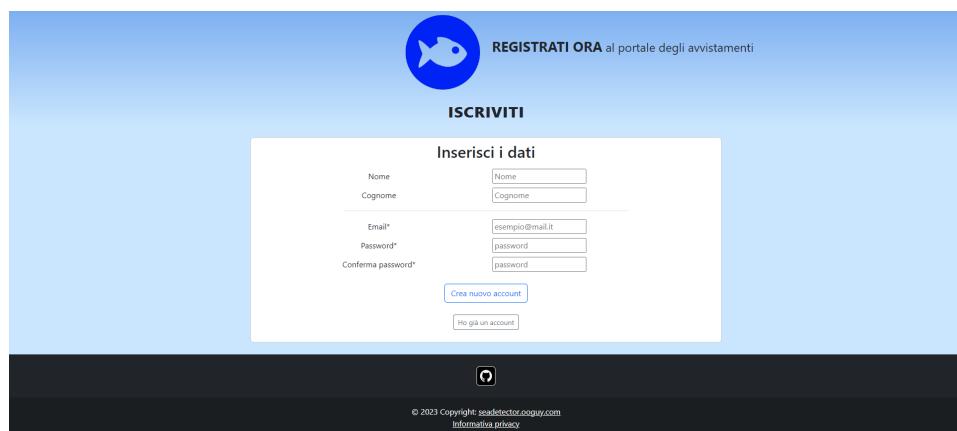


Figura 4.12: Modale per il recupero della password.

La seconda parte 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



The screenshot shows a registration form titled "ISCRIVITI". At the top, there is a blue circular logo with a white fish icon and the text "REGISTRATI ORA al portale degli avvistamenti". Below the logo, the word "ISCRIVITI" is centered. The main area is titled "Inserisci i dati" and contains fields for "Nome" (Name) and "Cognome" (Surname), both with placeholder text "Nome" and "Cognome". There are also fields for "Email*" (Email) with placeholder "esempio@mail.it", "Password*" (Password) with placeholder "password", and "Conferma password*" (Confirm Password) with placeholder "password". Below these fields are two buttons: "Crea nuovo account" (Create new account) and "Ho già un account" (I already have an account). At the bottom of the page, there is a dark footer bar with a small circular icon and the text "© 2023 Copyright: seadetector.eoguy.com Informativa privacy".

Figura 4.13: Struttura della pagina di Sign-Up.

La pagina 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 4.14: Struttura della pagina dell'informativa sulla privacy.

In questa pagina 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 4.15: Struttura della pagina principale del sito.

La struttura della pagina 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 login.

La parte principale della pagina è divisa in due sezioni: una mappa e una tabella. La mappa mostra tutti gli avvistamenti segnati con dei marker. 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 e 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 loro, per motivi di stile.

Infine, in alto a destra c'è un pulsante che permette di aggiungere un nuovo avvistamento. Questo viene fatto attraverso un modale che appare in sovrapposizione quando si clicca sul pulsante.

Aggiungi avvistamento X

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 4.16: 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 select in cui verranno visualizzati i nomi dei relativi elenchi estratti dal database. Si necessita prima della selezione dell'animale, e solo dopo si abiliterà la select della specie al fine di poter scegliere la specie corretta dell'individuo.

4.3.5 Pagina delle impostazioni

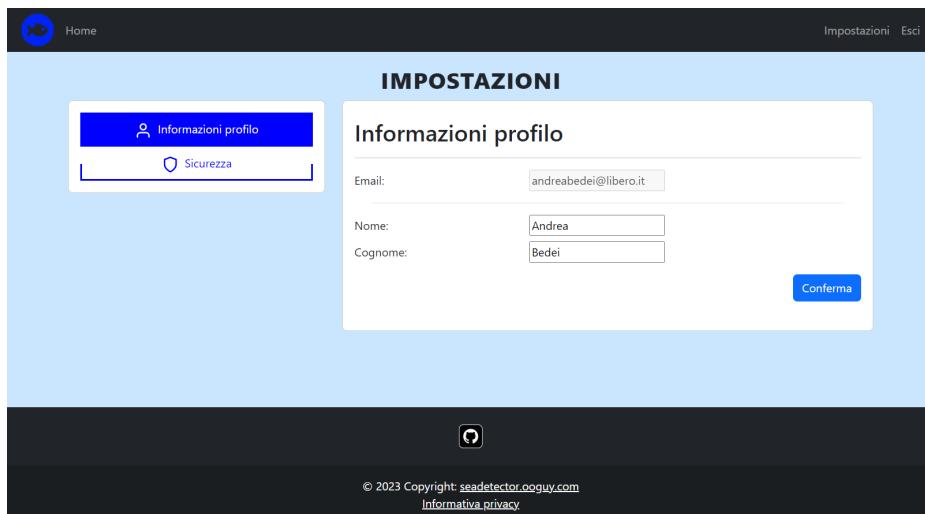


Figura 4.17: Pagina delle impostazioni del profilo utente.

Questa pagina 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 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, i quali indicano 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.

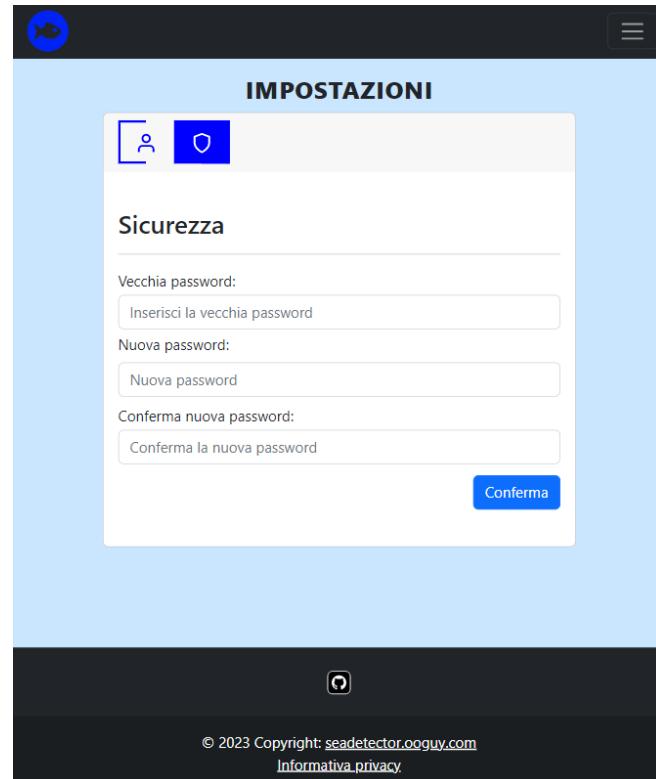


Figura 4.18: Pagina delle impostazioni sulla sicurezza in modalità mobile.

4.3.6 Pagina degli avvistamenti

La pagina degli avvistamenti consente di modificare e aggiungere informazioni sugli avvistamenti precedentemente caricati. Offre anche la possi-

bilità di soddisfare tutti i requisiti specificati dal committente, per questo verrà descritta in dettaglio.

The screenshot shows a web-based application for managing sightings. On the left is a map of the Cesenatico area, Italy, with a blue location marker. To the right is a form titled "DATI AVVISTAMENTO" (Observation Data). The form is divided into sections: "Specifiche" (Details) and "Avvistamento" (Sighting). In the "Specifiche" section, there are fields for User (Utente), Date, Latitude, Longitude, Animal Type (Animale), Species (Specie), Number of individuals (Num. esemplari), Wind speed (Vento), Sea state (Mare), Notes (Note), and a Leaflet button. The "Avvistamento" section has a red "Avvistamento" button. At the bottom are "Immagine" (Image) and "Salva" (Save) buttons.

Figura 4.19: 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 ci 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 l’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 alla specie è presente un pulsante a forma di punto interrogativo che aiuta l’utente a individuare la specie, fornendo informazioni dettagliate sull’elemento scelto. Il pulsante ”salva” invece permette di salvare le informazioni inserite negli input, mentre il bottone ”+Immgine” permette di

aggiungere un’immagine all’avvistamento. Per evitare errori involontari, viene sempre visualizzato un modale di conferma prima di eliminare un avvistamento.



Figura 4.20: Modale di eliminazione.

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



Figura 4.21: Modale di informazione sulla specie.

Per quanto riguarda la seconda parte della pagina essa è incentrata unica-

mente sulle immagine caricate.

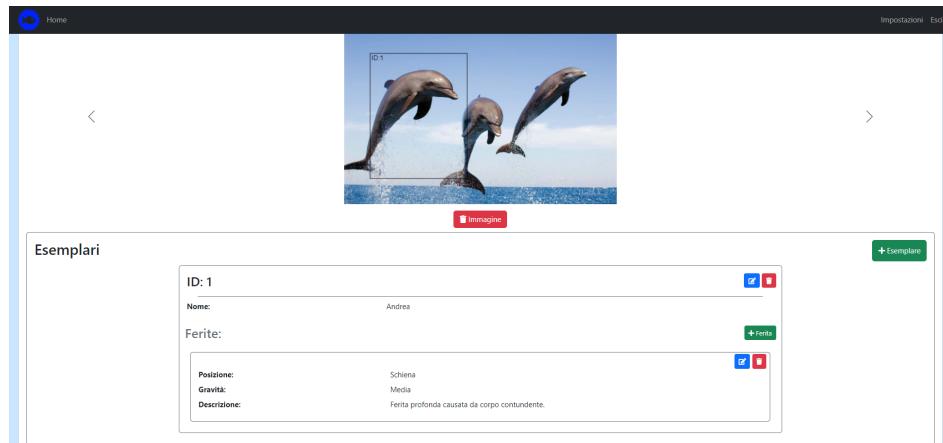


Figura 4.22: Seconda parte della pagina dell'avvistamento con caratteristiche degli esemplari.

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' presente un bottone elimina sotto all'immagine il quale eliminerà non solo l'immagine ma anche tutti i riferimenti ad essa associati. Ad ogni immagine c'è la possibilità di aggiungere un esemplare attraverso il relativo bottone, il quale 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 tali 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. Come negli altri casi anche qui è presente la funzione di eliminazione di un esemplare.

Per ciascuna sottoimmagine è associata una o più ferite, una volta individuate possono essere aggiunte grazie al relativo pulsante e sono possibili anche le operazioni di modifica e cancellazione delle stesse. Un esempio di aggiunta di un individuo può essere il seguente:



Figura 4.23: Modale con canvas per disegnare l’individuo.

Invece per quanto riguarda la modifica dell’individuo si ha il seguente modale:

Modifica

Nome corrente: **Andrea**

Scegliere l'esemplare:

1-Andrea

Nuovo Esemplare

Salva



Figura 4.24: Modifica dell'esemplare.

Come si vede dall'immagine 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, questo perché due esemplari possono avere lo stesso nome.

Per quanto riguarda le ferite, c'è la possibilità di aggiungerle, modificarle attraverso questo semplice modale.

Aggiungi Ferita

Posizione:

Schiena

Gravità:

Lieve

Descrizione:

Ferita profonda causata da corpo contundente.

Inserisci



Figura 4.25: Modale per l'aggiunta e modifica della ferita.

Capitolo 5

Documentazione Algoritmo Di Riconoscimento

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

5.1 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" e che consiste nell'identificare i

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 di oltre il 99%. 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 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.

5.2 Dataset

Le balene e i delfini presenti nei dati a disposizione possono essere identificati attraverso le loro forme, le caratteristiche e le segnalazioni (alcune naturali, altre acquisite) delle loro pinne dorsali, dorsi, teste e fianchi. Alcune specie e alcuni individui hanno caratteristiche molto distinte, mentre altre sono molto meno distinte. Inoltre, le caratteristiche individuali possono variare nel tempo. Questa competizione amplia in modo significativo

questo compito: i dati in gara contengono immagini di oltre 15.000 singoli mammiferi marini di 30 specie diverse, raccolte da 28 diverse organizzazioni di ricerca. Gli individui sono stati identificati manualmente e assegnati un ID-individuo dai ricercatori marini, e il compito è quello di identificare correttamente questi individui nelle immagini.

5.2.1 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:

Specie	Numero di immagini per ogni animale.
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
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

Da come si può notare dalla tabella per ciascuna specie c'è un diverso numero di immagini, quindi ci sarà un grado di accuratezza diverso per ciascuna categoria.

5.3 Algoritmo di riconoscimento

Per quanto riguarda l'algoritmo di riconoscimento è diviso in due parti:

- Fase di training
- Fase di riconoscimento dell'immagine

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

5.3.1 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 vengono utilizzati 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 scomposta in più parti in modo da descriverlo nel dettaglio.

Librerie utilizzate

Listing 5.1: 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:

- **numpy**: 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**: è una libreria per la creazione di visualizzazioni dati. Può essere utilizzata per creare grafici e altre forme di rappresentazione visiva dei dati.
- **pathlib**: 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**: 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**: è una libreria di visualizzazione dei dati che si basa su matplotlib.
- **tensorflow**: 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**: 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 5.2: Caricamento dei dati di training.

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

Il codice 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 il seguente:

	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

Controllo caricamento e distribuzione delle categorie

Listing 5.3: 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');
```

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 il seguente:

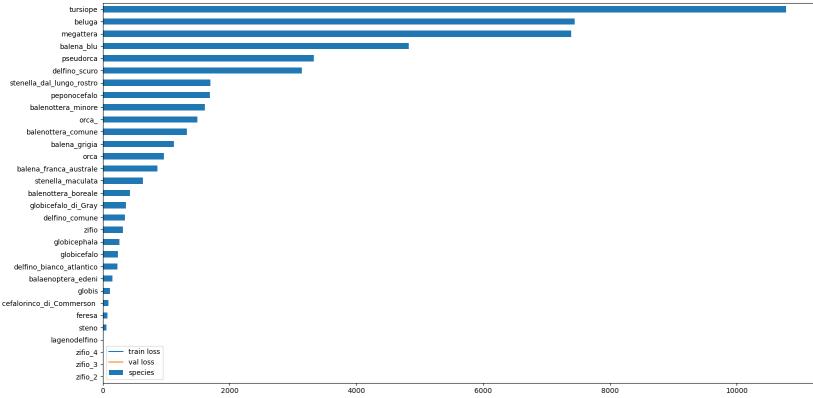


Figura 5.1: Distribuzione delle categorie nella colonna species.

Associazione immagine etichetta

Listing 5.4: 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, 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, i due 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 il seguente:

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

Divisione del dataframe

Listing 5.5: 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 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 impostato 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 5.6: 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 )
```

Questo codice 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, il 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.

Creazione di un generatore di immagini da un dataframe

Listing 5.7: 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 )
```

Questo codice 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 che consiste in 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 su 32 in questo caso. L’argomento `shuffle` viene impostato su `True`, il che significa che le immagini verranno mescolate casualmente durante l’iterazione del generatore.
- **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 del generatore di immagini di test

Listing 5.8: 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 )
```

Questo codice quasi identico a quello descritto sopra e crea un generatore di immagini di test per un modello di apprendimento automatico, settando i reletivi argomenti.

Creazione di un modello pre-addestrato di MobileNetV2

Listing 5.9: 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
```

Questo codice crea un modello pre-addestrato di MobileNetV2 con TensorFlow. 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 che estrae 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 5.10: 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 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". L'attivazione "softmax" è un'attivazione che 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 del modello di apprendimento automatico

Listing 5.11: Compilazione del modello di apprendimento automatico.

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

Il codice 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 descensione del gradiente stocastico. La descensione del gradiente stocastico (SGD) è un algoritmo di addestramento della rete. Il processo consiste nel trovare i pesi ottimali del modello che minimizzano la funzione di perdita, che misura la differenza tra le previsioni del modello e i valori desiderati. Nella descensione del gradiente stocastico, viene calcolato il gradiente (una misura della pendenza) della funzione di perdita 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 funzione di perdita. Questo processo viene ripetuto più volte su diverse iterazioni fino a quando i pesi non convergono verso un valore ottimale. La descensione 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 descensione 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. **Funzione di perdita:** La funzione di perdita utilizzata è `categorical_crossentropy`. Essa è una funzione di perdita 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 allenamento del modello

Listing 5.12: Fase di allenamento del modello.

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

Il codice rappresenta l’allenamento del modello. Il metodo fit accetta i dati di allenamento:

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

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

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

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 allenamento. Gli step di allenamento 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 volta.
- **Training loss:** è una misura della quantità di errore nel modello durante l'allenamento. In altre parole, rappresenta la differenza tra le previsioni del modello e i valori effettivi per le osservazioni di allenamento. Si utilizza come funzione di costo nell'ottimizzazione del modello, dove l'obiettivo è minimizzare questo valore. Una riduzione del Training Loss indica un miglioramento del modello e un'adattamento più preciso alle osservazioni di allenamento..
- **Training accuracy:** è una metrica che misura l'accuratezza della previsione del modello sui dati di allenamento. Viene calcolato come la percentuale di previsioni esatte rispetto al numero totale di previsioni effettuate dal modello sui dati di allenamento. Esso è utilizzato per valutare la performance del modello durante l'allenamento e per

verificare che esso stia effettivamente imparando dai dati di allenamento. Tuttavia, esso 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 allenamento 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 si 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'accuracy 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 5.13: 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()
```

Questo codice 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 varia 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 il seguente:

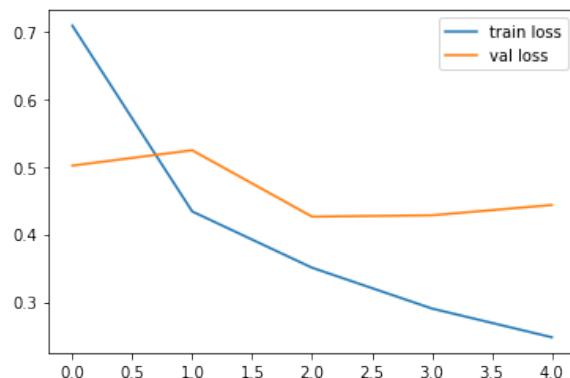


Figura 5.2: Perdita(loss) ottenuta dal modello durante l'addestramento(blu) e perdita ottenuta dal modello durante la validazione(arancio).

Rappresentazione delle curve di accuratezza

Listing 5.14: 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()
```

Questo codice 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 mostra come l'accuratezza del modello varia durante l'addestramento e la validazione, fornendo informazioni utili per valutare l'andamento del processo di apprendimento e l'efficacia del modello.

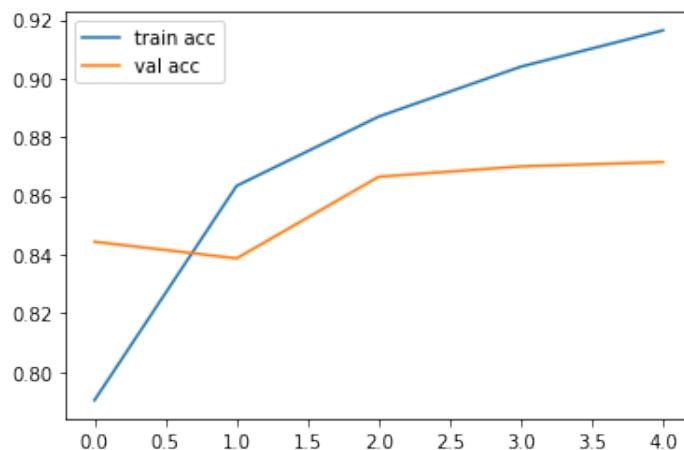


Figura 5.3: L'accuratezza ottenuta dal modello durante l'addestramento(blu) e l'accuratezza ottenuta dal modello durante la validazione(arancio).

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 essa si alza.

Determinazione dell’accuratezza media

Listing 5.15: Determinazione dell’accuratezza.

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

Il codice 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, utilizzando. Nel nostro caso l’accuratezza ottenuta è quella del 86.95%.

Matrice di confusione e report di classificazione

Listing 5.16: 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 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, il 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. Il 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 in cui il modello presenta maggiori difficoltà.

Visualizzazione della matrice di confusione

Listing 5.17: 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 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.

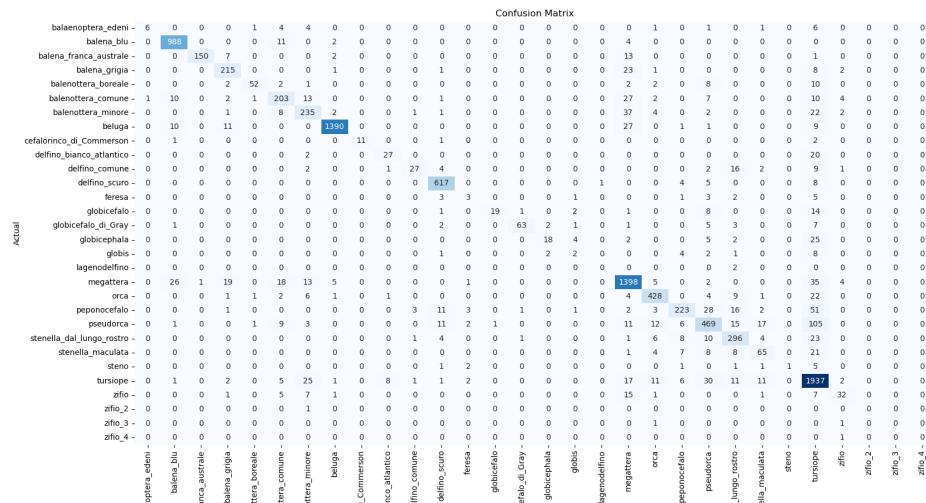


Figura 5.4: Matrice di confusione.

Visualizzazione della precisione per ogni specie

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

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 australe	0.99	0.87	0.93	173
balena grigia	0.82	0.86	0.84	251

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
zifio3	0.00	0.00	0.00	2
zifio4	0.00	0.00	0.00	1

5.3.2 Fase di riconoscimento dell'immagine

In questa fase esamineremo il codice al fine di identificare una nuova immagine che passeremo all'algoritmo addestrato al fine di riconoscere la specie. Per come è stato implementato 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 5.18: Librerie per il riconoscimento dell'immagine.

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

Queste librerie 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 5.19: 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 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 5.20: 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 prende in input la posizione di un’immagine, la carica, la converte in un array di numpy, normalizza i valori del pixel dell’immagine, aggiunge una dimensione per il batch e fa una previsione sulla classe dell’immagine usando un modello preaddestrato `model1`. 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 5.21: Codice della predizione delle immagini.

```
1 args = sys.argv[1:]
2 for arg in args:
3     img='/percorso/img/temp/' + arg
4     pic=load_img(img, target_size=(224, 224, 3))
5     ris = ris + output(pic) + "-"
6 print(ris)
```

Questa funzione legge gli argomenti passati da riga di comando da uno script bash, cioè il nome di una o più immagini. Per ogni immagine, carica l'immagine dal percorso specificato, elabora l'immagine utilizzando la funzione "output" e 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 " ".

5.4 Integrazione dell'algoritmo col sistema

Capitolo 6

Documentazione Test

6.1 Test con l'utente

Al fine di effettuare operazioni di test direttamente con l'utente di è deciso di creare un prototipo per far interagire direttamente l'utente 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.

Capitolo 7

Conclusioni

Capitolo 8

Utilità

Capitolo di elementi da inserire nei capitoli precedenti.

8.1 Ingegneria del software

8.1.1 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 stato 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.”[1]

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 T tali che, a seguito dell'esecuzione del programma P su tutti i casi di T, ogni arco del grafo di controllo di P 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.

8.1.2 Analisi

Analizzare il software consiste nel capire le caratteristiche e le funzionalità.

L'approccio per l'analisi del software è stato il 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 ad ogni comando staticamente il tipo di operazioni eseguite sulle variabili. In questo modo avremo delle sequenze di comandi corrispondenti a possibili esecuzioni, le quali sono riconducibili staticamente a sequenze di tali operazioni. Quest'ultime verranno analizzate al fine di individuare delle possibili anomalie nel programma.

Bibliografia

- [1] Khan, K. & Yadav, S. 2022, A Literature Review on Software Testing Techniques.