Phoenix

Specifiche Agente AI RehabSphere

[Progetto "*RehabSphere - Cutting-Edge Virtual Reality Platform for Comprehensive Patient-Centric Rehabilitation*"; Codici di progetto CUP E79J24004360004 COR 23005670 e COVAR 1499428. Progetto finanziato dall'Unione europea – Next Generation EU - PNRR – Missione 4, Componente 2, Investimento 1.5 "Ecosistemi dell'Innovazione", Programma di Ricerca dell'Ecosistema dellInnovazione Interconnected Nord-Est Innovation Ecosistem I-NEST, codice identificativo ECS00000043 e CUP E63C22001030007Spoke 2 (Health, Food & Lifestyles) – Università di Trento.]

Phoenix

Indice

Phoenix

Proprietà del documento

Registro delle modifiche

Edizione	Revisione	Data	Modifiche
01	00	23.04.2025	

Scopo

Questo documento definisce le specifiche del sistema Agente AI di RehabSphere.

Destinatari

Il seguente documento è destinato ai fornitori per l'elaborazione di una proposta tecnico economica per la realizzazione di un Agente AI RehabSphere.

Acronimi e definizioni

VR: Virtual Reality – Realtà Virtuale

AR: Augmented Reality – Realtà Aumentata

API: Application Programming Interface

GDPR: General Data Protection Regulation – Regolamento Europeo per la protezione dei dati

KPI: Key Performance Indicator – Indicatore chiave di prestazione

Phoenix

1 Descrizione di RehabSphere

Il progetto RehabSphere si colloca nel settore della riabilitazione neuromotoria e post-operatoria, sfruttando tecnologie di realtà virtuale (VR), intelligenza artificiale (IA) e machine learning (ML) per migliorare l'efficacia dei trattamenti fisioterapici. L'approccio tradizionale alla riabilitazione prevede sessioni supervisionate in presenza, con limitazioni legate alla disponibilità di fisioterapisti, ai costi e agli spostamenti dei pazienti. RehabSphere propone una soluzione innovativa che combina:

- Ambienti VR immersivi e interattivi per migliorare l'engagement dei pazienti durante gli esercizi.
- Interfaccia utente (UI) accessibile e intuitiva, adatta a diverse tipologie di utenti.
- Architettura software scalabile, con backend a microservizi e API RESTful per la gestione dei dati e delle sessioni.
- Agente intelligente basato su NLP, in grado di fornire supporto informativo e assistenza in linguaggio naturale.
- Database progettato per archiviare dati strutturati e non strutturati (video, immagini, parametri biometrici).
- Algoritmi di machine learning per l'analisi di grandi volumi di dati clinici al fine di identificare pattern e predire i risultati dei trattamenti, migliorando così l'efficacia delle terapie.

Grazie a RehabSphere, i pazienti potranno svolgere la riabilitazione in modo più comodo, personalizzato ed efficace, riducendo i tempi di recupero, aumentando la motivazione e migliorando l'aderenza al percorso terapeutico, anche da casa.

2 Scopo

Lo scopo dell'Agente AI è abilitare, all'interno dell'applicazione VR RehabSphere, un'interazione naturale, sicura e contestuale tra il paziente (o il terapista) e il sistema, fornendo spiegazioni degli esercizi, risposte mirate alle domande durante la sessione e feedback immediato sull'esecuzione. L'agente combina il riconoscimento vocale (ASR), la sintesi vocale (TTS) e un motore linguistico potenziato da Retrieval-Augmented Generation (RAG) per produrre risposte fondate su conoscenza verificata e sul contesto specifico della sessione.

Gli obiettivi principali che l'Agente dovrà garantire sono:

- Chiarezza e aderenza clinica: spiegare scopo, corretta esecuzione, errori frequenti e precauzioni dell'esercizio in corso, evitando raccomandazioni extra-cliniche o fuori protocollo.

- Personalizzazione contestuale: modulare le risposte in base a piano/playlist assegnati, livello di difficoltà, storico di aderenza ed eventuali limiti/controindicazioni note.
- Interazione fluida in VR: ridurre la latenza percepita (streaming della risposta testuale e vocale), sincronizzare la voce con l'avatar (lip-sync), supportare segnali visivi/indicatori in-scene.
- Affidabilità e sicurezza: garantire continuità del servizio, gestione del fallback (risposta "sicura" o rinvio al terapista) quando la confidenza è bassa o i dati non sono disponibili, e aderire a principi GDPR (minimizzazione, audit, diritti dell'interessato).
- Misurabilità: fornire metriche e log utili a misurare utilità delle risposte, errori, latenza e tasso di fallback per miglioramenti iterativi.

Sono da considerarsi nello scope del progetto:

- Comprensione di domande semplici o composte relative a esercizi, ergonomia del movimento, ritmo/respirazione, durata, numero di ripetizioni, sensazioni attese.
- Spiegazioni coerenti con il piano assegnato e con la fase dell'esercizio (setup, esecuzione, cool-down), con riferimenti a posture, range of motion e segnali di stop.
- Feedback generico sull'esecuzione quando disponibile (es. "mantieni la schiena dritta"), senza sostituire le valutazioni cliniche del terapista e senza diagnosi.

Sono invece da considerarsi out-of-scope:

- Diagnosi mediche, prescrizioni terapeutiche, interpretazione di esami clinici, decisioni di triage.
- Consigli nutrizionali o farmacologici individualizzati.
- Qualsiasi contenuto non coperto da linee guida e materiali inclusi nel corpus validato.

3 Contesto

L'Agente AI opera all'interno dell'app VR eseguita su visori Meta Quest (o compatibili), rappresentato da un avatar 3D che dialoga con l'utente tramite voce e gestualità minima sincronizzata. L'interazione è progettata per sessioni brevi e ripetute, con attenzione a comfort, focalizzazione e riduzione del carico cognitivo.

3.1 Ambiente applicativo

- Runtime VR: l'app gestisce cattura microfono, invio domanda e stato dell'esercizio all'AI Gateway; riceve testo/voce e segnali per lip-sync e overlay (es. evidenziazione articolazioni/posture, indicatori "più lento", "mantieni").
- Rete: connettività HTTPS (REST) per operazioni sincrone e, se attivata, WebSocket per streaming della risposta (testo parziale e marcatori temporali TTS).

- Esperienza utente: l'agente evita l'over-talking, rispetta i tempi dell'esercizio e usa un linguaggio semplice, con possibilità di richiedere "spiega meglio" o "mostra" per ricevere chiarimenti o aiuti visivi.

3.2 Dati clinici e piani/playlist

Il Gestionale RehabSphere (sistema esterno) fornisce all'agente il contesto minimo necessario: identificativo pseudonimizzato del paziente, piano/playlist assegnati, esercizio corrente, parametri di sessione (durata, ripetizioni, limiti).

I dati sono minimizzati e aggiornati prima o durante la sessione; l'agente non accede a dati sanitari superflui. Le scritture (es. completamento esercizio) restano responsabilità del gestionale.

3.3 Knowledge semantica e Vector Database

La conoscenza su esercizi, anatomia, biomeccanica, linee guida e FAQ validate è organizzata in un Vector Database con metadati (fonte, versione, lingua, validazione, esercizio correlato).

Il Retriever esegue similarity search con filtri per esercizio corrente e profilo utente; i passaggi recuperati vengono inseriti nel prompt (RAG) affinché il LLM citi o si ancori al contenuto approvato, riducendo il rischio di allucinazioni.

La base conoscitiva è versionata e soggetta a governance (chi può inserire/aggiornare, chi approva, tracciabilità delle modifiche).

3.4 LLM e RAG

Il LLM genera la risposta in linguaggio naturale (IT come must, EN come should), mantenendo tono empatico e istruzioni chiare.

Il prompt è costruito dinamicamente con: (a) contesto sessione, (b) estratti dal Vector DB, (c) policy cliniche (regole di guardrail: cosa può/non può dire), (d) segnali di confidenza.

Se la confidenza è bassa o i contenuti recuperati non sono sufficienti, l'agente degrada su una risposta sicura (es. rinvia al terapista, ricorda le regole generali di sicurezza) senza inventare contenuti.

4 Specifiche dell'architettura

L'architettura dell'agente prevede almeno i seguenti componenti

4.1 ASR (Automatic Speech Recognition)

Scopo	Convertire l'audio catturato in VR in testo con punteggiatura e timecode, preservando i termini tecnici della riabilitazione.
Input/output	**Input:** stream audio PCM/Opus (16–48 kHz), lingua IT (obbligatoria), eventuale EN (opzionale); segnali VAD (voice activity detection). **Output:** trascrizione testuale con timestamp per parola/frase, punteggio di confidenza, eventuali ipotesi alternative (N-best), tag di punteggiatura
Logica/algoritmi	Hotword/keyword boosting per lessico rehab (es. "abduzione spalla", "rachide", nomi esercizi). VAD per ridurre latenza e inviare chunk incrementali (streaming). Normalizzazione numerica (es. "quindici ripetizioni" → "15 ripetizioni").
Interfacce & configurazione	**Protocollo:** WebSocket (streaming) o REST (batch). **Parametri:** lingua, punteggiatura automatica on/off, boost vocabolario, soglia confidenza, dimensione buffer. **Output events:** `asr.partial`, `asr.final`.
Guasti & fallback	In caso di perdita rete/alto rumore: passaggio a input testuale o messaggio standard ("non ho capito, ripeti lentamente").
Metriche operative	Latenza media, WER su set di controllo, tasso di `asr.no-speech`, durata media enunciazione.

4.2 NLP Orchestrator

Scopo	Trasformare la trascrizione ASR in una query strutturata: identificare intento, entità cliniche (distretti corporei, esercizi, attrezzi), slot (ripetizioni, durata), e stimare il contesto (fase esercizio, obiettivo).
Input/output	**Input:** testo ASR + metadati sessione (esercizio corrente, lingua, ID paziente pseudonimo). **Output:** struttura JSON `{intent, entities, slots, confidence, context}`.
Logica/algoritmi	Pipeline: *language detection* (sanity check), *tokenization*, *NER* (distretti/esercizi), *intent classifier*, *slot filling*, *normalizzazione* (sinonimi → dizionario).

	Regole di safety: blacklist/whitelist, reindirizzamento (es. domande cliniche fuori perimetro → "rivolgersi al terapista"). Disambiguazione con il contesto dell'esercizio corrente (es. "quanto manca" → tempo residuo set).
Interfacce & configurazione	**REST**: `POST /nlp/parse` (batch o singolo turno). **Feature flags**: attiva/disable modelli NER specifici, dizionari clinici, soglie confidenza. **Dizionari** aggiornabili (termini esercizi, sinonimi anatomici).
Guasti & fallback	Se confidenza < soglia: domanda chiarificatrice brevissima ("Intendi durata o ripetizioni?") o fallback a risposta generale sicura.
Metriche operative	Accuracy intent/NER, tasso domande non interpretabili, tempo medio parsing, casi di disambiguazione.

4.3 Retriever su Vector DB

Scopo	Recuperare passaggi pertinenti dal corpus validato (esercizi, anatomia, linee guida, FAQ) filtrati sul contesto (esercizio corrente, profilo).
Input/output	**Input:** query strutturata dal NLP + filtri (exercise_id, lingua, livello, validazione). **Output:** lista ordinata di top-k passaggi `{text, source, score, metadata}`.
Logica/algoritmi	**Embedding** + **similarity search** (cosine/dot product); **re-ranking** opzionale. **Filtri**: lingua, versione documento, provenienza (clinico validato), compatibilità esercizio/fase. **Diversificazione** dei risultati (MMR) per evitare passaggi duplicati.
Interfacce & configurazione	**REST**: `POST /retriever/query` con `k`, filtri, namespace. **Admin**: `/vectors/upsert`, `/vectors/delete`, validazione e pubblicazione corpus (stati: draft → reviewed → approved).
Guasti & fallback	Nessun risultato: ritorno di prompt template neutrale (regole di sicurezza, principi generali). Indice non disponibile: fallback a knowledge locale minimale in app (messaggi standard).

Metriche operative	tasso "no-result", latenza media, qualità percepita (valutazione terapisti), copertura per esercizio/lingua.

4.4 LLM Layer

Scopo	Costruire il prompt con contesto (dati minimi paziente, top-k del Retriever, stato sessione) e generare una risposta naturale, sicura e allineata al protocollo.
Input/output	**Input:** `{query, top_k_passages, session_state, policy}`. **Output:** testo (streaming o full), citazioni/sorgenti (se richieste), confidenza, eventuali azioni (es. "mostra overlay postura").
Logica/algoritmi	**Template RAG** con sezioni: istruzioni di ruolo, regole cliniche ("do & don't"), contesto esercizio, estratti citabili, stile/tono. **Guardrails**: blocchi lessicali, classifica rischio contenuto, risposte **sicure** se confidenza bassa o topic out-of-scope. **Post-processing**: semplificazione linguaggio, liste puntate concise, callout "attenzione" quando pertinente.
Interfacce & configurazione	**REST/WebSocket**: `POST /llm/chat` (stream=True/False). Parametri: max tokens, temperatura, penalità ripetizione, lingua preferita, modalità "concisa" per VR. **Policy pack** versionato (linee guida validate, disclaimer).
Guasti & fallback	Timeouts/API error: messaggio breve standard + suggerimento di ripetere/attendere il terapista. Confidenza bassa: risposta de-risked (generalità, non prescrittiva).
Metriche operative	Latenza generazione, lunghezza media risposta, tasso interventi guardrail, valutazioni utilità da terapisti.

4.5 TTS (Text-To-Speech) con lip-sync

Scopo	Convertire la risposta testuale in voce naturale sincronizzata con l'avatar (visemi/phonemi) e coordinata con eventuali indicatori visivi.
Input/output	Input: testo (IT/EN), marcatori semantici (pause, enfasi), timestamps per lip-sync. Output: stream audio (Opus/PCM) + viseme timeline per l'avatar; eventi `tts.start`, `tts.marker`, `tts.end`.

Logica/algoritmi	**Neural TTS** con voce selezionata (empatica, ritmo controllato). Inserimento pause strategiche e segmentazione frasi in VR (battute 2–4 s). **Prosodia**: enfasi su parole chiave (es. "lento", "stop", "mantieni").
Interfacce & configurazione	**WebSocket/REST**: `POST /tts/synthesize` (streaming preferito per bassa latenza). Parametri: voce, velocità, pitch, punteggiatura prosodica, lingua. Integrazione con avatar engine per applicare la viseme timeline.
Guasti & fallback	Se TTS non disponibile: mostra testo in overlay + sintetica vibrazione/indicatore visivo. Se visemi assenti: fallback a auto-lip semplificato dell'engine.
Metriche operative	Latenza primo chunk, durata media clip, drift lip-sync, tasso fallback a testo.

4.6 Flusso di Interazione

1. Il paziente formula una domanda vocale durante l'esercizio.
2. ASR converte l'audio in testo; l'NLP Orchestrator rileva intenti ed entità.
3. Il Retriever interroga il Vector DB (top-k) applicando filtri di sicurezza e contesto.
4. L'LLM genera la risposta con RAG e policy cliniche/tono controllati.
5. La risposta è resa vocale via TTS e sincronizzata con l'avatar (lip-sync), con eventuale supporto visivo all'esercizio.

5 Requisiti Funzionali (RF)

- RF-AI-001 – Comprensione domanda utente (intent, entità cliniche) con accuratezza ≥80% su set pilota.
- RF-AI-002 – Recupero conoscenza: top-k documenti da Vector DB con filtri per esercizio/paziente.
- RF-AI-003 – Costruzione prompt dinamico (RAG) con contesto clinico minimo necessario (data minimization).
- RF-AI-004 – Generazione risposta naturale in IT/EN; tono empatico, registri adeguati.

Phoenix

- RF-AI-005 – TTS naturale con lip-sync; latenza audio < 400 ms dalla risposta testuale.
- RF-AI-006 – Visual aiuti 3D sull'esercizio (se disponibili) in sincronia con la spiegazione.
- RF-AI-007 – Gestione errori/fallback: risposta sicura quando la confidenza è bassa; suggerimento di rivolgersi al fisioterapista.
- RF-AI-008 – Telemetry/Audit delle interazioni, con anonimizzazione.
- RF-AI-009 – Localizzazione: supporto IT (MUST), EN (SHOULD).
- RF-AI-010 – Policy di sicurezza: nessun consiglio medico oltre il perimetro validato; disclaimer automatico quando necessario.

6 Requisiti Non Funzionali (RNF)

- RNF-AI-001 – Latenza end-to-end Q→A ≤ 3 s (escluso rendering), p95.
- RNF-AI-002 – Affidabilità: disponibilità 99.5%; retry/backoff su dipendenze.
- RNF-AI-003 – Sicurezza & Privacy: TLS, RBAC, minimizzazione dati, cifratura at-rest/in-transit; log compliant GDPR.
- RNF-AI-004 – Osservabilità: metriche (latenza, tasso fallback, confidenza), trace distribuito, alert SLO breach.
- RNF-AI-005 – Manutenibilità: componenti containerizzati, versionamento modelli, feature flags per rollout controllato.

7 Interfacce

- I-001 – Vector DB: /vectors/upsert, /vectors/query; HNSW/IVF; filtri per namespace e metadati.
- I-002 – LLM API: completions/chat con prompt RAG; guardrails e moderazione.
- I-003 – App VR: WebSocket/HTTPS per invio domanda/contesto e ricezione risposta; eventi per lip-sync e overlay esercizi.
- I-004 – Gestionale: lettura minima dei dati del paziente necessari al contesto (piano, esercizio corrente, limiti).

8 Dati & Sicurezza

- Namespace Vector DB: patient_notes, exercises, clinical_guidelines, patient_faq.
- Metadati: fonte, classe contenuto, validazione clinica, lingua, timestamp, PII=false.
- Crittografia AES-256 at-rest; TLS 1.2+ in-transit; controllo accessi RBAC; audit immutabile.
- Gestione consensi, export per paziente, diritto all'oblio; retention configurabile.

9 KPI

- KPI-1: tasso fallback < 10% in uso standard; KPI-2: CSAT ≥ 4/5 su utenti pilota.