



Relazione Tecnica

NAOCHALLENGE 2025

Per ulteriori informazioni

Visita il nostro sito [website](#)

Vedi la nostra [repository](#)

Scrivi una mail a: socialnaoartemis@gmail.com

Contents

1 Metodologia e tecniche usate	3
1.1 DevOps e metodologia Agile	3
1.2 GitHub	3
1.3 File gestione presenze	4
2 Descrizione progetto Nao Challenge 2025	5
2.1 Dalla sfida alla realtà: NaoArtemis e la NAO Challenge 2025	5
2.2 Significato del nome NAO Coach & Care (NC&C)	5
2.3 Audace al fianco di NaoArtemis 2025	5
3 Task 1 – Supporto tecnico e motivazionale agli atleti	6
3.1 Descrizione Task 1	6
3.2 Architettura generale del sistema	6
3.3 Tecniche di Computer Visiom	7
3.4 Web Application	7
3.5 Descrizione generale dell'applicazione	11
3.6 Parametri rilevati dall'app	11
3.7 Interfaccia utente	12
3.8 Infrastruttura applicativa	13
4 Task 2 – Inclusione, accessibilità e interazione con il pubblico	13
4.1 Posizionamento e accesso all'infrastruttura digitale	13
4.2 Interazione utente e accessibilità aumentata	13
4.3 Riconoscimento visivo tramite marker ArUco	14
4.4 Miglioramento del sistema vocale tramite moduli TTS avanzati	15
5 Oltre il codice: NaoArtemis sui social	15
5.1 NaoArtemis: il significato del nome tra robotica e futuro	15
5.2 Strumenti di produzione visiva: dal design grafico al set video	15
5.3 Identità visiva: logo, palette cromatica e tipografia	16
5.4 Presenza digitale: gestione strategica dei social media	18
5.5 Piattaforma web: il sito ufficiale NaoArtemis	19

6 Membri del team e ruoli	19
6.1 Tecnologia, inclusione e scuola: il team NaoArtemis nasce alle Stimate	19
6.2 Divisione dei ruoli	20

Abstract:

Il progetto di NaoArtemis è stato sviluppato per la NAO Challenge 2025 con l'obiettivo di promuovere l'integrazione avanzata della tecnologia nel settore sportivo, al fine di migliorare l'esperienza sia degli atleti sia degli spettatori. La missione del progetto si fonda sull'unione tra innovazione tecnologica e inclusività, con particolare attenzione all'accessibilità e all'inclusione.

Uno degli obiettivi principali è l'ottimizzazione della preparazione atletica attraverso l'utilizzo del robot umanoide NAO. Quest'ultimo sarà in grado di monitorare i movimenti degli atleti, fornendo feedback per il miglioramento delle performance. Inoltre, verranno rilevati parametri biometrici quali la frequenza cardiaca e il numero di passi, con l'obiettivo di prevenire infortuni e supportare i processi di recupero fisico. Anche la salute mentale degli sportivi sarà oggetto di attenzione: NAO assumerà il ruolo di assistente motivazionale virtuale, offrendo rinforzi positivi e suggerimenti personalizzati.

Parallelamente, il progetto intende rendere l'esperienza sportiva più inclusiva per il pubblico, con un focus specifico sulle persone con disabilità. In particolare, NAO fornirà supporto comunicativo a persone nello spettro autistico, mediante l'adozione di sistemi di Comunicazione Aumentativa e Alternativa (CAA), favorendo così l'interazione e la partecipazione attiva agli eventi sportivi.

Infine, in collaborazione con la società sportiva Audace, il progetto prevede la sperimentazione sul campo delle soluzioni proposte, contribuendo concretamente all'innovazione del panorama sportivo attraverso l'uso della robotica e dell'intelligenza artificiale.

1. Metodologia e tecniche usate

1.1. DevOps e metodologia Agile

Per la gestione del progetto e il monitoraggio continuo dei progressi, il team ha adottato pratiche e strumenti in linea con i principi del paradigma DevOps (Development & Operations), che promuove la collaborazione integrata tra sviluppo software e operazioni tecnologiche.

Lo sviluppo è stato organizzato secondo la metodologia Agile, che prevede una pianificazione iterativa e incrementale. I task principali sono stati suddivisi in sub-task, assegnati ai membri del team in base alle competenze specifiche e agli obiettivi delle singole fasi progettuali. Questo approccio ha favorito una maggiore efficienza operativa, migliorando la produttività e l'organizzazione del lavoro.

L'impiego di strumenti per la pianificazione delle attività, il monitoraggio dell'avanzamento e l'organizzazione dei flussi di lavoro ha garantito una distribuzione chiara delle responsabilità e una comunicazione efficace all'interno del gruppo di lavoro. Parallelamente, il progetto è stato gestito e versionato tramite *GitHub*, permettendo il rilascio di diverse versioni migliorate nel tempo.

1.2. GitHub

GitHub è una piattaforma cloud-based che consente la gestione distribuita del codice sorgente, favorendo la collaborazione tra sviluppatori all'interno di progetti software. Alla base di GitHub vi è Git, un sistema di versionamento distribuito che registra in modo puntuale ogni modifica apportata al codice, consentendo la gestione parallela di più versioni del progetto e prevenendo la perdita di dati.

All'interno della piattaforma, i progetti sono strutturati in repository. Gli sviluppatori possono eseguire operazioni di clonazione per ottenere copie locali dei repository, effettuare modifiche in locale, e sincronizzarle con la repository remota tramite comandi come *commit* e *push*.

La collaborazione tra membri del team è facilitata da strumenti avanzati come le *pull request*, che permettono di proporre modifiche e sottoporle a revisione prima dell'integrazione nel branch

principale, e dal sistema di *issue tracking*, utile per la gestione strutturata di bug, segnalazioni e richieste di nuove funzionalità.

Per ottimizzare l'organizzazione del lavoro, il team ha adottato una struttura modulare delle repository, distinguendo due branch, uno dedicato allo sviluppo del codice (“coding”) e uno multimediale (“social”).

1.3. File gestione presenze

Per garantire una gestione efficace delle attività progettuali, il team NaoArtemis ha monitorato costantemente la presenza e il contributo operativo di ciascun componente attraverso un sistema tabellare condiviso che riepiloga i dati aggregati per ogni membro:

- Ore lavorate complessive
- Numero di giorni di presenza effettiva
- Percentuale di partecipazione sul totale delle giornate di lavoro

Queste informazioni sono state fondamentali per valutare l'impegno individuale e per organizzare in modo bilanciato il carico di lavoro, anche in fase di revisione delle responsabilità. Si tratta di ore PCTO (Percorsi per le Competenze Trasversali e l'Orientamento), ovvero attività obbligatorie di formazione esterna previste per gli studenti delle scuole superiori.

L'organizzazione del lavoro ha seguito un approccio ispirato al modello startup/impresa, con una gestione autonoma ed efficiente dei compiti. Questo sistema ha permesso di quantificare con precisione il contributo di ciascun membro, fornendo una base oggettiva per l'analisi dell'operatività del team e per la stesura finale della documentazione relativa alla NAO Challenge 2025.

DIVISIONE TEAM						
Team	Name	Main Task	Email	Phone		
Social	Laura	Gestione social	mascalzonilaura@gmail.com	3755670020		
	Giacomo	Website + CAA (Task 2)	jacksantil22@gmail.com	3923333437		
	Alessandra	Produzione testi e contenuti	afiorini649@gmail.com	345024746		
	Axel	Gestione social (Linkedin)				
	Alessandro	Video	ale.albertini08@gmail.com	3925535800		
Coding	Mattia	App Task1	mattia.begali007@gmail.com	3470948246		
	Haseeb	Task 1 + web app	haseebnabi306@gmail.com	3450447973		
	Marco	Task 2 - nao sugli spalti (CAA)	marcotormazzoli2008@gmail.com	3476449234		
Team Leader	Laura		mascalzonilaura@gmail.com	3755670020		

Fig. 1. Tabella DIVISIONE TEAM: suddivisione dei membri nei rispettivi team, con indicazione del compito principale, email e numero di telefono.

PRESENZE_2						
Date	Attendance	Team	Who	Worked time	% attend.	working day
27/01/2025	YES	Coding	Mattia	2	1	1
27/01/2025	YES	Coding	Haseeb	2	1	
27/01/2025	YES	Coding	Marco	2	1	
27/01/2025	YES	Social	Laura	2	1	
27/01/2025	NO	Social	Alessandra		0	
27/01/2025	NO	Social	Alessandro		0	
27/01/2025	YES	Social	Giacomo	2	1	

Fig. 2. Tabella PRESENZE: registro presenze giornaliero del team per la data del 27/01/2025, con indicazione di partecipazione e team.

2. Descrizione progetto Nao Challenge 2025

2.1. Dalla sfida alla realtà: NaoArtemis e la NAO Challenge 2025

La NAO Challenge è una competizione annuale di robotica educativa rivolta a scuole secondarie, istituti tecnici, università e team di ricerca. L'obiettivo dell'iniziativa è promuovere l'utilizzo del robot umanoide NAO nello sviluppo di soluzioni tecnologiche innovative, capaci di rispondere a esigenze reali della società. Attraverso questa sfida, i partecipanti sono chiamati ad applicare competenze avanzate di programmazione, intelligenza artificiale, progettazione di interfacce e interazione uomo-robot in scenari pratici e multidisciplinari.

Ogni edizione della competizione è incentrata su un tema specifico, pensato per stimolare la creatività progettuale e l'approccio ingegneristico dei team. L'edizione 2025 è focalizzata sullo sport: i partecipanti dovranno progettare applicazioni che utilizzino le funzionalità del robot NAO per ottimizzare, personalizzare o ampliare l'esperienza sportiva, sia dal punto di vista dell'atleta sia da quello del pubblico.

Il focus progettuale comprende aspetti quali il monitoraggio delle performance fisiche, il supporto all'inclusione e all'accessibilità, e l'utilizzo della robotica per migliorare l'engagement e la fruizione dello sport. Partecipare alla NAO Challenge significa affrontare un contesto competitivo ad alto contenuto tecnologico, in cui è fondamentale integrare competenze tecniche con sensibilità verso tematiche sociali e di impatto.

2.2. Significato del nome NAO Coach & Care (NC&C)

Il nome NAO Coach & Care, abbreviato in NC&C, rappresenta in modo sintetico ed efficace la doppia finalità del nostro progetto per la NAO Challenge 2025.

Da un lato, il termine *Coach* evidenzia il ruolo del robot NAO come assistente tecnico per l'allenatore, capace di supportare l'analisi tattica e fisica durante le partite e gli allenamenti. Dall'altro, il termine *Care* richiama la componente inclusiva del progetto, che mira a migliorare l'accessibilità sugli spalti per persone nello spettro autistico, grazie al supporto del robot e all'uso della Comunicazione Aumentativa e Alternativa (CAA).

2.3. Audace al fianco di NaoArtemis 2025

Audace Calcio a 5 Femminile, una delle realtà sportive più consolidate e innovative del panorama veronese, è sponsor ufficiale del progetto NaoArtemis per l'edizione 2025 della NAO Challenge. Il club si distingue a livello nazionale per l'eccellenza nei risultati sportivi e per l'approccio multidimensionale alla preparazione atletica, con un focus costante sul benessere fisico e mentale delle proprie atlete.

La collaborazione tra Audace e il team NaoArtemis nasce da una visione condivisa che pone al centro l'innovazione tecnologica applicata allo sport, l'inclusività e la promozione del talento giovanile. In particolare, il supporto di Audace ha permesso di testare in un contesto reale le soluzioni sviluppate nel progetto, che integrano robotica e intelligenza artificiale a beneficio della performance, della prevenzione degli infortuni e dell'accessibilità dell'attività sportiva.

Audace si conferma così non solo come una società impegnata sul campo, ma anche come partner attivo nello sviluppo di nuove tecnologie a servizio dello sport femminile, della sostenibilità e dell'inclusione. Il sostegno al progetto NaoArtemis rappresenta un ulteriore passo verso una visione dello sport come strumento evolutivo, educativo e socialmente responsabile.



Fig. 3. Logo of Audace C5 Verona

3. Task 1 – Supporto tecnico e motivazionale agli atleti

3.1. Descrizione Task 1

Il primo robot NAO sarà configurato per svolgere il ruolo di assistente tecnico virtuale, operando direttamente nell’area tecnica. Le sue funzioni principali includono:

- Tracciamento e Analisi del Movimento Sportivo: attraverso sistemi di computer vision e tracciamento dei movimenti, il robot sarà in grado di analizzare in modo asincrono la postura e le dinamiche motorie degli atleti, elaborando i dati raccolti durante allenamenti e competizioni anche in un secondo momento.
- Analisi biomeccanica in tempo reale: Nel contesto del progetto, vengono acquisiti dati biometrici e biomeccanici mediante dispositivi wearable (smartwatch), includendo frequenza cardiaca (bpm), conteggio dei passi e velocità di spostamento. I dati rilevati vengono trasmessi in tempo reale e memorizzati all’interno di un database relazionale basato su *PostgreSQL*, dove sono strutturati in formato tabellare per consentire operazioni di interrogazione, analisi ed elaborazione successive in ambiente software dedicato.
- Elaborazione di dati tattici e visuali: i dati raccolti saranno trasformati in un unico diagramma di Voronoi trasmesso sulla webapp, e saranno utilizzati per generare suggerimenti strategici personalizzati per gli allenatori, utili per studiare le partite degli avversari o gli allenamenti.

3.2. Architettura generale del sistema

Il sistema NaoArtemis è strutturato secondo un’architettura distribuita, composta da due server indipendenti sviluppati in ambienti Python differenti per garantire la piena compatibilità tra componenti SDK e modelli avanzati di intelligenza artificiale (AI).

- Il server in Python 2 è destinato esclusivamente alla comunicazione diretta con il robot NAO, resa possibile tramite l’uso della SDK di NAOqi, che richiede un ambiente Python 2.x per l’accesso alle API di basso livello del robot. A supporto dello sviluppo, è stata costantemente consultata la documentazione ufficiale disponibile online: <http://doc.aldebaran.com/2-8/naoqi/index.html>.
- Il server in Python 3 ospita invece la Web Application e l’infrastruttura dedicata ai modelli AI più recenti. In questo contesto sono attivi:
 - YOLO per la rilevazione e tracciamento in tempo reale di giocatori e oggetti in campo;
 - Per ogni giocatore viene calcolato il colore medio della divisa, dopodiché l’algoritmo K-means assegna un’etichetta di squadra (ad esempio rosso o blu) in base alla somiglianza cromatica.

- Generazione vocale di messaggi motivazionali o comunicazioni tattiche tramite le API Text-to-Speech di OpenAI;
- Un algoritmo per il supporto tattico destinato a suggerire strategie in tempo reale durante le fasi di gioco.

I due server comunicano tra loro tramite interfacce TCP/IP, garantendo la sincronizzazione tra input visivi, output vocali e dati di stato degli atleti. Tale configurazione consente di superare le limitazioni di compatibilità tra il framework NAO e gli strumenti di AI moderni.

3.3. Tecniche di Computer Vision

Il componente "Viceallenatore" integra funzionalità di computer vision, generazione automatica di suggerimenti tattici e monitoraggio in tempo reale dei parametri di gioco.

- L'analisi della scena sportiva avviene tramite un sistema basato su YOLO e K-means, i quali rilevano i giocatori e ne determinano la posizione sul campo attraverso coordinate spaziali (X, Y).
- A differenza di una heatmap tradizionale, il sistema elabora un unico diagramma di Voronoi (Voronoi Soccer) che rappresenta lo spazio di influenza di ciascun giocatore sul campo. Questo approccio consente una visualizzazione tattica più efficace per l'allenatore, evidenziando coperture, spazi liberi e squilibri posizionali.
- I dati biometrici ricevuti dalla WebApp (battito cardiaco, numero di passi, velocità media, ecc.) vengono analizzati per valutare lo stato di forma dell'atleta. Il sistema può così suggerire in tempo reale sostituzioni strategiche, anche in risposta a condizioni critiche rilevate (affaticamento, calo di rendimento, rischio infortunio).
- Infine, un algoritmo fornisce suggerimenti tattici dinamici, adattando il gioco sulla base del comportamento avversario e della situazione in campo. Il robot NAO può poi trasmettere tali indicazioni agli atleti tramite sintesi vocale, rendendo l'interazione uomo-macchina più immediata ed efficace.

3.4. Web Application

La WebApp sviluppata in ambiente Python 3 è il cuore gestionale del sistema e consente il controllo dell'intera infrastruttura tramite un'interfaccia utente reattiva, accessibile via browser.

Funzionalità principali

- Gestione degli infortuni, con tracciamento delle condizioni mediche e storico sanitario di ciascun atleta;
- Pianificazione delle partite, calendario eventi, avvisi e statistiche pre/post-gara;
- Database centralizzato, che raccoglie dati biometrici, comportamentali e tecnici, resi accessibili e aggiornabili in tempo reale da ciascun modulo dell'architettura.

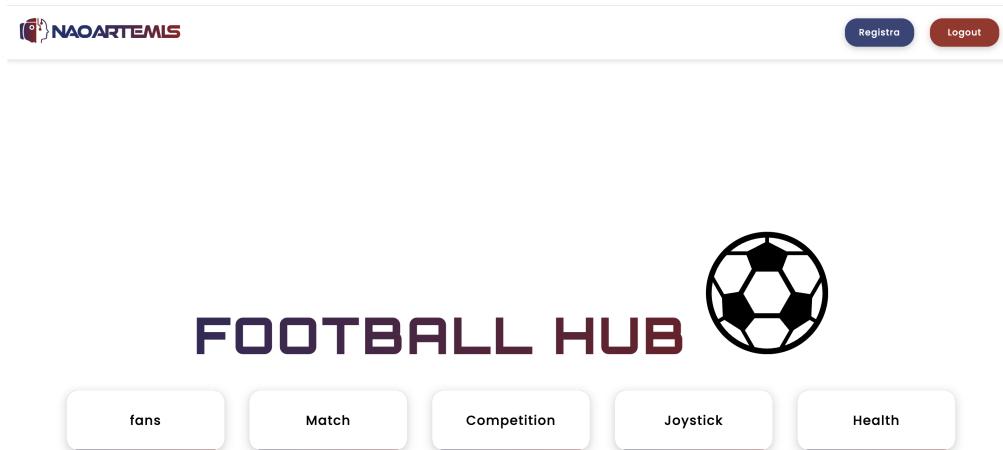


Fig. 4. Home screen della webapp *Football Hub*, punto di accesso alle varie funzionalità: gestione della partita, competizione, controllo del robot tramite joystick e monitoraggio della salute.

This section of the webapp is titled "GESTIONE PARTITA". It features a central button group with five buttons: "clear db health" (blue), "tactics" (red, highlighted), "tactics" (blue), "time" (blue), and "points" (blue). Below this are two side-by-side sections. The left section is titled "VIDEO VORONOI" with a "Carica Video Voronoi" button. The right section is titled "VIDEO DA ANALIZZARE" with "Carica Video" and "Analyze" buttons, and a "Scegli file" input field.

Fig. 5. Sezione “Gestione Partita” della webapp, che consente il caricamento di video per analisi tattiche.



Fig. 6. Sezione dedicata alla giornata di campionato e ai risultati della partita dell'Audace.

Controllo manuale

Per offrire massimo controllo e flessibilità all’utente, il team ha deciso di non adottare Choregraphe, il software ufficiale per la programmazione del robot NAO, a causa di alcune limitazioni tecniche e operative. In alternativa, è stato sviluppato un sistema di comandi manuali essenziali ma strategici, tramite la WebApp. Questo modulo consente all’utente di:

- Attivare/disattivare la trasmissione video;
- Avviare il riconoscimento ArUco;
- Inviare messaggi vocali;
- Controllare azioni base del NAO in modo diretto e flessibile.

Appendice tecnica: componenti di elaborazione video

Flusso in Python 3 Elaborazione del flusso video registrato tramite OpenCV e decodifica dei frame JPEG. I frame vengono analizzati successivamente alla partita con modelli di visione artificiale (come YOLO e Kmeans).

```

results = MODEL_PLAYERS(frame, conf=0.2)
det = sv.Detections.from_ultralytics(results[0])
players = det[det.class_id != 0].with_nms(threshold=0.5)

frame_data = []
if len(players) > 0:
    tracked = tracker.update_with_detections(players)
    colors = [np.mean(frame)]
    if len(colors) >= 2:
        kmeans = KMeans(n_clusters=2, n_init=10).fit(colors)
        teams = ["blue" if l == 0 else "red" for l in kmeans.labels_]
    else:
        teams = ["unknown"] * len(tracked.xyxy)
    tracked.data["team"] = teams

    labels = [{"#(tid) ({team})" for tid, team in zip(tracked.tracker_id, teams)}]
    annotated = ellipse_annotator.annotate(scene=frame, detections=tracked)
    annotated = label_annotator.annotate(scene=annotated, detections=tracked, labels=labels)

    for i, box in enumerate(tracked.xyxy):
        center_x = (box[0] + box[2]) / 2
        center_y = (box[1] + box[3]) / 2
        center = np.array([center_x, center_y, 1])
        if homography_matrix is not None:
            pt = homography_matrix @ center
            pt /= pt[2]
        else:
            pt = center
        frame_data.append((float(pt[0]), float(pt[1]), teams[i]))
frame = annotated

```

Fig. 7. Codice Python 3 per l'analisi dei frame con YOLO e clustering KMeans

S

```

@app.route('/api/app/utente/<id>', methods=['POST'])
def api_app_utente(id):
    if id != None and id != '':
        if request.method == 'POST':
            try:
                #("username":value, "password":value)
                json = request.json
                username = json["username"]
                password = json["password"]
                data = db_helper.select_account_player(username, password)
                return jsonify({'code': 200, 'message': 'OK', 'data': data}), 200
            except Exception as e:
                logger.error(str(e))
                return jsonify({'code': 500, 'message': str(e)}), 500
        else:
            logger.error('No id argument passed')
            return jsonify({'code': 500, 'message': 'No id was passed'}), 500
    else:
        logger.error('No id argument passed')
        return jsonify({'code': 500, 'message': 'No id was passed'}), 500

@app.route('/api/app/dati/id>', methods=['POST'])
def api_app_dati(id):
    if id != None and id != '':
        if request.method == 'POST':
            try:
                #("id_player":id, "bpm":98, "passi":72, "velocita":4")
                json = request.json
                id_player = json["id_player"]
                bpm = json["bpm"]
                passi = json["passi"]
                velocita = json["velocita"]
                db_helper.insert_dati(id_player, bpm, passi, velocita)

                if bpm >= 185:
                    player = db_helper.get_player_by_id(id_player) # ricevo una lista con [nome, cognome]
                    text = f'il giocatore {player[0]} ({player[1]}) deve abbassare il ritmo'
                    nao_animatedSayText(text)

                return jsonify({'code': 200, 'message': 'OK'}), 200
            except Exception as e:
                logger.error(str(e))
                return jsonify({'code': 500, 'message': str(e)}), 500
        else:
            logger.error('No id argument passed')
            return jsonify({'code': 500, 'message': 'No id was passed'}), 500
    else:
        logger.error('No id argument passed')
        return jsonify({'code': 500, 'message': 'No id was passed'}), 500

```

Fig. 8. Salvataggio dati nel database da parte dell'app

Flusso in Python 2 Acquisizione dell'immagine dalla webcam del robot NAO tramite ALVideoDevice, con parametrizzazione (fps, risoluzione, luminosità) e invio del frame in formato JPEG al server Python 3 per l'elaborazione successiva.

```

def nao_get_image(nao_ip, nao_port):
    video_proxy = ALProxy("ALVideoDevice", nao_ip, nao_port)           # NAO webcam

    # Set the camera parameters
    name_id      = "video_image_" + str(random.randint(0,100))          # The same Name could be used only six time
    camera_id    = 0                                                       # Use the top camera (1 for bottom camera)
    resolution   = 1                                                       # Image of 320*240px
    color_space   = 13                                                      # RGB
    camera_fps   = 10                                                       # fps
    brightness_value = 55                                                 # default of 55
    video_proxy.setParameter(camera_id, 0, brightness_value)             # brightness

    # Subscribe to the video feed
    video_client = video_proxy.subscribeCamera(name_id, camera_id, resolution, color_space, camera_fps)
    try:
        while True:
            image       = video_proxy.getImageRemote(video_client)  # Capture a frame from the camera

            image_width  = image[0]
            image_height = image[1]
            image_channels = 3
            image_data   = np.frombuffer(image[6], dtype=np.uint8).reshape((image_height, image_width, image_channels))

            desired_width = 640
            desired_height = 480
            resized_image = cv2.resize(image_data, (desired_width, desired_height))

            ret, buffer = cv2.imencode('.jpg', resized_image)
            frame       = buffer.tobytes()

            yield (b'--frame\r\n'
                   b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
    except Exception as e:
        logger.error(str(e))
    finally:
        video_proxy.unsubscribe(video_client)

```

Fig. 9. Codice Python 2 che riceve un testo in ingresso e genera un file audio MP3 utilizzando la libreria OpenAI.

Qualità video e accuratezza dell'analisi Durante le fasi di test, è emerso che l'utilizzo della videocamera integrata del NAO, presenta limitazioni in termini di risoluzione e qualità dell'immagine. Per questo motivo, si è scelto di sperimentare la registrazione delle partite tramite smartphone o videocamere professionali, ottenendo risultati migliori grazie alla maggiore nitidezza e stabilità del video, che agevolano le fasi successive di analisi basata su computer vision.

WebApp Flask Gestione delle API di autenticazione e raccolta dati biometrici tramite richieste HTTP POST da applicazione client o mobile.

3.5. Descrizione generale dell'applicazione

L'applicazione è progettata per il monitoraggio continuo e in tempo reale di parametri biometrici relativi all'attività fisica e alle condizioni fisiologiche dell'utente. Il sistema consente l'acquisizione, la trasmissione e la gestione centralizzata dei dati.

3.6. Parametri rilevati dall'app

- Numero di passi: calcolato mediante l'utilizzo dei sensori presenti nello smartphone, in particolare accelerometro e giroscopio, che rilevano variazioni nella postura e nel

movimento dell’utente. L’elaborazione avviene localmente tramite un modulo di analisi del pattern motorio.

- Velocità di movimento: derivata dal tracciamento GPS attraverso un algoritmo che valuta la variazione di posizione rispetto al tempo, con eventuali correzioni su base temporale per compensare anomalie dovute al segnale satellitare o a cambiamenti di direzione.
- Frequenza cardiaca (bpm): ottenuta da un dispositivo indossabile che comunica in modalità wireless con la piattaforma Google Fit. L’applicazione si interfaccia con tale piattaforma tramite API REST, effettuando interrogazioni periodiche per la raccolta dei dati aggiornati, poi integrati nel sistema interno.

3.7. *Interfaccia utente*

All’avvio, l’applicazione presenta all’utente un’interfaccia grafica intuitiva per l’autenticazione o la registrazione, in cui vengono richiesti dati identificativi obbligatori, tra cui nome, cognome, ID giocatore univoco e password cifrata. Questi dati sono trattati in conformità alle normative vigenti in materia di protezione dei dati personali con memorizzazione sicura lato server.

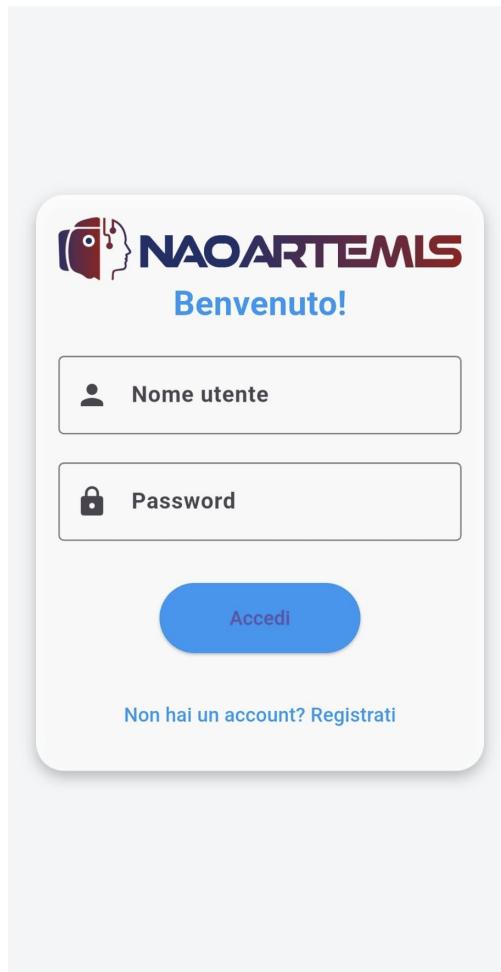


Fig. 10. Interfaccia della schermata di login

3.8. Infrastruttura applicativa

L'applicazione mantiene una connessione persistente (HTTP) con un server remoto che gestisce l'infrastruttura applicativa e le seguenti funzionalità principali:

- Autenticazione e autorizzazione degli utenti: verifica delle credenziali fornite in fase di login e accesso controllato ai servizi.
- Sincronizzazione automatica dei dati biometrici: i dati rilevati localmente vengono aggregati e trasmessi periodicamente al server secondo una logica basata su intervalli temporali o eventi specifici (ad es. raggiungimento soglia bpm o variazione posizione significativa).
- Archiviazione, validazione e analisi: i dati biometrici vengono archiviati in un database centralizzato, dove possono essere elaborati tramite algoritmi per estrarre informazioni sulla performance, rilevare anomalie o generare alert.
- Visualizzazione: l'infrastruttura server è predisposta per esporre i dati a dashboard esterne, come la webapp, in grado di mostrare grafici, mappe, statistiche e cronologie individuali, utili al monitoraggio dell'attività sportiva o dello stato di salute degli utenti.

4. Task 2 – Inclusione, accessibilità e interazione con il pubblico

4.1. Posizionamento e accesso all'infrastruttura digitale

Durante la Task 2, il robot NAO viene collocato sugli spalti, in una postazione sopraelevata, al fine di ottenere una visuale panoramica sul campo e facilitare la comunicazione diretta con il pubblico. In questa configurazione, il NAO è connesso a Internet e integrato in un'architettura server distribuita a due livelli:

- Un server Python 3, responsabile dell'elaborazione dei dati, dell'inferenza tramite modelli di intelligenza artificiale e dell'interfaccia con il database.
- Un server in Python 2, che garantisce la compatibilità con le API proprietarie NAOqi, necessarie per la comunicazione a basso livello con l'hardware del robot.

Il NAO, durante questa fase, accede alle informazioni elaborate nella Task 1, interrogando il database tramite il livello Python 3.

4.2. Interazione utente e accessibilità aumentata

Il robot NAO agisce come interfaccia conversazionale principale per l'utente. Per garantire un livello elevato di accessibilità comunicativa, in particolare per utenti con disabilità, è stato progettato un sistema cartaceo basato su CAA (Comunicazione Aumentativa Alternativa). I fogli CAA contengono:

- Simboli visivi e immagini di facile interpretazione.
- Un marker visivo ArUco, utilizzato per l'attivazione di funzioni specifiche tramite riconoscimento visivo da parte del NAO.

Quando l'utente mostra un foglio contenente un ArUco marker, il robot lo identifica tramite la propria fotocamera, decodifica l'ID associato e attiva la funzione corrispondente sul server Python 3. Quest'ultimo, qualora necessario, invoca procedure sul server Python 2 per l'interazione diretta con il robot.



Fig. 11. Interfaccia della sezione *Gestione Tribuna*, con controlli dedicati alla partita e alle funzionalità accessibili via NAO.

4.3. Riconoscimento visivo tramite marker ArUco

La scelta dei marker ArUco è giustificata da caratteristiche tecniche che li rendono ottimali per ambienti complessi:

- Affidabilità elevata nel riconoscimento.
- Velocità di rilevamento superiore rispetto ad altri codici visivi.
- Capacità di gestire più marker simultaneamente nella stessa inquadratura.
- Robustezza a rotazioni, distorsioni e parziali occlusioni.

Nel progetto è stato scelto l'utilizzo dei marker ArUco in sostituzione dei NaoMark nativi di NAO. Questa scelta è motivata dalla maggiore compatibilità e stabilità offerte dalla libreria open source *OpenCV*, che include un modulo dedicato alla rilevazione e decodifica degli ArUco marker.

Ogni marker ArUco è mappato a una funzione specifica nel sistema software e rappresenta una modalità semplice ed efficace per comandare il robot senza interfacce vocali o touch.



Fig. 12. Esempio di marker ArUco utilizzato per il riconoscimento tramite visione artificiale.

4.4. *Miglioramento del sistema vocale tramite moduli TTS avanzati*

Per ottimizzare la comunicazione tra il robot e gli utenti, è stato integrato un modulo di *Text-to-Speech* basato su tecnologia OpenAI. Questo aggiornamento ha permesso di:

- Migliorare la qualità e la naturalezza della voce del robot.
- Ridurre la frequenza di errori di pronuncia e aumentare la chiarezza espressiva.
- Rendere le comunicazioni vocali più fluide e comprensibili, anche in contesti dinamici.

Grazie a questa integrazione, il NAO risulta più efficace e accessibile nella trasmissione di messaggi, aumentando l'inclusività e l'impatto delle interazioni vocali.

5. Oltre il codice: NaoArtemis sui social

5.1. *NaoArtemis: il significato del nome tra robotica e futuro*

Il nome NaoArtemis nasce dall'unione tra il robot umanoide NAO, protagonista tecnologico del progetto, e Artemis, divinità greca associata alla forza, alla protezione e alla libertà. Questa scelta non è casuale: NAO rappresenta il presente dell'innovazione robotica, mentre Artemis richiama un ideale di guida verso il futuro, simbolo di equilibrio tra tecnologia e umanità. Il nome riflette quindi la missione del progetto: coniugare intelligenza artificiale, inclusione e sport, per creare un sistema in grado di supportare atleti e tifosi in modo intelligente, etico e sostenibile.

5.2. *Strumenti di produzione visiva: dal design grafico al set video*

Per la realizzazione dell'identità visiva del progetto NaoArtemis, il team ha adottato un approccio professionale, selezionando strumenti e ambienti adatti alla comunicazione digitale e multimediale.

I loghi ufficiali sono stati progettati utilizzando Adobe Illustrator, che ha permesso una costruzione precisa e coerente degli elementi grafici.

Per la produzione dei contenuti social — come post, storie e caroselli per Instagram, Facebook e TikTok — è stato utilizzato Canva, uno strumento online intuitivo e collaborativo, particolarmente efficace per la creazione rapida di grafiche accattivanti nel rispetto del branding del progetto. La combinazione di template, font personalizzati e palette ha garantito uniformità e impatto visivo su tutti i canali.

Infine, i video promozionali e dimostrativi sono stati registrati all'interno di un vero e proprio set fotografico. Questo ha permesso di ottenere contenuti professionali, valorizzando l'interazione con i robot NAO. L'integrazione di strumenti tecnici e ambienti professionali ha contribuito a rendere la comunicazione di NaoArtemis non solo efficace, ma anche allineata agli standard delle produzioni multimediali contemporanee.

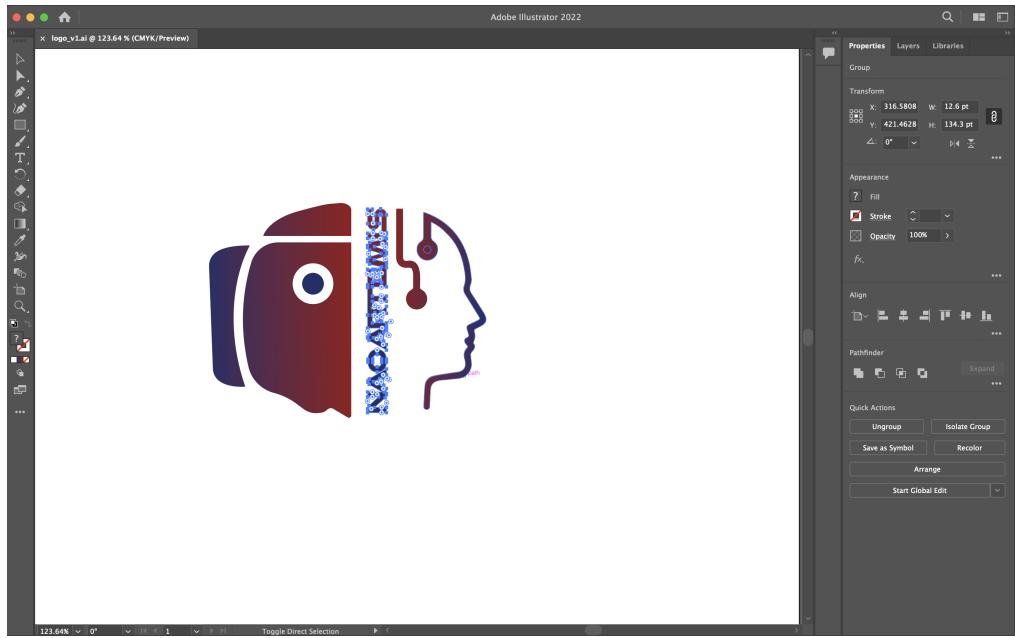


Fig. 13. Illustrator

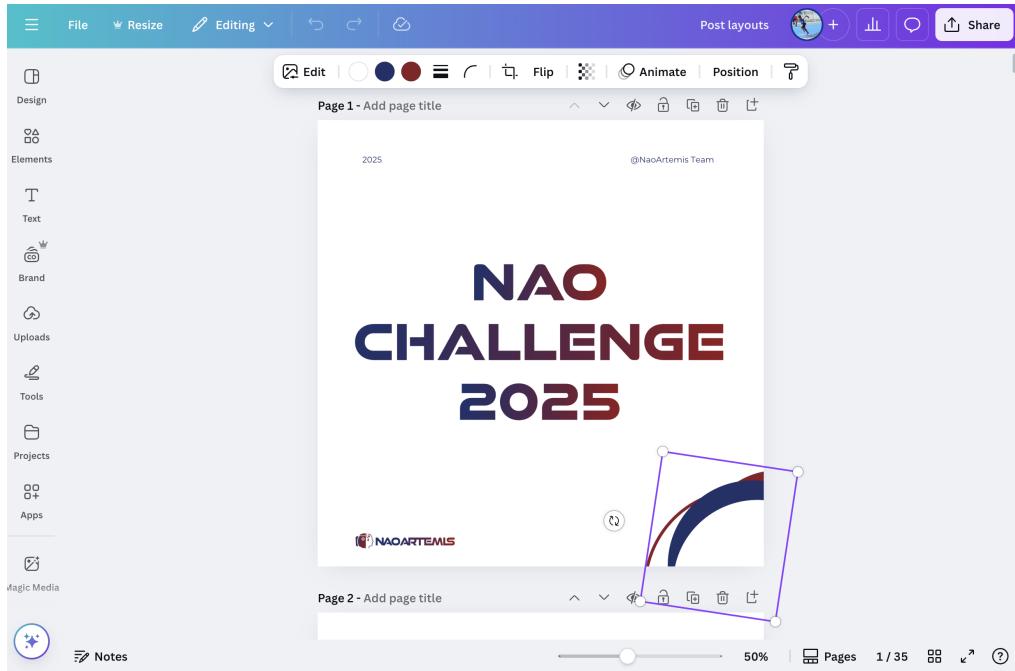


Fig. 14. Canva

5.3. Identità visiva: logo, palette cromatica e tipografia

L'identità visiva di NaoArtemis è stata progettata per comunicare immediatamente i valori di futuro, accessibilità e innovazione.

Il logo di NaoArtemis è pensato per rappresentare visivamente l'essenza del progetto: l'integrazione tra tecnologia e umanità.

L'elemento centrale del logo è costituito dalla fusione tra la testa stilizzata di un robot umanoide NAO e il profilo di un volto umano.

A collegare le due parti è inserita una rete neurale stilizzata, elemento grafico che rappresenta l'intelligenza artificiale.

Nel suo insieme, il logo comunica futuro, inclusività e innovazione, valori chiave su cui si fonda l'intero progetto NaoArtemis.

Per garantire versatilità comunicativa e coerenza visiva in diversi contesti, il team NaoArtemis ha sviluppato tre versioni distinte del proprio logo. Ciascuna versione è stata progettata per rispondere a esigenze di formato, leggibilità e impatto visivo, mantenendo comunque una forte unità grafica.

Logo 1 – Profilo social (versione compatta e verticale)

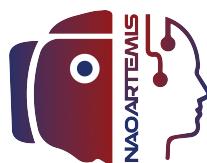


Fig. 15. Logo versione 1

Questa versione è ottimizzata per l'uso come immagine profilo sulle piattaforme social (Instagram, Facebook, TikTok, YouTube). Grazie al formato verticale e alla composizione compatta, il logo mantiene chiarezza e riconoscibilità anche in dimensioni ridotte. Il nome del progetto è integrato verticalmente tra la sagoma robotica e il volto umano.

Logo 2 – Merchandising e stampa (versione centrale)



Fig. 16. Logo versione 2

Questa variante, pensata per materiali stampati e fisici (magliette e biglietti da visita), presenta una composizione centrata e bilanciata tra il disegno e il nome del team. La disposizione orizzontale della scritta assicura leggibilità immediata.

Logo 3 – Comunicazione ufficiale (versione estesa)



Fig. 17. Logo versione 3

La terza versione è progettata per l'utilizzo in contesti digitali o editoriali in cui è importante

valorizzare il nome del team. Con il testo “NAOARTEMIS” disposto in grande evidenza accanto all’icona composita, questo logo è stato impiegato per intestazioni di documenti e nel sito web.

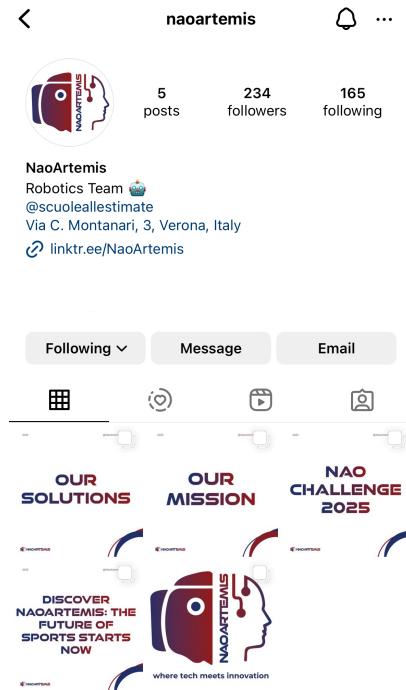
La paletta cromatica utilizza principalmente due colori: blu e rosso, talvolta sfumati come nel caso del logo. Le font scelte sono Montserrat e Good Timing. L’uniformità visiva tra tutti i materiali (social, presentazioni, sito web) garantisce coerenza e riconoscibilità al team.

5.4. Presenza digitale: gestione strategica dei social media

La presenza online si è sviluppata su diverse piattaforme: Instagram, TikTok, Facebook, YouTube e LinkedIn, ciascuna con contenuti adattati al proprio pubblico di riferimento e con un piano editoriale condiviso.

Instagram

Su Instagram sono stati pubblicati post statici, caroselli e biografie per presentare il team, la scuola, gli obiettivi e i partner; descrizione del progetto suddivisa in post sequenziali; approfondimenti su sport, AI, inclusività, Audace, benessere e prestazione sportiva; stories dinamiche e reel tematici.



TikTok

Sono stati pubblicati gli stessi reel di Instagram, adattati al formato della piattaforma. I contenuti hanno privilegiato velocità, immediatezza e originalità, per attrarre l’attenzione di un pubblico ancora più giovane.

Facebook

Contenuti condivisi con taglio più istituzionale, per raggiungere genitori, scuole e partner esterni.

YouTube

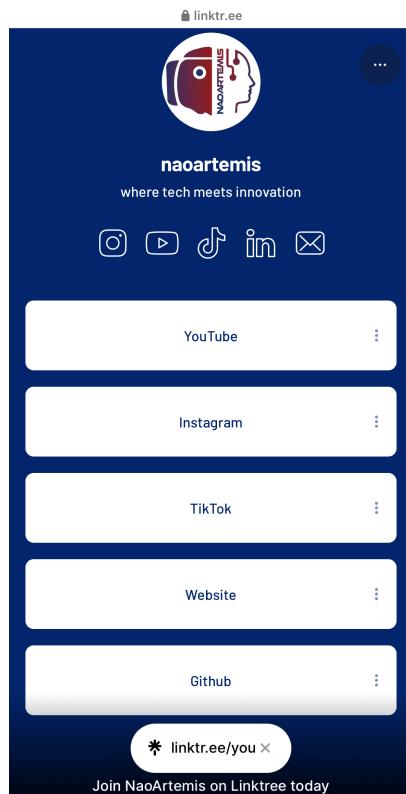
Raccolta di tutti i video pubblicati, più un video completo per presentazioni ufficiali ed eventi.

LinkedIn

Utilizzato per la comunicazione tecnica e professionale con docenti e aziende.

Infine, è stato utilizzato Linktree, uno strumento utile in contesti professionali per organizzare e condividere in modo efficace una raccolta di link per gli accessi ai diversi social, attraverso

un'unica pagina facilmente accessibile.



5.5. Piattaforma web: il sito ufficiale NaoArtemis

È stato realizzato un sito web dedicato al progetto NaoArtemis, sviluppato per la NAO Challenge 2025, al fine di documentare in modo strutturato e accessibile tutte le componenti principali dell'iniziativa. È stata inserita una descrizione tecnica del progetto, articolata nei due task principali (ottimizzazione dell'allenamento sportivo e inclusione dei tifosi). Sono specificati i ruoli dei membri e la scuola di appartenenza. Vi è infine una sezione dedicata alla collaborazione con la società sportiva Audace.

6. Membri del team e ruoli

6.1. Tecnologia, inclusione e scuola: il team NaoArtemis nasce alle Stimate

Il team NaoArtemis nasce all'interno del Liceo “Alle Stimate” di Verona, un istituto con oltre due secoli di storia educativa, fondato nel 1816 da San Gaspare Bertoni. La scuola si distingue per il suo impegno nella formazione integrale della persona, che unisce l'eccellenza accademica alla promozione di valori umani, sociali e spirituali.

In questo contesto di apertura e innovazione, la scuola sostiene attivamente progetti interdisciplinari come la partecipazione alla NAO Challenge. Il team NaoArtemis, composto da studenti del liceo scientifico delle scienze applicate e guidato dal prof. Giovanni Bellorio, nasce proprio da questa visione educativa: coniugare passione per la tecnologia, spirito di squadra e impegno sociale.

6.2. Divisione dei ruoli

Giovanni Bellorio

Docente di informatica, è il coach. Coordina gli incontri e gestisce gli aspetti organizzativi e amministrativi della NAO Challenge.

Mattia Begali – App developer

Responsabile dello sviluppo dell'app per Task 1. Lavora sulla programmazione e sull'integrazione dei sistemi intelligenti in Python.

Laura Mascalzoni – Social media manager e Team Leader

Coordina le piattaforme social, cura i contenuti multimediali e garantisce la coerenza comunicativa.

Giacomo Santi – Website developer

Responsabile del sito web ufficiale. Cura l'accessibilità, l'aggiornamento e la presentazione tecnica del progetto.

Alessandro Albertini – Video content developer

Produce e monta i video. Gestisce la pipeline audiovisiva e l'adattamento ai diversi formati.

Alessandra Fiornini – Social media content developer

Supporta la comunicazione visiva e testuale sui social, mostrando precisione e cura.

Haseeb Nabi – Webapp developer

Programmatore avanzato del robot NAO. Lavora su Task 1, computer vision e Web App.

Marco Tomazzoli – Nao Programmer

Responsabile delle funzionalità di Task 2. Sviluppa l'interazione tramite CAA e il riconoscimento dei simboli ArUco.

