

Relazione Tecnica

NAO Challenge 2025



Team NaoNexus

Per maggiori informazioni:

Visita il nostro sito, la nostra repository
e scrivi una mail a: socialnaonexus@gmail.com

Scuole “Alle Stimate”, Verona

Sommario

La Nao Challenge quest'anno si concentra sul tema dello sport, con l'obiettivo di sviluppare soluzioni innovative in ambito medico e sportivo. Il team NaoNexus ha scelto di impiegare il robot NAO come assistente medico per atleti in fase di riabilitazione, fornendo supporto personalizzato e mirato in base all'infortunio e alle condizioni fisiche del paziente.

Il NAO raccoglie i dati dell'atleta tramite una scheda medica e, grazie a un sistema basato su decision tree, propone una scheda di esercizi di riabilitazione su misura. Il robot considera anche la data dell'infortunio per fornire indicazioni accurate sui tempi di recupero. Il tutto è integrato con un'applicazione che consente la registrazione degli utenti e la selezione del tipo di infortunio, rendendo l'esperienza semplice e interattiva.

In questa relazione verranno illustrati i metodi di lavoro adottati, le tecnologie impiegate, l'architettura del sistema e l'organizzazione del team NaoNexus.

Indice

1 Metodologia e tecniche utilizzate	5
1.1 GitHub	5
1.2 Organizzazione e metodologia Agile	5
1.3 Comunicazione nel team	6
2 Il progetto NaoRecovery	7
2.1 Obiettivi e finalità del progetto	7
2.2 Obiettivi e finalità del progetto	7
2.3 NAO come assistente medico per atleti	8
2.4 La struttura tecnica	9
3 Social e comunicazione visiva	15
3.1 Brand identity e logo	15
3.2 Il nostro partner, Steeb Basket	16
3.3 Contenuti, font e palette colori	18
4 Componenti del team	20
4.1 Team Coding	21
4.2 Team Social	21
5 Conclusioni e ringraziamenti	23

1. Metodologia e tecniche utilizzate

Il team NaoNexus è composto da diversi membri, perciò era importante tenere traccia dei compiti da svolgere, garantire una comunicazione fluida e assicurarsi che tutti lavorassero sull'ultima versione del software. Per questo motivo, il team ha utilizzato principalmente quattro strumenti: GitHub, Google Sheets, Discord e Microsoft Teams.

1.1. GitHub

GitHub è una piattaforma cloud-based per la gestione del codice sorgente, che permette di salvare e tenere traccia di tutte le modifiche apportate al software. Grazie al sistema di controllo di versione, ogni modifica viene registrata e può essere annullata in caso di errori. Utilizzando il branching e il merging, ogni membro ha potuto lavorare in modo sicuro su parti separate del codice senza interferire con la versione principale.

1.2. Organizzazione e metodologia Agile

Per coordinare le attività e gestire l'uscita dei contenuti social, il team ha utilizzato Google Sheets condivisi, con calendari per programmare i post e tenere sotto controllo le scadenze.

Il progetto è stato gestito seguendo la metodologia Agile, che consiste nel suddividere il lavoro in piccole parti da completare in sequenza. Questo approccio permette di monitorare frequentemente i progressi, ricevere feedback e apportare modifiche in modo rapido ed efficiente. In questo modo, il team ha potuto lavorare in modo collaborativo e flessibile, migliorando continuamente il progetto.

1.3. Comunicazione nel team

Per la comunicazione quotidiana e il confronto tra i membri, è stato utilizzato Discord, che ha permesso di scambiare idee rapidamente attraverso canali vocali e di testo. Inoltre, per la condivisione di file di grandi dimensioni, come video e materiale grafico, il team ha utilizzato Microsoft Teams, che ha garantito stabilità e ordine nella gestione dei documenti.

2. Il progetto NaoRecovery

2.1. Obiettivi e finalità del progetto

La NaoChallenge è una competizione internazionale rivolta alle scuole superiori, ideata per stimolare l'interesse degli studenti nei confronti della robotica, della programmazione e delle nuove tecnologie. Ogni anno, i team partecipanti affrontano una sfida diversa, progettando e realizzando soluzioni innovative che impiegano il robot umanoide NAO per risolvere problemi concreti ispirati al mondo reale. La competizione non valuta solo le capacità tecniche, ma anche l'originalità del progetto, la collaborazione nel team e la presentazione finale.

Per l'edizione 2025, il team NaoNexus ha scelto di concentrarsi sul tema della riabilitazione sportiva, con particolare attenzione al recupero fisico dei giocatori di basket dopo un infortunio. L'obiettivo è stato quello di sviluppare un sistema intelligente e accessibile, capace di affiancare gli atleti nel loro percorso di riabilitazione, offrendo supporto motivazionale, monitoraggio dei progressi e interazioni personalizzate.

2.2. Obiettivi e finalità del progetto

L'obiettivo del progetto NaoRecovery è sviluppare un assistente medico robotico pensato per supportare gli atleti nella fase di riabilitazione post-infortunio. L'idea nasce dalla volontà di rendere questo percorso più coinvolgente, accessibile e interattivo, sfruttando le potenzialità del robot NAO.

NAO è in grado di dialogare vocalmente con l'atleta, che può descrivere a parole il proprio infortunio. Le informazioni vengono poi elaborate da un sistema basato su OpenAI, che interpreta il linguag-

gio naturale e identifica l'infortunio più probabile, facendo riferimento a un database di patologie. A partire da questo, vengono generate delle informazioni personalizzate: i tempi di recupero stimati e una serie di esercizi riabilitativi adatti alla condizione dell'atleta.

Una delle funzionalità più importanti del sistema è la capacità di NAO di mostrare fisicamente alcuni degli esercizi, rendendo più chiaro come eseguirli correttamente. Questa interazione contribuisce a rendere la riabilitazione più efficace e motivante, anche in assenza di un fisioterapista.

Tutti gli esercizi vengono salvati e associati al profilo dell'atleta, ma non possono essere modificati né valutati per mantenere la coerenza del percorso riabilitativo.

A completamento del sistema è stata realizzata un'app in Flutter, pensata per gestire le squadre e i rispettivi giocatori, ciascuno associato a un infortunio. Tutte le informazioni sono archiviate in un database centralizzato, accessibile anche tramite una dashboard in HTML e JavaScript.

Questo progetto unisce tecnologie avanzate in un'unica soluzione che rende la riabilitazione più immediata, personalizzata e sempre disponibile.

2.3. NAO come assistente medico per atleti

Il ruolo del robot NAO all'interno del progetto è quello di assistente medico vocale. L'interazione avviene interamente tramite voce: l'atleta descrive il proprio infortunio, e NAO risponde con un piano riabilitativo creato su misura.

Il funzionamento del sistema segue queste fasi principali:

1. Raccolta delle informazioni

L'atleta comunica verbalmente il proprio infortunio a NAO, in modo naturale e semplice.

2. Interpretazione dell'infortunio

Le parole vengono inviate a una funzione basata su OpenAI, che interpreta la descrizione e identifica l'infortunio, confrontandolo con un database già esistente.

3. Generazione della risposta

In base al tipo di infortunio rilevato, NAO fornisce:

- il nome dell'infortunio;
- il metodo di recupero suggerito;
- i tempi di recupero stimati;
- una scheda di esercizi personalizzati per quell'atleta.

4. Esecuzione e supporto

NAO spiega gli esercizi e ne mostra alcuni tramite movimenti fisici, rendendo l'esperienza più intuitiva.

5. Salvataggio nel database

Gli esercizi vengono associati all'atleta e archiviati, in modo che siano disponibili per consultazioni successive. Tuttavia, l'atleta non può modificare né commentare gli esercizi assegnati.

2.4. La struttura tecnica

Il progetto è stato costruito su una struttura modulare, composta da vari elementi che lavorano insieme:

- **Python 3:** usato per la logica principale del sistema, come l'elaborazione dei dati e la gestione delle API.

```

def nao_audiorecorder(sec_sleep):
    data = {"nao_ip":nao_ip, "nao_port":nao_port, "nao_user":nao_user,
            "nao_password":nao_password, "sec_sleep":sec_sleep}
    url = "http://127.0.0.1:5011/nao_audiorecorder/" + str(data)
    response = requests.get(url, json=data, stream=True)

    local_path = f'recordings/microphone_audio.wav'
    if response.status_code == 200:
        with open(local_path, 'wb') as f:
            for chunk in response.iter_content(chunk_size=8192):
                if chunk:
                    f.write(chunk)
        logger.info("File audio ricevuto: " + str(response.status_code))
    else:
        logger.error("File audio non ricevuto: " + str(response.status_code))

    while True:
        speech_recognition = SpeechRecognition(local_path)
        if (speech_recognition.result != None or speech_recognition.result != ''):
            break

    logger.info("nao_audiorecorder: " + str(speech_recognition.result))
    return str(speech_recognition.result)

```

- **Python 2:** utilizzato per comunicare con il robot NAO, tramite le API ufficiali NAOqi. I due ambienti devono essere eseguiti contemporaneamente.

```

@app.route('/nao_audiorecorder/<params>', methods=['GET'])
def nao_get_audio(params):
    if [params != None and params != '']:
        if request.method == "GET":
            try:
                #("nao_ip":value, "nao_port":value, "nao_user":value, "nao_password":value, "sec_sleep":value)
                json = eval(params)
                nao_ip = json['nao_ip']
                nao_port = json['nao_port']
                nao_user = json['nao_user']
                nao_password = json['nao_password']
                sec_sleep = json['sec_sleep']

                audio_device_proxy = AlProxy("AlAudioRecorder", nao_ip, nao_port)
                remote_path = "/data/home/nao/recordings/microphones/microphone_audio.wav" # sul nao
                sample_rate = 16000
                # Registra l'audio dal microfono del NAO per 'sec_sleep' secondi
                audio_data = audio_device_proxy.startMicrophonesRecording(remote_path, "wav", sample_rate, [0, 0, 1, 0])
                time.sleep(sec_sleep)
                audio_device_proxy.stopMicrophonesRecording()

                # Connessione SSH al Nao
                try:
                    transport = paramiko.Transport((nao_ip, 22))
                    transport.connect(username=nao_user, password=nao_password)
                    scp = paramiko.SFTPClient.from_transport(transport)
                    local_path = "recordings/microphone_audio.wav"
                    scp.get(remote_path, local_path)
                    scp.close()
                    transport.close()
                except Exception as e:
                    logger.error(str(e))
                    return jsonify({'code': 500, 'message': str(e)}), 500
                else:
                    audio_device_proxy = None
                    return send_file(local_path, as_attachment=True)
            except Exception as e:
                logger.error(str(e))
                return jsonify({'code': 500, 'message': str(e)}), 500
            else:
                return jsonify({'code': 500, 'message': 'methods error'}), 500
        else:
            return jsonify({'code': 500, 'message': 'params error'}), 500
    else:
        return jsonify({'code': 500, 'message': 'params error'}), 500

```

- **Server locale:** Il robot NAO è programmato utilizzando il linguaggio di programmazione Python, con il codice di base diviso in due componenti: uno scritto in Python 2 e l'altro in Python 3.

Il componente Python 2 è responsabile della comunicazione diretta con il robot, gestendo i suoi sensori, attuatori e controlli di basso

livello. Questa scelta è dovuta al fatto che la libreria NAOqi, essenziale per l’interfacciamento con il robot NAO, è disponibile solo in Python 2.

Il componente Python 3 funge da ambiente principale per le integrazioni dell’utente e ospita un server Flask che facilita la comunicazione tra il robot e l’applicazione complementare.

Entrambi i componenti vengono eseguiti contemporaneamente su un server locale. Il modulo Python 3 richiama funzioni specifiche definite nel modulo Python 2, che a loro volta gestiscono la comunicazione in tempo reale con il robot NAO. Questa architettura garantisce un sistema modulare ed efficiente, in grado di interagire e controllare in modo reattivo.

- **Dashboard web:** Per supportare il processo di sviluppo e facilitare la gestione interna dei dati, abbiamo creato NaoDashboard, una dashboard personalizzata realizzata con HTML, CSS e JavaScript. Questo strumento è destinato al team di programmazione e fornisce una visione chiara e organizzata dei dati del sistema, aiutandoci a monitorare le prestazioni, eseguire il debug delle funzionalità e interagire con il database in modo più efficiente.

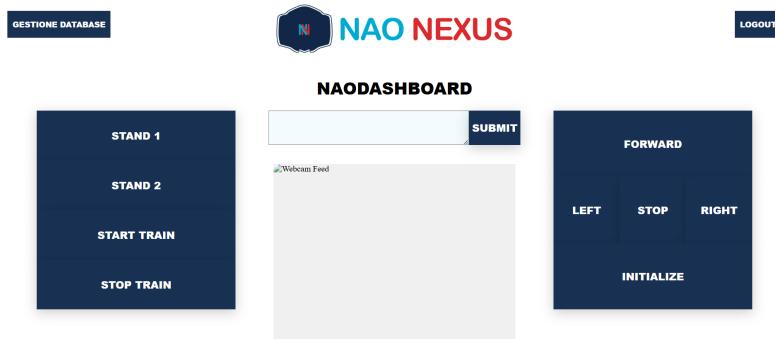
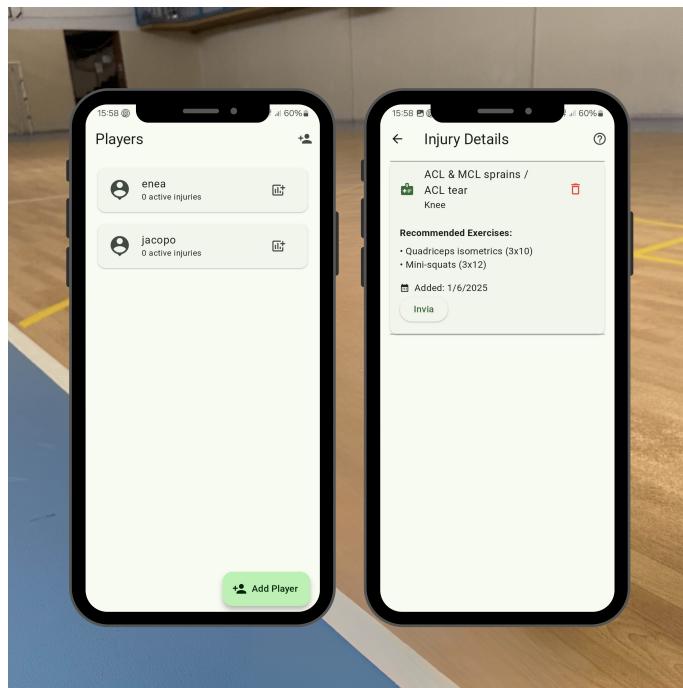


Figura 1: Dashboard web del progetto NaoRecovery

- **App Flutter:** Tramite l'app complementare, i giocatori possono monitorare lo stato del loro percorso riabilitativo e visualizzare i progressi degli esercizi assegnati. Inoltre, hanno la possibilità di chiamare il robot NAO, che fornisce un breve aggiornamento personalizzato sullo stato di recupero, offrendo incoraggiamento e rafforzando il coinvolgimento durante tutto il percorso riabilitativo.



- **Database Cloud Firestore di Google Firebase:** I nostri dati sono archiviati in modo sicuro nel Cloud Firestore di Google Firebase, un database cloud NoSQL scalabile progettato per la sincronizzazione in tempo reale e il supporto offline. L'utilizzo di Cloud Firestore ci consente di gestire e aggiornare in modo efficiente i dati di riabilitazione su più dispositivi, garantendo che gli utenti e il sistema abbiano sempre accesso alle informazioni più aggiornate. Questo approccio basato sul cloud migliora inoltre l'affidabilità, la

sicurezza e l'accessibilità dei dati, semplificando al contempo la manutenzione e la scalabilità del backend man mano che il progetto cresce.

Il database può essere gestito direttamente tramite la nostra Nao-Dashboard interna, fornendo al team di programmazione accesso e controllo in tempo reale sui dati archiviati. Inoltre, il robot NAO comunica con il database Cloud Firestore tramite il componente Python 3, consentendo un'interazione fluida tra il robot e il cloud per l'aggiornamento e il recupero delle informazioni rilevanti durante il processo di riabilitazione.

```
def db_add_players(players_list):
    results = []
    for player_name in players_list:
        try:
            player_name = player_name.strip().lower()
            if not player_name:
                results.append({'success': False, 'message': 'Player name is required.'})
                continue
            doc_ref = db.collection('players').document(player_name)
            if doc_ref.get().exists():
                results.append({'success': False, 'message': f'Player "{player_name}" already exists.'})
                continue
            doc_ref.set({
                'Injury list': [],
                'Time': 0,
                'Exercise list': [],
                'Last date': datetime.now()
            })
            results.append({'success': True, 'message': f'Player "{player_name}" added successfully.'})
        except Exception as e:
            results.append({'success': False, 'message': f'Error with player "{player_name}": {str(e)}'})
    for result in results:
        print(result)
```

- **NaoAi:** Il robot NAO sfrutta l'intelligenza artificiale sia per elaborare i dati di input sia per interagire utilizzando un prompt conversazionale simile a quelli utilizzati nelle comuni interfacce di chat basate su intelligenza artificiale. Genera inoltre risposte audio tramite sintesi vocale, consentendo una comunicazione naturale e coinvolgente con gli utenti. Per raggiungere questo obiettivo, utilizziamo le API di OpenAI, tra cui ChatGPT, consentendo al robot di fornire interazioni intelligenti e contestuali che migliorano l'esperienza riabilitativa.

Esempio generatore testo:

```

def nao_ai(user_input):
    "Genera una risposta AI usando ChatGPT"
    try:
        add_to_history("user", user_input)

        response = client.chat.completions.create(
            model="gpt-4o", # Usa un modello valido
            messages=conversation_history,
            temperature=0.7,
            max_tokens=133
        )

        ai_response = response.choices[0].message.content
        add_to_history("assistant", ai_response)

        return ai_response
    except Exception as e:
        print(f"Errore nella generazione della risposta: {e}")
        return "Mi dispiace, ho avuto un problema a generare una risposta."

```

Esempio generatore audio:

```

def audio_generator(testo, nome_file):
    """
    Genera un file audio da un testo usando OpenAI TTS
    e lo salva nella cartella specificata.
    """
    # Percorso di destinazione
    directory = "[percorso_file_audio]"
    os.makedirs(directory, exist_ok=True)

    if not nome_file.endswith(".mp3"):
        nome_file += ".mp3"

    percorso = os.path.join(directory, nome_file)

    try:
        response = client.audio.speech.create(
            model="tts-1",
            voice="fable",
            input=testo
        )

        with open(percorso, "wb") as f:
            f.write(response.content)

        print(f"Audio salvato in: {percorso}")
        return percorso
    except Exception as e:
        print(f"Errore nella generazione dell'audio: {e}")
        return None

```

3. Social e comunicazione visiva

3.1. Brand identity e logo

Il logo di NaoNexus rappresenta il fulcro della nostra idea: essere un “nexus”, un nesso che unisce tecnologie e applicazioni. La base del logo è la testa del robot NAO, il mezzo con cui lavoriamo, sovrastata da due “N” disposte una sull’altra in senso opposto, quasi a completarsi a vicenda. Questo simboleggia come la tecnologia (NAO) non possa esistere senza la sua applicazione pratica (Nexus) e viceversa.

La palette colori originale del logo si basa sul ciano, declinato in una tonalità più scura per la prima “N”, e un corallo complementare per la seconda “N”, ispirata alla “orange & teal vibe”. Questa combinazione è stata utilizzata in tutte le nostre grafiche e documentazioni ufficiali non tecniche.

Negli ultimi due anni, il logo ha subito alcune modifiche per riflettere temi e partnership:

- Nel 2023, sono state aggiunte due foglie in basso a destra per sottolineare il nostro impegno verso la sostenibilità ambientale.
- Nel 2024, le foglie sono state sostituite da un diamante, simbolo dello sponsor Swarovski, collegato alla realizzazione del progetto di quell’anno.

Rinnovo 2025 Con l’ingresso di una nuova squadra composta interamente da membri nuovi, il logo è stato rinnovato per rappresentare questo rinnovamento anche visivo. La nuova versione include una cornice che racchiude il simbolo, con i colori invertiti rispetto alla versione

precedente, sottolineando così il cambiamento e la freschezza del team attuale.

logo rinnovato:



logo con scritta bassa:



NAO NEXUS

logo con scritta laterale:



3.2. Il nostro partner, Steeb Basket

Per sviluppare un progetto realistico e mirato, il team NaoNexus ha collaborato con la Steeb Basket, una squadra amatoriale di pallacanestro fondata nel 2023. Il nome della squadra prende ispirazione dalla chiesa del Beato Carlo Steeb situata a San Michele Extra (Verona), sede dell'Associazione Sportiva Dilettantistica Circolo Parrocchiale Beato Carlo Steeb, di cui la squadra fa parte.

Il roster della stagione 2024/2025 è composto da 19 giocatori, con un'età compresa tra i 21 e i 29 anni, e da un allenatore. La squadra partecipa al campionato Open Serie A2 organizzato dal CSI di Verona (Centro Sportivo Italiano), una lega amatoriale che coinvolge centinaia di squadre in tutta Italia. In particolare, la sezione veronese del campionato conta 34 squadre, suddivise in due serie.

Durante la stagione sportiva precedente, la Steeb Basket ha ottenuto un importante risultato, vincendo il girone Playout della Serie A2. Per la stagione in corso, l'obiettivo della squadra è lottare per la promozione in Serie A1.

Gli allenamenti si svolgono presso la palestra del Liceo Stimate di Verona, mentre le partite casalinghe vengono giocate nella palestra del Liceo Copernico, anch'essa situata a Verona.

Grazie alla collaborazione con lo staff tecnico e i giocatori della Steeb Basket, il progetto NaoRecovery ha potuto basarsi su esigenze reali legate alla riabilitazione sportiva, testando funzionalità e interazioni del sistema in un contesto concreto e attinente al mondo dello sport amatoriale.



3.3. Contenuti, font e palette colori

Per la comunicazione del progetto NaoNexus 2025, il team social ha ideato una strategia mirata a valorizzare la presenza online e offline, utilizzando diversi tipi di contenuti per coinvolgere il pubblico e promuovere il lavoro svolto.

Tipologie di contenuti

- **Reel e video brevi:** creati per Instagram e TikTok, mostrano momenti salienti del progetto, dimostrazioni pratiche del robot NAO e dietro le quinte del team al lavoro.
- **Storie:** utilizzate per aggiornamenti quotidiani e contenuti più informali, coinvolgendo il pubblico in modo diretto e spontaneo.
- **Post scritti:** pubblicati su LinkedIn e Instagram, approfondiscono aspetti tecnici, novità del progetto e risultati ottenuti, mantenendo un tono professionale ma accessibile.

foto profilo ig:

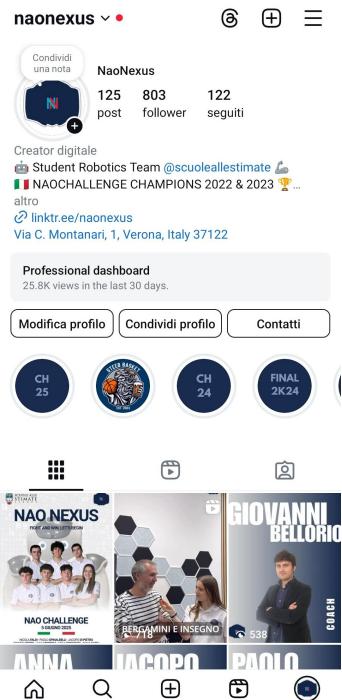
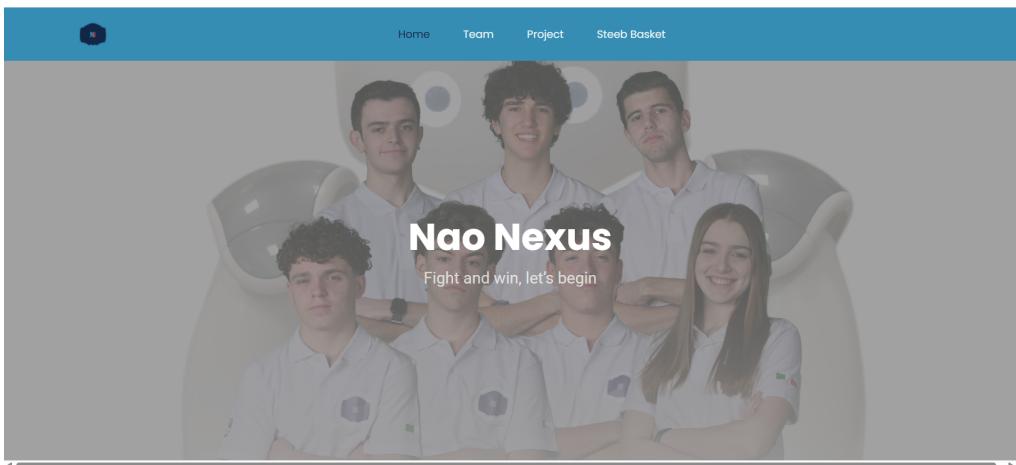


foto sito:



Calendario editoriale Per garantire una presenza costante e organizzata sui social media, è stato creato un calendario editoriale

che pianifica la pubblicazione dei contenuti, bilanciando momenti di promozione, aggiornamenti tecnici e interazione con la community.

Font utilizzati

- Per la scritta vicina al logo è stato scelto il font **Arial Rounded MT Bold**, per la sua leggibilità e modernità, che si adatta bene all'identità visiva del progetto.
- Per il merchandising e le grafiche promozionali è stato scelto il font **Anton**, caratterizzato da uno stile forte e deciso, accompagnato dal **Arial Rounded MT Bold** per testi secondari, assicurando coerenza e impatto visivo.

Palette colori La palette colori ufficiale del progetto riprende i toni del logo e si compone principalmente di:

- Rosso corallo: `#d9413d`
- Blu scuro: `#1a2b50`
- Azzurro medio: `#368db3`

4. Componenti del team

Il team NaoNexus 2025 è composto da sette studenti della classe 3^a Scienze Applicate dell'Istituto Stimate di Verona. Tutti i membri sono nuovi al progetto, ma hanno saputo collaborare con spirito di squadra ed entusiasmo, suddividendo in modo chiaro le responsabilità tra l'area coding e quella social.

4.1. Team Coding

Il team coding ha curato la parte tecnica del progetto, occupandosi dello sviluppo dell'applicazione, della gestione del server e dell'integrazione con il robot NAO. Il lavoro è stato suddiviso tra i seguenti membri:

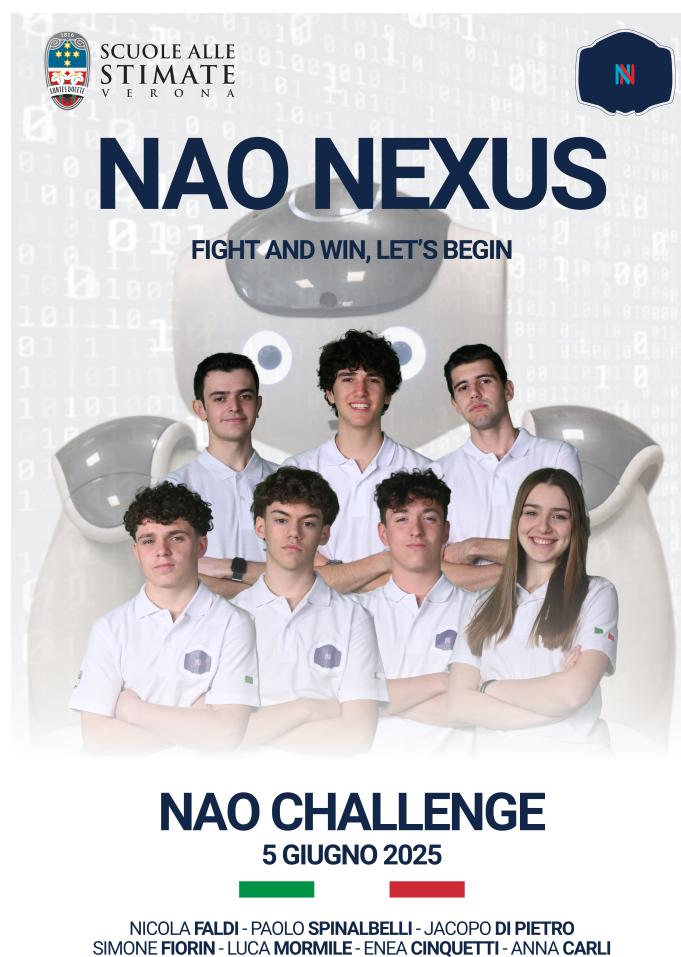
- **Jacopo Di Pietro** – *Sviluppatore App*: si è occupato della creazione dell'applicazione in Flutter per la gestione degli atleti e degli infortuni.
- **Luca Mormile** – *Sviluppatore App*: ha collaborato allo sviluppo dell'interfaccia e delle funzionalità dell'app.
- **Enea Cinquetti** – *Responsabile Server*: ha sviluppato la dashboard web e ha curato gli esercizi del NAO.
- **Nicola Giovanni Faldi** – *Integrazione Server*: ha programmato il server del robot NAO, l'interazione vocale con l'atleta, il collegamento con il database Firestore e le funzionalità basate sull'intelligenza artificiale.

4.2. Team Social

Il team social ha gestito l'identità visiva e la comunicazione del progetto, promuovendo il lavoro svolto attraverso i social media e curando la documentazione multimediale. Ogni membro ha contribuito con un ruolo specifico:

- **Anna Carli** – *Team Leader, Sito Web e Gestione Social*: ha coordinato il lavoro del gruppo, curato il sito web del team e gestito i contenuti sui social.

- **Paolo Spinalbelli** – *Foto e Social Media*: ha documentato il progetto con materiale fotografico e ha contribuito alla gestione dei canali social.
- **Simone Fiorin** – *Video*: ha realizzato i video di presentazione e dimostrazione del progetto, occupandosi di riprese e montaggio.



5. Conclusioni e ringraziamenti

La Nao Challenge è stata un'esperienza intensa e formativa per tutto il team. Siamo riusciti a sviluppare un progetto innovativo che sfrutta le capacità del robot NAO nel campo della riabilitazione sportiva, dimostrando le nostre competenze tecniche, organizzative e creative.

Ringraziamo calorosamente la squadra Steeb Basket per averci supportato e per aver collaborato con noi in qualità di partner tecnico, offrendoci spunti concreti legati al mondo dell'atletica e della riabilitazione.

Noi tutti membri del team ringraziamo particolarmente il professor Giovanni Bellorio (*Il Supremo*) per averci dato la possibilità di partecipare a questo progetto coinvolgente. Lo ringraziamo inoltre per averci seguito passo dopo passo nella realizzazione dell'intero lavoro, istruendoci e guidandoci nel mondo della progettazione e del lavoro di gruppo.