

Osteoarthritis Portal

Magnani, Matteucci, Montanari

Dipartimento di Informatica – Scienza e Ingegneria (DISI)
Alma Mater Studiorum – Università di Bologna

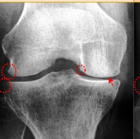
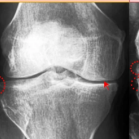
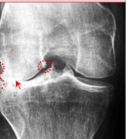
Sistemi Distribuiti

April 28, 2025

- **Applicazione web** per medici e pazienti
 - Caricamento e gestione delle radiografie
 - Diagnosi automatizzata
 - Visualizzazione e gestione delle diagnosi
 - Calendario e sistema di notifiche
 - Sistema di feedback e miglioramento continuo

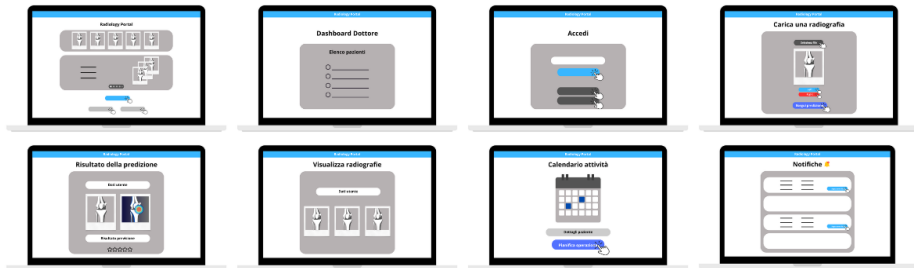
Caso di studio: osteoartrite e gradi Kellgren-Lawrence

• 5 Gradi della scala Kellgren-Lawrence

Grade 0 No OA	Grade 1 Doubtful OA	Grade 2 Mild OA	Grade 3 Moderate OA	Grade 4 Severe OA
				
No Osteophytes	Possible Osteophytes	Definite Osteophytes	Moderate Osteophytes	Large Osteophytes
No JSN	Doubtful JSN	Possible JSN	Definite JSN	Great JSN

- **Single-Page Application**
- **Applicazione AJAX**
 - livello di presentazione
 - livello dei servizi
 - livello di logica applicativa
 - livello di persistenza dei dati
- **Signature Stack**
 - Firestore e Google Cloud Storage - database non relazionale
 - Flask - framework js lato backend
 - Vue.js - framework js lato frontend
 - Node.js - ambiente di esecuzione
- **Agile** - metodologia di sviluppo

- **Analisi Target User**
- **Creazione di mockup**
- **Coinvolgimento medici**



● Obiettivi:

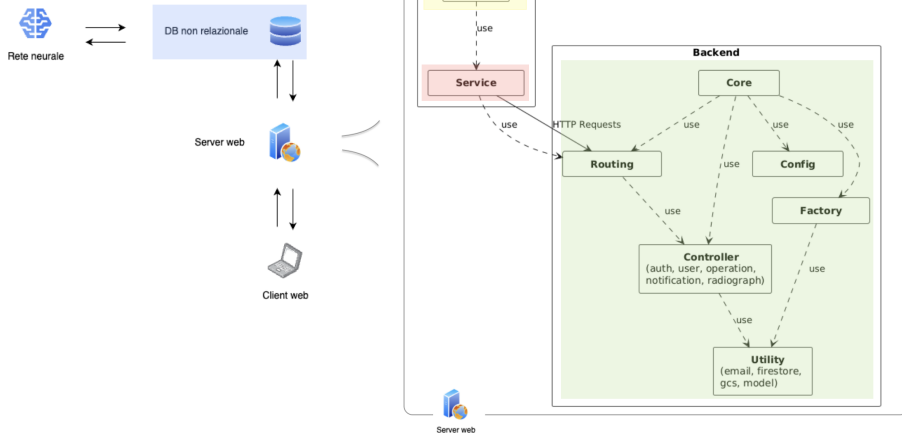
- Costruire un'interfaccia intuitiva e semplice da usare per qualsiasi utente
- Offrire un'esperienza utente soddisfacente e piacevole

● Principi seguiti:

- Responsive design
- KISS
- Less is More
- Uso di elementi grafici

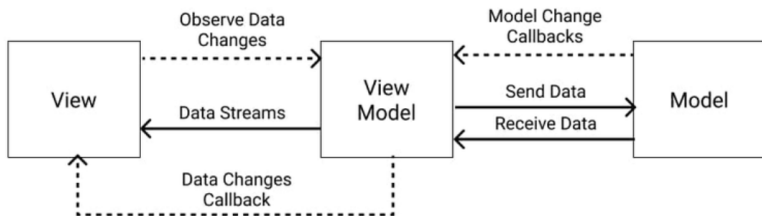
- **Test automatizzati**
- **Euristiche di Nielsen**
 - controllo e libertà
 - aiuto utente
 - prevenzione errori
 - riconoscimento più che ricordo
 - visibilità dello stato del sistema
 - corrispondenza con il mondo reale
 - consistenza e standard
 - progettazione minimalista
 - flessibilità ed efficienza
- **Cognitive walkthrough**

Struttura generale del sistema



Livello di presentazione: Modulo UI

- Framework: **Vue.js**
- Architettura: **Componenti modulari e reattivi**
- **Ottimizzazioni** per ridurre i tempi di caricamento
- **Supporto multi-utente**
- Pattern: **MVVM**

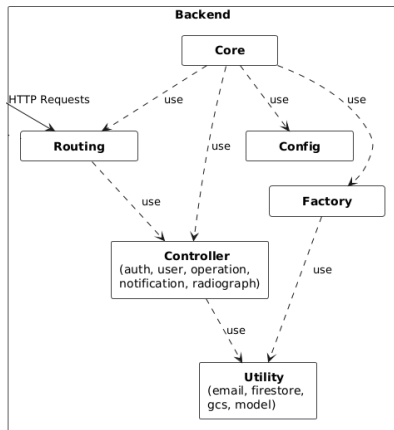


- Meccanismo **Remote Procedure Call**
- La comunicazione avviene tramite **richieste HTTP (API RESTful)**
- **Gestione unificata** (file `api_service.js`)

```
export const getPatientsFromDoctor(...) // GET
export const checkEmailVerification(...) // POST
export const patchNotifications(...) // PATCH
```

Livello di logica applicativa: Moduli Core, Config, Utility, Factory, Controller, Routing

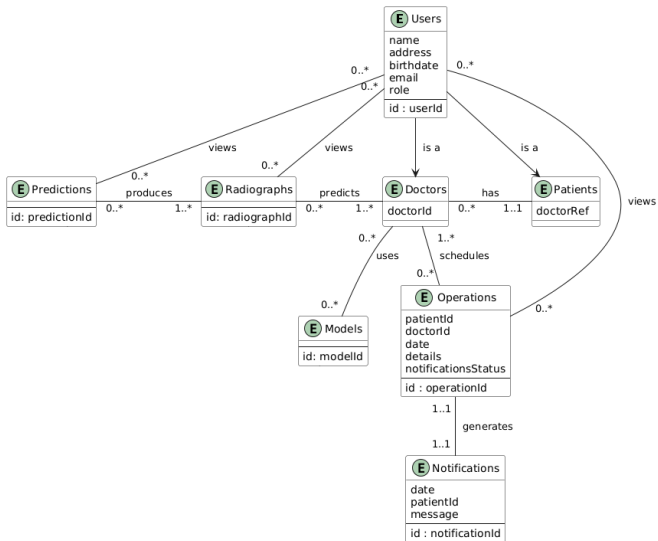
- **Modulo Core:** coordina tutte le operazioni e i servizi, responsabile di inizializzare il sistema.
- **Modulo Config:** gestisce la configurazione e le credenziali del sistema
- **Modulo Utility:** fornisce servizi di supporto per i diversi moduli
- **Modulo Factory:** crea e configura le classi di Utility
- **Modulo Controller:** coordina le operazioni richieste dagli utenti
- **Modulo Routing:** gestisce l'instradamento delle richieste



Livello di persistenza dei dati (1/3)

Entità:

- Users:
 - Doctors
 - Patients
- Radiographs
- Predictions
- Models
- Operations
- Notifications



Livello di persistenza dei dati (2/3)

- **Firebase Firestore e Firebase Authentication**

- Archivio non relazionale per utenti, notifiche e operazioni

The screenshot displays the Google Cloud Firestore console interface. At the top right, there are buttons for 'Visualizzazione riquadro' and 'Query Builder'. The breadcrumb navigation shows the path: Home > operations > WuZxX4Iqueme... In the top right corner of the console area, there is a link 'Altro in Google Cloud' with a dropdown arrow. The main content area is divided into three panes. The left pane shows a tree view with 'notifications', 'operations' (selected), and 'users'. The middle pane shows a list of documents under the 'operations' collection, including 'WuZxX4IquemerT8oH0LE' which is selected. The right pane shows the details of the selected document, including fields like 'createdAt', 'description', 'doctorId', 'notificationStatus', 'operationDate', and 'patientId'.

(default)	operations	WuZxX4IquemerT8oH0LE
+ Avvia raccolta notifications operations > users	+ Aggiungi documento 8vZ6MGzJJE0eehZMPCPL 6sAIFPbR2zw6jwDA4iNM 0ho81gPEebqonflqdPM1 P5N09x9T1w8jV3L6JIIk WuZxX4IquemerT8oH0LE > cLPiOzEje8WRC6mz9tFI o05dRRNpo8YxmyXvN0oB9	+ Avvia raccolta + Aggiungi campo createdAt: '2024-11-29T13:10:49.546242' description: "Varo" doctorId: "12345" notificationStatus: 'pending' operationDate: '2024-12-08T00:00:00' patientId: 'cFcgfvxbP8XMF5cVSqNZK3n8S2'

Livello di persistenza dei dati (3/3)

● Google Cloud Storage (GCS)

- Modello pre-addestrato
- Radiografie caricate dai medici
- Heat map generate dalla rete
- Risultati delle predizioni

● Accesso ai dati

- Ogni radiografia è associata a un singolo medico
- Concorrenza gestita con politica **Last Write Wins** in caso di accessi simultanei allo stesso profilo

Bucket > osteoarthritis-portal-archive 

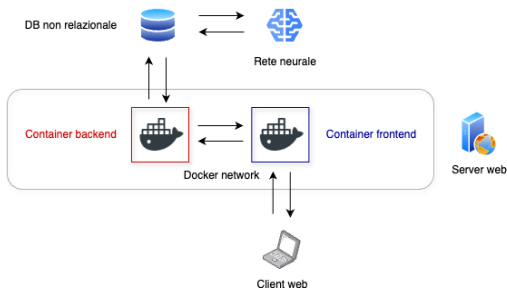
[CREA CARTELLA](#) [CARICA](#) [TRASFERIMENTO DEI DATI](#)

Filtra solo per prefisso nome   Filtra Filtra oggetti e cartelle

☐	Nome	Dimensioni	Tipo
☐	 .ipynb_checkpoints/	—	Cartella
☐	 9gsFIWp9Kkhhm0n7TSJbn11UA06...	—	Cartella
☐	 Dataset.zip	203,9 MB	application/zip
☐	 Dataset/	—	Cartella
☐	 MODELLO/	—	Cartella
☐	 TFRecord/	—	Cartella
☐	 dataset/	—	Cartella
☐	 google-cloud-dataproc-metainfo/	—	Cartella
☐	 notebooks/	—	Cartella

Containerizzazione

- **Frontend e Backend** sono containerizzati con **Docker**
- Vantaggi:
 - Isolamento e indipendenza dei moduli
 - Deployment in qualsiasi ambiente



```
andyalemonta@osteoarthritis-portal-vm:~$ docker compose up --build -d
[+] Running 3/3
✓ Network andyalemonta_app-network Created
✓ Container backend_container Started
✓ Container frontend_container Started
```

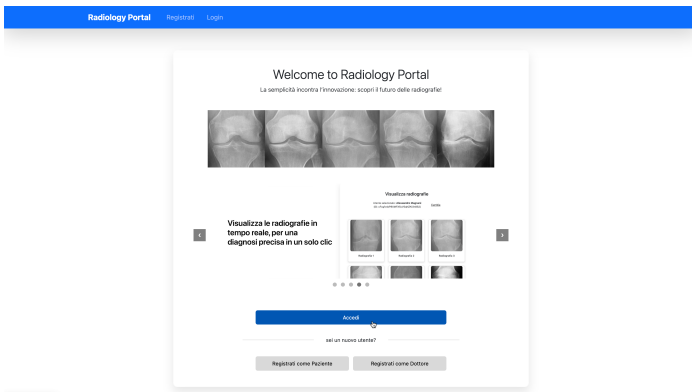
- **Replicazione**
 - Firestore e GCS garantiscono replicazione geografica automatica (gestita da Google Cloud)
- **Gestione errori**
 - In caso di fallimento delle operazioni (es. caricamento radiografie), viene mostrato un messaggio di errore
- **Riavvio servizi**
 - Nessun meccanismo di riavvio automatico previsto per il backend
- Politica **Last Write Wins** per accessi simultanei

- **Disponibilità**

- **Caching** (lazy loading e risorse statiche nel frontend)
- Deploy su **singolo server** senza bilanciamento del carico

- **Sicurezza**

- **Firebase Authentication** per la gestione degli accessi
- Accessi differenziati per medici e pazienti
- Crittografia standard di Google Cloud

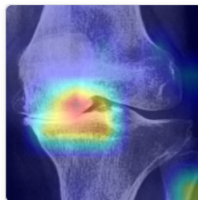


Integrazione del modello di machine learning

- Rete neurale addestrata su un dataset medico contenente radiografie
- **Tensorflow e Keras**
- **Funzionalità:**
 - **Predizione automatica** del grado KL di osteoartrite
 - **Generazione di heat map** per supportare la diagnosi
 - **Addestramento periodico e incrementale** con nuove radiografie



Radiografia Originale



Applicazione Grad-CAM

Classe 5: Avanzata osteoartrite (Ginocchio sinistro)

È necessaria un'operazione protesica per il paziente.

Parere degli esperti

"Con questo applicativo il medico di base può indirizzare il paziente con maggiore sicurezza e rapidità su un trattamento conservativo o chirurgico. In futuro si può integrare con ulteriori dettagli medici, ma è già molto efficace alla versione attuale."

"Inoltre, nei casi dubbi (soprattutto in quelli 3 e 4 della scala), avere una conferma e un supporto da parte di un'intelligenza artificiale fornisce un orientamento più sicuro sui successivi trattamenti."

Dott. Federico Polidoro

(Chirurgo ortopedico presso l'ospedale di Ravenna)

"L'utilizzo nella pratica di tutti i giorni di un applicativo come questo potrebbe ridurre al minimo gli errori di interpretazione delle immagini, consentendo al medico radiologo di produrre un referto più accurato e al chirurgo di porre l'indicazione corretta."

Dott. Andrea Colombelli

(Primario e Chirurgo ortopedico presso l'ospedale di Lugo)

- Aggiunta di ulteriori funzionalità
- Integrazione nella tesi di gruppo
- Restart policy di Docker
- Collaborazioni future

