
PROGETTO CLINICA NOVA

ddddindice

PROGETTO CLINICA NOVA

Presentazione progetto

Nel contesto attuale, molte strutture sanitarie di piccole e medie dimensioni gestiscono ancora le attività cliniche e amministrative in modo parzialmente manuale, con evidenti limiti in termini di efficienza, tracciabilità e accessibilità delle informazioni. Questo progetto nasce dall'esigenza di una clinica diagnostica privata di modernizzare la propria organizzazione interna, offrendo ai pazienti un servizio più rapido e trasparente, e al personale sanitario strumenti digitali per ottimizzare il lavoro quotidiano.

Il progetto consiste nello sviluppo di un sistema informativo per digitalizzare i processi fondamentali legati alla gestione delle attività sanitarie. L'obiettivo è quello di sostituire le procedure manuali con un'applicazione web intuitiva ed accessibile in modo sicuro sia dai pazienti che dal personale della struttura.

L'applicazione offre funzionalità differenziate in base al tipo di utente. I pazienti, dopo aver effettuato la registrazione, possono accedere alla piattaforma per prenotare esami clinici, selezionando la sede e le prestazioni desiderate, possono anche consultare tali prenotazioni effettuate. Una volta effettuata la visita, possono consultare i risultati degli esami direttamente online e scaricare la fattura sanitaria in formato PDF.

Il personale sanitario, composto principalmente da medici e tecnici, ha invece accesso a una sezione dedicata per la consultazione e la gestione delle visite, delle prestazioni assegnate e delle attività cliniche in corso. L'interfaccia dedicata garantisce un'organizzazione efficiente delle risorse e consente la visualizzazione centralizzata delle prenotazioni e dei turni.

Tra le funzionalità principali dell'applicazione figurano la gestione delle prenotazioni multiple, l'associazione dinamica tra esami e personale coinvolto, la generazione automatica delle fatture e l'accesso sicuro tramite credenziali personalizzate. L'intero sistema è pensato per migliorare l'efficienza organizzativa della clinica e per offrire un servizio digitale completo e trasparente agli utenti.

Descrizione generale del dominio

Il dominio applicativo riguarda la gestione informatizzata di una clinica medica in cui i pazienti possono registrarsi e prenotare esami specialistici da svolgere in sedi differenti. Il sistema coinvolge principalmente due categorie di utenti: i pazienti e il personale sanitario (medici e tecnici).

Pazienti

I pazienti sono coloro che usufruiscono dei servizi sanitari. Sono registrati nel sistema con i dati anagrafici, l'email e una password che consente loro di accedere alla webapp. Ciascun paziente può:

- Effettuare più prenotazioni.
- Essere associato a più fatture.
- Avere una sola cartella clinica, creata alla prima visita effettuata e utile a raccogliere lo storico delle visite.

Personale

Il personale è composto da medici e tecnici. Ogni membro del personale è registrato con i dati anagrafici, email e password, e un attributo ruolo che ne indica la funzione (medico o tecnico). Un ulteriore attributo specializzazione rappresenta l'ambito clinico di competenza e permette di associare il personale agli esami corrispondenti per categoria.

L'associazione tra personale e le attività cliniche è rappresentata da una relazione che collega il personale alle prestazioni mediche a cui prende parte. Una prestazione può coinvolgere più membri del personale (es. un tecnico e un medico), e ogni membro del personale può partecipare a più prestazioni.

Prenotazioni ed esami

Il paziente può prenotare uno o più esami. L'associazione tra prenotazioni ed esami è gestita da una relazione intermedia (equivalente a una "lista esami"), in cui vengono specificati tutti gli esami richiesti dal paziente per quella prenotazione.

A seguito della prenotazione viene generata una visita medica iniziale. La visita non è immediatamente associata a informazioni cliniche, ma viene popolata nel tempo con le prestazioni effettivamente svolte. Non è detto che tutti gli esami prenotati vengano poi eseguiti: la prestazione rappresenta l'effettiva erogazione di un esame e include data e referto.

Prestazione

La prestazione rappresenta l'esecuzione concreta di un esame e può coinvolgere più operatori sanitari. Ogni prestazione è associata a un esame, a una visita medica e include la data di esecuzione e il referto clinico.

Una volta che tutte le prestazioni associate a una visita sono state completate e refertate, viene impostata la data della visita (corrispondente alla data dell'ultima prestazione). Solo a quel punto si genera la fattura.

Visita medica

La visita medica è creata a partire dalla prenotazione, e viene poi completata con le prestazioni cliniche. Ogni visita:

- È associata a una prenotazione.
- È svolta in una sede.
- Porta, al termine, alla generazione della fattura.
- È registrata nella cartella clinica del paziente.

Sede

Ogni visita si svolge in una sede. La sede è identificata da un codice ed è descritta da nome, città, indirizzo e CAP.

Fattura

Al completamento della visita, viene emessa una fattura. La fattura:

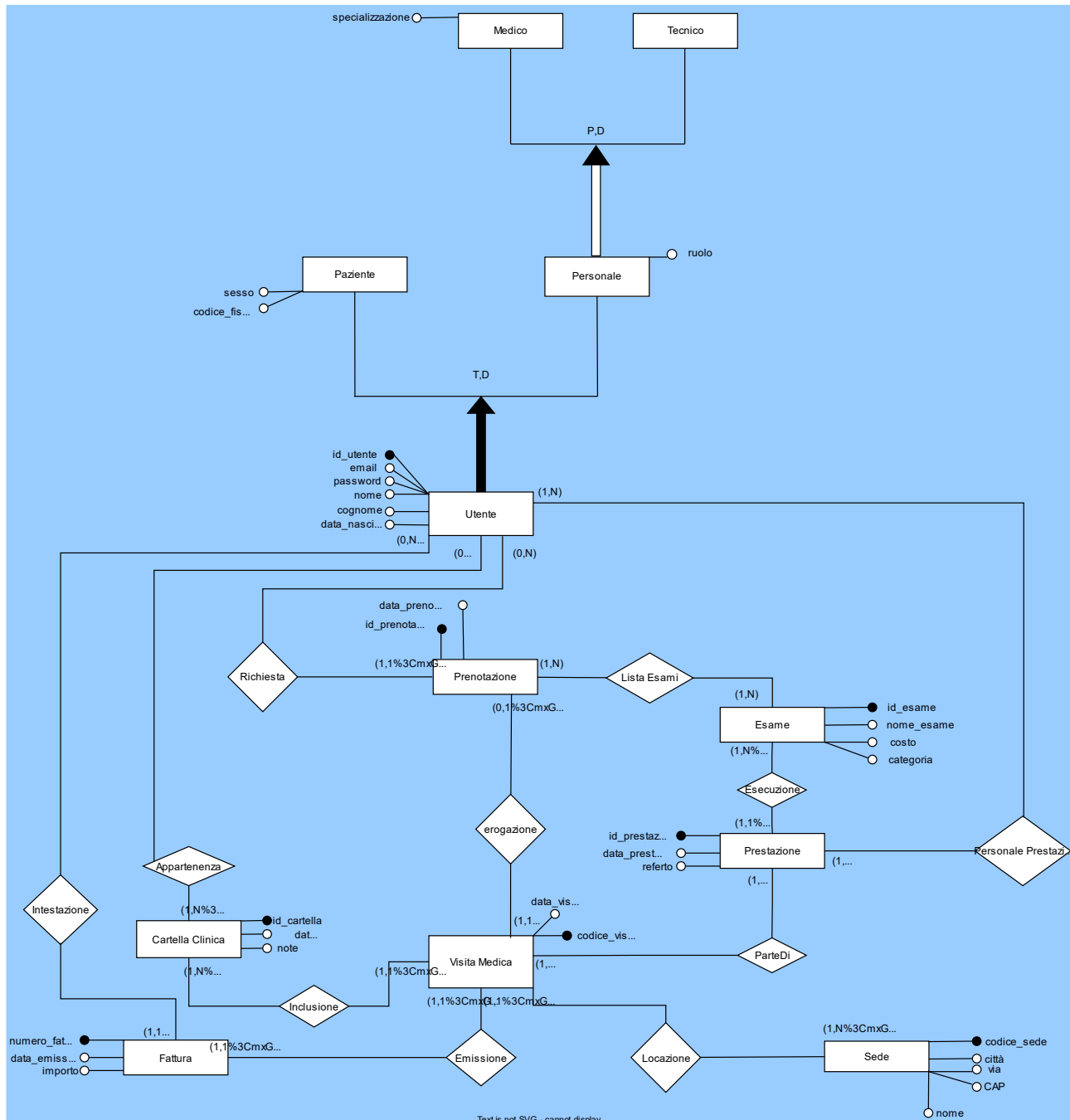
- È univocamente associata alla visita.
- Include la data di emissione (coincidente con la data della visita).
- Contiene l'importo, calcolato come somma dei costi degli esami effettivamente svolti (non quelli solo prenotati).
- È intestata al paziente.

Cartella clinica

Ogni paziente è associato a una cartella clinica, che raccoglie lo storico delle visite. La cartella clinica viene creata solo alla prima visita completata.

1. Progettazione concettuale e logica

Schema ER con generalizzazioni e specializzazioni



Scelte progettuali: specializzazioni e generalizzazioni

Una prima riflessione importante ha riguardato la modellazione dell'entità **Utente**. In una versione iniziale si era ipotizzato di gestire pazienti e personale come sottotipi di un'unica entità Utente. Tuttavia, data la differenziazione a livello di attributi (es. ruolo e specializzazione il personale) ed associazioni (con prenotazioni, cartelle cliniche, fatture per i pazienti; prestazioni per il personale), è stato ritenuto più opportuno promuovere **Paziente** e **Personale** a entità autonome, piuttosto che utilizzare una gerarchia forzata. Questa scelta evita sovrapposizioni e semplifica la gestione delle relazioni.

Tuttavia, all'interno del personale è stata mantenuta una **specializzazione parziale**, risolta con l'attributo ruolo che distingue tra **Medico** e **Tecnico**. Tale specializzazione è **parziale**, poiché in prospettiva il modello può essere esteso con ulteriori tipologie di personale (es. amministrativi, personale delle pulizie, addetti alla segreteria). Questa flessibilità permette una futura estendibilità del sistema senza stravolgimenti strutturali.

La specializzazione tra gli utenti è invece **totale e disgiunta**, in quanto nella clinica possono esistere solamente pazienti o personale, con ruoli ben distinti.

Relazioni significative e associazioni complesse

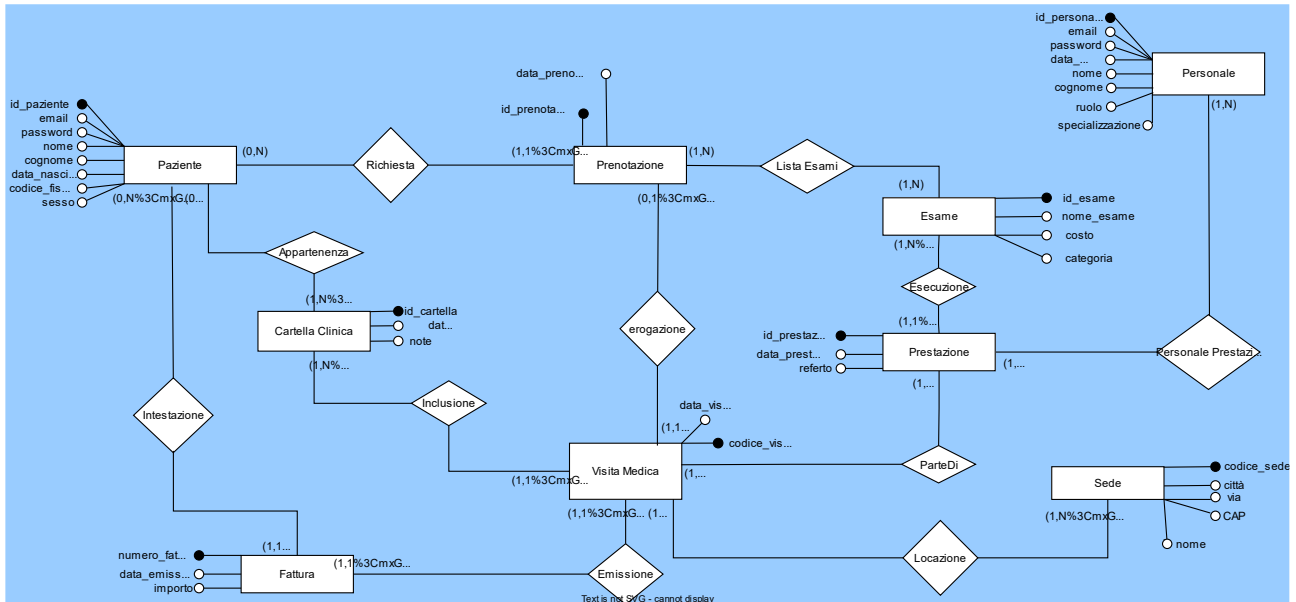
Un altro aspetto centrale del modello è la gestione delle prenotazioni, che avviene tramite l'entità **Prenotazione**, associata a più esami attraverso la relazione molti-a-molti modellata dall'entità associativa **ListaEsami**. Questa tabella intermedia consente di specificare, per ogni prenotazione, i singoli esami richiesti.

Analogamente, anche l'erogazione degli esami prevede il coinvolgimento di più figure professionali (medici e/o tecnici), pertanto è stato necessario introdurre un'ulteriore entità associativa **PersonalePrestazione**, che lega ogni **Prestazione** al **Personale** che lo ha effettuato. Questo approccio riflette la realtà operativa della clinica, dove, ad esempio, un esame radiologico può richiedere sia un medico radiologo sia un tecnico di radiologia.

Da evidenziare è inoltre la relazione uno-a-uno tra **Prenotazione** e **VisitaMedica**, poiché ogni prenotazione porta a una singola visita. A sua volta, la **VisitaMedica** è legata a una **Fattura**, anch'essa in relazione uno-a-uno, in quanto per ogni visita viene emessa un'unica fattura. La **VisitaMedica** è inoltre associata a una **Sede**, riflettendo il luogo in cui l'esame o la prestazione viene effettuata.

La **CartellaClinica**, collegata al paziente, rappresenta una componente essenziale per l'archiviazione dello storico clinico, e si lega anch'essa alle visite mediche per mantenere tracciata tutta l'attività sanitaria.

Schema ER definitivo



2. Progettazione Logica

A partire dallo schema concettuale presentato in precedenza, è stato costruito il corrispondente **schema logico-relazionale**, che riflette le entità, le associazioni e le generalizzazioni identificate. Il modello relazionale è stato normalizzato e strutturato per garantire la coerenza con i vincoli informativi emersi dall'analisi del dominio.

Segue lo schema logico, dove per ogni relazione sono specificate le chiavi primarie (**PK**) e le chiavi esterne (**FK**):

Paziente(id_paziente [PK] , email, password, nome, cognome, data_nascita, codice_fiscale, sesso, id_cartella_clinica : CartellaClinica(id_cartella));

Prenotazione(id_prenotazione [PK], data_prenotazione, id_paziente : Paziente(id_paziente));

Personale (id_personale [PK] , email, password, nome, cognome, data_nascita, ruolo, specializzazione);

Esame(id_esame [PK], nome_esame, costo);

Prestazione(id_prestazione[PK], data_prestazione, referto, id_esame : Esame(id_esame)
,codice_visita : VisitaMedica(codice_visita));

VisitaMedica(codice_visita [PK], data_visita, id_prenotazione : Prenotazione(id_prenotazione),
sede :Sede(codice_sede), id_cartella_clinica : CartellaClinica(id_cartella));

CartellaClinica(id_cartella [PK], data_apertura, note);

Fattura(numero_fattura [PK], data_emissione, importo,codice_visita :
VisitaMedica(codice_visita), id_paziente :Paziente(id_paziente));

Sede (codice_sede [PK], nome, via, città, CAP);

-- Associazione N:N tra Prenotazione e Esame (prenotazione
può includere più esami) e tra Personale e Prestazione

ListaEsami(id_prenotazione[PK] :Prenotazione(id_prenotazione), id_esame[PK] :
Esame(id_esame));

PersonalePrestazione(id_personale [PK]:Personale(id_personale) , id_prestazione[PK]:
Prestazione(id_prestazione)).

Per ogni relazione individuata nel modello logico, si riportano di seguito i **vincoli di dominio** (tipi di dato, dimensione) e i **vincoli di integrità** per ciascun attributo, al fine di guidare la successiva implementazione fisica del database:

Nota: Alcuni vincoli applicativi, come il controllo di valori numerici (es. costo o importo positivo), sono stati gestiti a livello di implementazione nel database

ALTRIMENTI AGGIUSTA

Paziente			
ATTRIBUTO	TIPO	DIMENSIONE	VINCOLI
id_paziente	INTEGER	-	PK, AUTO_INCREMENT
email	VARCHAR	255	NOT NULL, UNIQUE
password	VARCHAR	128	NOT NULL
nome	VARCHAR	100	NOT NULL
cognome	VARCHAR	100	NOT NULL
data_nascita	DATE	-	NOT NULL
codice_fiscale	CHAR	16	NOT NULL, UNIQUE
sex	CHAR	1	CHECK (sex IN ('M','F'))
id_cartella_clinica	INTEGER	-	FK → CartellaClinica(id_cartella), NOT NULL
CartellaClinica			
ATTRIBUTO	TIPO	DIMENSIONE	VINCOLI
id_cartella	INTEGER	-	PK, AUTO_INCREMENT
data_apertura	DATE	-	NOT NULL
note	TEXT	-	NULLABLE
Prenotazione			
ATTRIBUTO	TIPO	DIMENSIONE	VINCOLI
id_prenotazione	INTEGER	-	PK, AUTO_INCREMENT
data_prenotazione	DATE	-	NOT NULL
id_paziente	INTEGER	-	FK → Paziente(id_paziente), NOT NULL
Esame			
ATTRIBUTO	TIPO	DIMENSIONE	VINCOLI
id_esame	INTEGER	-	PK, AUTO_INCREMENT
nome_esame	VARCHAR	150	NOT NULL, UNIQUE
costo	DECIMAL	6,2	CHECK (costo >= 0)
ListaEsami			
ATTRIBUTO	TIPO	DIMENSIONE	VINCOLI
id_prenotazione	INTEGER	-	PK, FK → Prenotazione(id_prenotazione)
id_esame	INTEGER	-	PK, FK → Esame(id_esame)
VisitaMedica			
ATTRIBUTO	TIPO	DIMENSIONE	VINCOLI
codice_visita	INTEGER	-	PK, AUTO_INCREMENT
data_visita	DATE	-	NOT NULL
id_prenotazione	INTEGER	-	FK → Prenotazione(id_prenotazione), UNIQUE, NOT NULL
codice_sede	INTEGER	-	FK → Sede(codice_sede), NOT NULL
id_cartella_clinica	INTEGER	-	FK → CartellaClinica(id_cartella), NOT NULL
Prestazione			
ATTRIBUTO	TIPO	DIMENSIONE	VINCOLI
id_prestazione	INTEGER	-	PK, AUTO_INCREMENT
data_prestazione	DATE	-	NOT NULL
referto	TEXT	-	NULLABLE
id_esame	INTEGER	-	FK → Esame(id_esame), NOT NULL

codice_visita	INTEGER	-	FK → VisitaMedica(codice_visita), NOT NULL
Personale			
ATTRIBUTO	TIPO	DIMENSIONE	VINCOLI
id_personale	INTEGER	-	PK, AUTO_INCREMENT
email	VARCHAR	255	NOT NULL, UNIQUE
password	VARCHAR	128	NOT NULL
nome	VARCHAR	100	NOT NULL
cognome	VARCHAR	100	NOT NULL
data_nascita	DATE	-	NOT NULL
ruolo	VARCHAR	50	CHECK (ruolo IN ('Medico', 'Tecnico'))
specializzazione	VARCHAR	100	NULLABLE
PersonalePrestazione			
ATTRIBUTO	TIPO	DIMENSIONE	VINCOLI
id_personale	INTEGER	-	PK, FK → Personale(id_personale)
id_prestazione	INTEGER	-	PK, FK → Prestazione(id_prestazione)
Fattura			
ATTRIBUTO	TIPO	DIMENSIONE	VINCOLI
numero_fattura	INTEGER	-	PK, AUTO_INCREMENT
data_emissione	DATE	-	NOT NULL
importo	DECIMAL	8,2	CHECK (importo >= 0)
codice_visita	INTEGER	-	FK → VisitaMedica(codice_visita), UNIQUE, NOT NULL
id_paziente	INTEGER	-	FK → Paziente(id_paziente), NOT NULL
Sede			
ATTRIBUTO	TIPO	DIMENSIONE	VINCOLI
codice_sede	INTEGER	-	PK, AUTO_INCREMENT
nome	VARCHAR	100	NOT NULL
via	VARCHAR	150	NOT NULL
città	VARCHAR	100	NOT NULL
CAP	CHAR	5	CHECK (CAP ~ '^\\d{5}\$'), NOT NULL

Vincoli interrelazionali

1. Data delle prestazioni rispetto alla prenotazione

- o La data di una prestazione deve essere **successiva o uguale** alla data della prenotazione associata, in quanto rappresenta l'esecuzione concreta di un esame richiesto.
- o Questo vincolo è applicato a livello applicativo, durante la refertazione, impedendo l'inserimento di date non coerenti.

2. Associazione tra personale ed esami in base alla specializzazione

- o Il personale può essere assegnato a una prestazione solo se la sua **specializzazione** coincide con la **categoria dell'esame** da svolgere.

- Il controllo è applicato in fase di assegnazione, e garantisce la coerenza tra ambito clinico dell'operatore e tipologia dell'esame.

3. Prestazioni diverse dagli esami prenotati

- Non è garantito che ogni esame prenotato venga effettivamente svolto. L'effettiva esecuzione è rappresentata dalle prestazioni associate a una visita. Pertanto, il sistema consente una possibile **discrepanza tra esami prenotati e prestazioni effettuate**, da verificare eventualmente a posteriori.

4. Unicità della cartella clinica

- Ogni paziente può essere associato a **una sola cartella clinica**.
- La relazione tra paziente e cartella è realizzata tramite una chiave esterna in Paziente, impostata nullable per gestire i pazienti che non hanno ancora svolto visite.
- L'unicità è garantita dal fatto che ogni cartella ha un identificativo primario e che l'associazione è uno-a-uno.

5. Univocità tra visita e fattura

- Ogni visita genera **una sola fattura**, e ogni fattura è associata a una sola visita.
- Questo vincolo è rappresentato logicamente con una relazione uno-a-uno.

6. Completamento della visita

- Una visita può essere considerata completata solo quando tutte le prestazioni associate sono state refertate.
- In quel momento viene calcolata la **data della visita**, assegnata come la data dell'ultima prestazione eseguita.

3.Implementazione del Sistema informativo e Funzioni della WebApp

L'applicazione è stata sviluppata utilizzando il framework **Django**, che gestisce sia la logica di backend che la generazione dinamica delle pagine HTML attraverso il suo sistema integrato di **template**. L'interfaccia utente è stata costruita con **Bootstrap CSS**, così da garantire un layout responsivo e ordinato, senza l'utilizzo esteso di JavaScript, se non nei casi strettamente necessari.

La struttura dell'app segue l'architettura **MVC** (Model-View-Controller), secondo l'approccio proprio di Django, dove:

- i **Model** definiscono la struttura dei dati e interagiscono con il database tramite l'ORM (Object Relational Mapper) integrato;
- le **View** gestiscono la logica applicativa e le operazioni sui dati;
- i **Template** (equivalenti alla "View" nel pattern MVC classico) si occupano della presentazione HTML.

L'applicazione, realizzata in Django, si propone di gestire in modo digitale il flusso clinico di una struttura sanitaria privata.

Tra le funzionalità teoricamente realizzabili, rientrano:

- Registrazione e autenticazione degli utenti, con ruoli differenziati (pazienti e personale sanitario).
- Inserimento e gestione delle prenotazioni.
- Attribuzione automatica delle prestazioni al personale competente.
- Refertazione delle prestazioni da parte del medico.
- Visualizzazione dei referti da parte del paziente.
- Emissione e download delle fatture in PDF.
- Visualizzazione delle visite da completare con medici assegnati.
- Dashboard personalizzate in base al tipo di utente.
- Visualizzazione dello storico delle prenotazioni.
- Accesso alla piattaforma solo per personale accreditato.
- Possibilità di registrazione autonoma per i pazienti

Funzionalità effettivamente realizzate

Nell'ambito di questo progetto, sono state implementate con successo le seguenti funzionalità:

- **Autenticazione differenziata** tra pazienti e personale medico/tecnico.
- **Registrazione pazienti** tramite form personalizzato.
- **Dashboard dinamiche** per pazienti e medici.
- **Creazione prenotazioni**, con selezione di data, sede e esami.

- **Attribuzione automatica delle prestazioni** ai medici (e tecnici se necessario), in base a specializzazione e carico di lavoro.
- **Gestione referti** da parte del medico, con salvataggio e controllo delle prestazioni non ancora refertate.
- **Consultazione dei referti** lato paziente.
- **Emissione e download delle fatture** con riepilogo degli esami eseguiti.
- **Visualizzazione delle visite da completare**, utile per l'organizzazione interna della struttura.
- Utilizzo della sessione per mantenere **lo stato utente** e il relativo contesto di navigazione.

Home page e accesso differenziato

Benvenuto in Clinica Nova

Gestisci esami, visite e cartelle cliniche in modo semplice e digitale.

Area Pazienti

[Accesso Paziente](#)

[Registrazione Paziente](#)

[Accesso Medico](#)



Cosa puoi fare con Clinica Nova

 [Vedi Visite da Completare](#)

La **homepage** rappresenta il punto di accesso principale al sistema. Essa offre un'interfaccia semplificata per gli utenti, con una chiara distinzione tra l'area **pazienti** e quella **personale sanitario**.

In particolare:

- I **pazienti** possono registrarsi autonomamente attraverso l'apposito form, poiché rappresentano gli utenti finali della clinica.
- Il **personale medico o tecnico**, invece, **non può registrarsi autonomamente**: l'accesso è possibile solo se l'account è stato precedentemente creato e abilitato da un amministratore. Questo meccanismo riflette la logica organizzativa della clinica, in cui è il sistema (o la segreteria) a gestire e accreditare i lavoratori interni.

In fase di login, il sistema distingue automaticamente tra pazienti e personale in base all'indirizzo email inserito. Il controllo avviene nel seguente ordine:

1. Se l'email è presente nella tabella dei **pazienti**, viene verificata la relativa password. Se valida, l'utente accede alla **dashboard paziente**, dove può prenotare esami e visualizzare lo storico delle visite.
2. Se l'email è associata a un membro del **personale sanitario**, viene eseguita una logica analoga per la validazione. Solo in caso di ruolo **"Medico"** viene data la possibilità di accedere alla gestione dei **referti** e delle **prestazioni sanitarie**.

Le informazioni relative all'identità dell'utente vengono salvate in sessione (request.session), consentendo una navigazione persistente e personalizzata all'interno del sistema. In questo modo, l'applicazione può mostrare contenuti riservati in base al ruolo utente, oppure impedire l'accesso ad aree non autorizzate.

Il sistema prevede due tipologie di utenti: **pazienti** e **personale medico/tecnico**. Sebbene venga utilizzato un unico form di login, il sistema è in grado di riconoscere a quale categoria appartiene l'utente in base all'email inserita.

Nel dettaglio, al momento dell'autenticazione viene eseguito un controllo sull'indirizzo email. Se l'email è presente nella tabella dei pazienti, verrà verificata la password corrispondente e, in caso positivo, l'utente sarà reindirizzato alla propria dashboard personale. In alternativa, se l'email appartiene a un membro del personale, il sistema esegue la stessa logica di validazione e accesso, ma verso l'interfaccia riservata al personale.

Il codice sottostante riassume il funzionamento della view:

```
def login_auth(request): 1 usage  Rosario *
    if request.method == 'POST':
        email = request.POST['email']
        password = request.POST['password']

        # Verifica se esiste un paziente con quell'email
        if Paziente.objects.filter(email=email).exists():
            user = Paziente.objects.get(email=email)
            if check_password(password, user.password):
                print(user.password)
                request.session['user_type'] = 'paziente'
                request.session['user_id'] = user.id_paziente
                return render(request, template_name='dashboard_paziente.html', context={'nome': user.nome})

        # Verifica se esiste un membro del personale con quell'email
        if Personale.objects.filter(email=email).exists():
            user = Personale.objects.get(email=email)
            if check_password(password, user.password):
                request.session['user_type'] = 'personale'
                request.session['user_id'] = user.id_personale
                return render(request, template_name='dashboard_personale.html', context={'nome': user.nome})

    return render(request, template_name='login.html', context={'error': 'Credenziali non valide'})
```

L'integrazione con l'**ORM di Django** non solo semplifica l'accesso ai dati, ma garantisce anche una maggiore **sicurezza e manutenibilità del codice**, riducendo il rischio di vulnerabilità come SQL injection e migliorando l'astrazione del database.

(Nota sulla prenotazione)

Una volta autenticato, il **paziente** ha accesso a una sezione dedicata alla **gestione delle prenotazioni**, dove può visualizzare lo storico e lo stato delle proprie visite. In particolare, è possibile sapere:

- quali esami sono stati prenotati,
- quali **medici o tecnici** sono stati assegnati alle singole prestazioni,
- e lo stato di completamento/refertazione delle stesse.

Queste funzionalità costituiscono la base della digitalizzazione del percorso clinico del paziente.

Autenticazione differenziata degli utenti

L'informazione relativa all'identità dell'utente viene conservata nella sessione (`request.session`) per l'intera durata della navigazione. Questo consente al sistema di riconoscere l'utente attivo

e di mostrare contenuti personalizzati, o di limitarne l'accesso a funzionalità riservate, in base al ruolo (paziente o personale).

L'uso dell'ORM di Django in questo contesto garantisce anche una maggiore sicurezza e manutenibilità del codice, poiché evita la costruzione manuale di query SQL e protegge nativamente da attacchi come SQL injection.

Registrazione paziente

Attraverso la sezione "Registrazione Paziente", un utente può creare in autonomia un nuovo account personale. Il sistema presenta un form dove vengono richiesti i dati anagrafici essenziali: nome, cognome, data di nascita, codice fiscale, sesso, indirizzo email e password.

Una volta inviato il form, il sistema valida automaticamente i dati forniti (tramite `form.cleaned_data`), salvandoli nel database se conformi. La password non viene mai memorizzata in chiaro: prima di essere salvata, viene **hashata** tramite l'algoritmo **PBKDF2** (Password-Based Key Derivation Function 2), un sistema di derivazione sicuro e standardizzato, che protegge le credenziali contro gli attacchi a forza bruta.

Nuovo Paziente

Nome:

Cognome:

Data di Nascita:



Codice Fiscale:

Sesso:

☐ Maschio

☐ Femmina

Email:

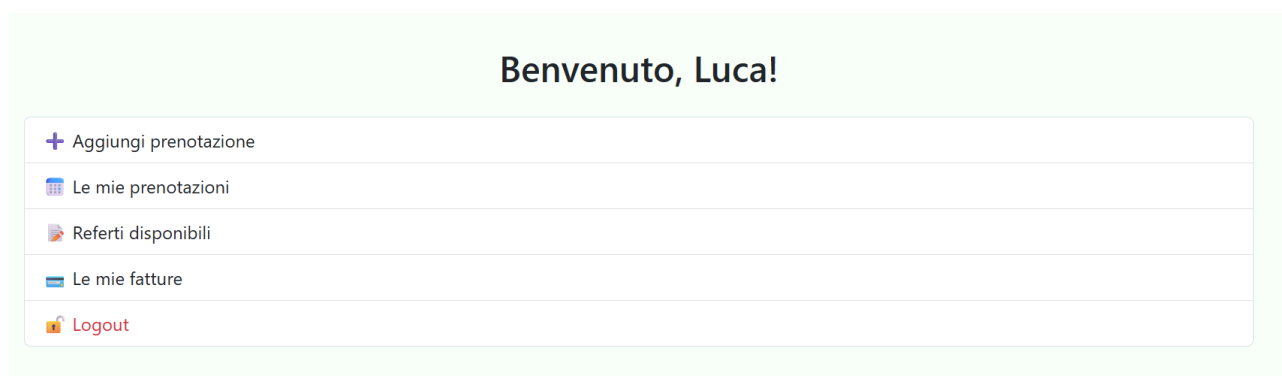
Password:

Registrati

Al termine della registrazione, il sistema conferma l'avvenuto salvataggio con un messaggio di successo visibile a schermo.

Dashboard paziente e funzione 'Aggiungi prenotazione'

Una volta effettuato il login, il paziente viene reindirizzato alla propria **dashboard personale**, dove può accedere alle principali funzioni previste dal sistema. Nella parte superiore della pagina viene mostrato il **nome del paziente loggato**, così da confermare l'identità dell'utente attivo.



Le funzionalità accessibili dalla dashboard includono:

1. **Aggiungi Prenotazione**

Permette al paziente di selezionare:

- o la **data** in cui intende effettuare gli esami,
- o la **sede** tra quelle disponibili,
- o uno o più **esami** da svolgere tra quelli presenti nel catalogo.

2. **Le mie prenotazioni**

Visualizza l'elenco delle prenotazioni effettuate, con dettagli su esami, sede, data e personale sanitario assegnato.

3. **Referti disponibili**

Permette di consultare i **referti** inseriti dal medico per ciascuna prestazione completata. Solo dopo che un medico ha refertato un esame, questo diventa visibile al paziente.

4. **Le mie fatture**


Una volta che tutte le prestazioni associate a una prenotazione risultano completate, il paziente può accedere alla **fattura digitale** dettagliata, con elenco degli esami e importo totale. La fattura è anche scaricabile in formato **PDF**.


5. **Logout**

Termina la sessione utente e **scollega il paziente dal sistema**, garantendo la sicurezza dei dati personali.

Vediamo la prima, l'utente avrà questa schermata

Nuova Prenotazione

Data prenotazione: 20/04/2026 

Sede: Centro Diagnostico Roma Est 

Seleziona

- ☐ Ecografia (ultrasuoni)
- ☐ TAC
- ☐ Poliambulatorio Salute Napoli
- ☐ Risonanza magnetica
- ☒ Radiografia (RX)
- ☐ Elettrocardiogramma (ECG)
- ☐ Ecocardiogramma
- ☐ Holter cardiaco (24-48h)
- ☒ Spirometria
- ☐ Test da sforzo (prova da sforzo)
- ☒ Audiometria

Prenota

Una volta confermata, il sistema:

- crea una nuova **prenotazione** associata al paziente;
- genera una **visita medica** collegata, inizialmente **senza data** (la data verrà assegnata in base alla data dell'ultima prestazione);
- per ogni esame scelto:
 - viene inserito nella **tabella ListaEsami**, creando una coppia prenotazione-esame;
 - viene creata una **prestazione** associata alla visita medica in questione;
 - viene assegnato un **medico** in base alla specializzazione richiesta dall'esame, scegliendo tra i meno occupati;
 - se l'esame appartiene alla categoria **Radiologia**, viene associato anche un **tecnico specializzato**.

Nota Bene: La data della visita e delle relative prestazioni è sempre **successiva** alla data di prenotazione. Questa scelta nasce da una **semplificazione didattica**, adottata per mantenere la coerenza temporale tra le fasi del processo clinico senza dover gestire la complessità di una reale agenda sanitaria. Lo scopo è mostrare la corretta **struttura logica** dell'applicazione e delle relazioni nel database, piuttosto che replicare fedelmente tutte le casistiche reali.

Le mie prenotazioni

Le mie Prenotazioni

Dopo aver effettuato una prenotazione, il paziente può consultare lo storico nella sezione "**Le mie prenotazioni**", accessibile dalla dashboard.

All'interno di questa vista viene mostrato:

- l'elenco delle prenotazioni associate al paziente loggato,
- la lista degli esami richiesti per ciascuna prenotazione,
- la sede scelta e la data richiesta.

Le Tue Prenotazioni

ID Prenotazione: 29
Data: 20/04/2026

Sede: Centro Diagnostico Roma Est

Esami prenotati:

Radiografia (RX)	50.00 €
Spirometria	70.00 €
Audiometria	45.00 €

Torna alla Dashboard

Visite da completare

Tornando nella *home del sito* e accedendo alla sezione "**Visite da completare**", è possibile visualizzare in ordine tutte le prenotazioni effettuate.

Questa funzione è pensata per il **personale clinico** o per la segreteria, e mostra:

- l'ID della prenotazione,

- il nome del paziente,
- tutti gli esami associati,
- e soprattutto il **personale sanitario incaricato** per ciascuna prestazione, inclusi i medici e, se necessario, i tecnici.

Questo consente di avere una **visione d'insieme operativa** delle visite future e delle risorse coinvolte.

Prenotazione ID 29 - April 20, 2026

Paziente: Luca Bianchi

Esame: Radiografia (RX)

Personale assegnato:

- Elena Ferrari (Medico)
- Laura Marini (Tecnico)

Esame: Spirometria

Personale assegnato:

- Alberto Verdi (Medico)

Esame: Audiometria

Personale assegnato:

- Marco Conti (Medico)


[← Torna alla Home](#)


Dashboard medico e Gestione Prestazioni

Una volta effettuato l'accesso come medico, viene mostrata una **dashboard personale** con il nome del dottore e le funzioni disponibili:

- **Gestione prestazioni**
- **Logout**

Benvenuto, Dott. Elena!

 Gestione prestazioni

 Logout

Se il medico seleziona “**Gestione prestazioni**”, avrà accesso alla lista delle **prestazioni a lui assegnate** che **non sono ancora state refertrate**.

Il sistema filtra in automatico le prestazioni in base al medico loggato, mostrando solo quelle **senza referto associato**, grazie al seguente criterio logico:

```
# Recupera solo prestazioni assegnate a questo medico, senza referto
prestazioni = Prestazione.objects.filter(
    personale__id_personale=medico,
    referto__isnull=True
).select_related(
    *fields: 'id_esame',
    'codice_visita__id_prenotazione__id_paziente'
).distinct()
```

Vengono mostrati:

- nome del paziente,
- esame da refertare,
- data della prestazione,
- campi da compilare (esito, descrizione, data referto).

Una volta compilati **tutti i campi obbligatori**, il medico può cliccare su **“Salva referto”**. La pagina viene ricaricata e la prestazione **scompare dalla lista**, poiché ora è considerata completata.

Esame: Radiografia (RX)

Paziente: Luca Bianchi

Codice Prenotazione: 29

Data referto:

21/04/2026

Testo referto:

I campi polmonari appaiono ben espansi e normotrasparenti. Non si evidenziano addensamenti parenchimali in atto. Il profilo cardiaco è nei limiti. Seni costofrenici liberi. Non si osservano lesioni ossee di significato patologico.

Salva Referto

Torna alla Dashboard

Visualizzazione referti disponibili

Dopo che tutte le prestazioni sono state refertate, il paziente può consultare i risultati nella sezione **“Referti disponibili”**.

Questa vista mostra un **riepilogo completo** di:

- data dell'esame,
- nome dell'esame,
- medico che ha eseguito il referto,
- esito e descrizione forniti.

I dati sono ottenuti in base all'**ID del paziente salvato nella sessione**.

Il sistema esegue una query che **recupera tutte le prestazioni** appartenenti al paziente, **con referto e data compilati**, e li passa alla pagina HTML tramite **context**, permettendo la visualizzazione dinamica nell'interfaccia.

I tuoi referti

Radiografia (RX)

Data prestazione: April 21, 2026

Referto:

I campi polmonari appaiono ben espansi e normotrasparenti. Non si evidenziano addensamenti parenchimali in atto. Il profilo cardiaco è nei limiti. Seni costofrenici liberi. Non si osservano lesioni ossee di significato patologico. Conclusioni: Radiografia del torace nei limiti della norma.

Codice prenotazione: 29

Spirometria

Data prestazione: April 22, 2026

Referto:

Esame eseguito correttamente. FVC: 3.8 L (95% del valore teorico) FEV1: 3.0 L (88% del valore teorico) FEV1/FVC: 79% Conclusioni: Funzione ventilatoria nei limiti della norma. Non si evidenziano segni di ostruzione bronchiale.

Codice prenotazione: 29

Audiometria

Data prestazione: April 21, 2026

Referto:

Audiogramma tonale eseguito in cabina silente. Orecchio destro: ipoacusia neurosensoriale lieve sulle frequenze acute. Orecchio sinistro: soglia uditiva nei limiti della norma. Buona discriminazione vocale bilaterale. Conclusioni: Ipoacusia neurosensoriale lieve a destra. Si consiglia controllo periodico.

Codice prenotazione: 29

[Torna alla Dashboard](#)

```
def visualizza_referti(request):  
    if request.session.get("user_type") != "paziente":  
        return redirect('login')  
  
    paziente_id = request.session.get("user_id")  
  
    referti = Prestazione.objects.filter(  
        codice_visita__id_prenotazione__id_paziente=paziente_id,  
        referto__isnull=False,  
        data_prestazione__isnull=False  
    ).select_related(*fields: 'id_esame', 'codice_visita__id_prenotazione')  
  
    return render(request, template_name: 'visualizza_referti.html', context: {'referti': referti})
```

Visualizzazione e download fattura

Una volta che tutte le prestazioni di una prenotazione sono state completate (cioè **refertrate**), il paziente può accedere alla sezione “**Le mie fatture**”.

Qui è possibile:

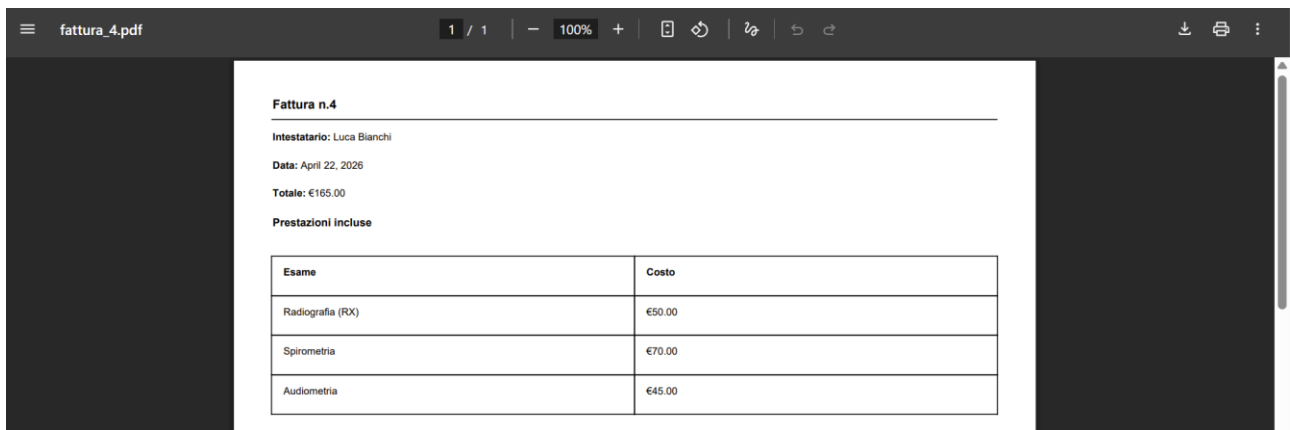
- visualizzare il riepilogo della **fattura digitale**, con dettagli su ciascun esame svolto,
- vedere il **totale complessivo**,
- e **scaricare il PDF** della fattura, completo di intestazione, elenco prestazioni, importi e riferimenti anagrafici.

Fatture disponibili

Data Fattura Intestatario Totale (€) Azioni

April 22, 2026 Luca Bianchi 165.00 [Scarica PDF](#)

[Torna alla Dashboard](#)



Fattura n.4

Intestatario: Luca Bianchi

Data: April 22, 2026

Totale: €165.00

Prestazioni incluse

Esame	Costo
Radiografia (RX)	€50.00
Spirometria	€70.00
Audiometria	€45.00

Conclusioni

L'intero progetto ha finalità **didattiche** e non intende rappresentare un'applicazione clinica completa o perfettamente aderente alla realtà sanitaria.

Al contrario, alcune **semplificazioni logiche e strutturali** sono state adottate per rendere più chiara la progettazione e per concentrarsi sulla **struttura del database**, la **gestione del flusso dati** e l'integrazione tra frontend e backend.

Aspetti migliorabili

Una delle principali semplificazioni riguarda il **meccanismo di prenotazione** e attribuzione delle prestazioni:

- Attualmente la prenotazione si limita a registrare gli esami scelti, senza prevedere date e orari specifici per ciascuna prestazione.
- Le prestazioni vengono poi assegnate automaticamente al medico (o tecnico) **con meno carico di lavoro**, in base alla specializzazione.

In un contesto reale, si potrebbe:

- Integrare un sistema avanzato di **prenotazione su slot orari** disponibili, selezionabili direttamente dal paziente.
- Introdurre un **ruolo amministrativo** che, in autonomia, **assegna manualmente** le prestazioni al personale.

Integrazioni future possibili

Per arricchire l'esperienza d'uso e rendere la piattaforma più completa, si potrebbero implementare:

- **Storico clinico del paziente**, con riepilogo di visite, referti e patologie.
- **Gestione prenotazioni** con possibilità di **modifica o cancellazione** da parte del paziente.
- **Messaggistica interna** tra pazienti e personale medico.
- **Funzionalità di filtro e ricerca** tra le prenotazioni effettuate.
- **Allegati multimediali nei referti**, come immagini diagnostiche.
- **Gestione del profilo utente**: modifica dati anagrafici, immagine profilo, contatti, ecc.
- **Ruoli aggiuntivi**, come personale amministrativo o altri operatori tecnici.

4. Sicurezza App

L'applicazione web è stata eseguita su una macchina Windows 11, con server Django in ascolto sulla porta 8000. Per renderla raggiungibile da altri dispositivi nella rete locale, è stato avviato il server con il comando:

```
python manage.py runserver 0.0.0.0:8000
```

In questo modo, l'app è risultata accessibile dall'indirizzo IP della macchina, ovvero 192.168.1.100.

Per l'attività di analisi delle vulnerabilità, è stata utilizzata una seconda macchina con Kali Linux installata su VirtualBox, configurata in modalità bridge, con indirizzo IP 192.168.1.101. Ciò ha permesso di raggiungere la web app da Kali attraverso l'indirizzo `http://192.168.1.100:8000`.

Nel sistema è presente una view dedicata al login, che gestisce l'autenticazione degli utenti. In questa fase, la gestione dell'accesso è implementata tramite una query SQL scritta manualmente, che recupera i dati dell'utente confrontando email e password fornite con quelle presenti nel database:

```
query = f"""
SELECT id_paziente, nome FROM CentroMedico_paziente
WHERE email = '{email}' AND password = '{password}'
"""
```

Analizzando questa porzione di codice, si osserva che la query viene costruita concatenando direttamente i dati provenienti dall'utente. Questo tipo di costruzione apre la possibilità a manipolazioni attraverso tecniche di SQL injection, poiché non è presente alcun meccanismo di separazione tra il codice SQL e i valori di input. È quindi naturale, in una fase di verifica della sicurezza, considerare la pagina di login come un potenziale punto di ingresso per un attacco SQLi.

Dal momento che la pagina riceve i dati di login via POST e interagisce direttamente con il database attraverso SQL dinamico, è stato deciso di utilizzare `sqlmap`, uno strumento automatico in grado di rilevare e sfruttare vulnerabilità di tipo SQL injection. Attraverso un'analisi del traffico o semplicemente osservando la struttura del form di login, si è potuto ricostruire la richiesta POST inviata all'endpoint e preparare il test con `sqlmap`.