

UNIVERSITÀ DEGLI STUDI DI NAPOLI
PARTHENOPE
DIPARTIMENTO DI INGEGNERIA



**CORSO DI LAUREA IN INGEGNERIA E SCIENZE
INFORMATICHE PER LA CYBERSECURITY**

TITOLO PROGETTO

UniLabManager

DOCENTE

PROFESSORE DANIELE GRANATA

STUDENTE

Andrea Di Palo

ANNO ACCADEMICO 2024/2025

Descrizione del progetto:

Il progetto *UniLabManager* è un sistema web progettato per gestire in maniera centralizzata le attività didattiche e sperimentali svolte all'interno dei laboratori universitari.

L'obiettivo principale è offrire un'interfaccia moderna ed efficiente per la **creazione e supervisione di progetti scientifici**, la **gestione di esperimenti**, e la **prenotazione di attrezzature e laboratori**, con ruoli differenziati per **professori, studenti e tecnici**.

Il sistema è stato realizzato per simulare un contesto reale accademico, in cui diversi attori interagiscono in un ambiente strutturato. La progettazione tiene conto sia della flessibilità di utilizzo sia delle regole di integrità dei dati, con l'ausilio del framework Django.

Obiettivi:

- Permettere la **registrazione differenziata** degli utenti secondo il proprio ruolo (Professore, Studente, Tecnico)
- Consentire al Professore di creare progetti sperimentali e associarvi esperimenti
- Fornire la possibilità di **prenotare attrezzature e laboratori** per ogni esperimento
- Offrire una **dashboard personalizzata** per ciascun utente
- Garantire **sicurezza dei dati**, autenticazione tramite matricola e accessi controllati

Requisiti Funzionali

RF1 – Registrazione dell'utente

Il sistema deve permettere la registrazione di nuovi utenti specificando il ruolo (Professore, Studente, Tecnico), tramite un form personalizzato per ciascun tipo. Durante la registrazione, viene verificata e salvata una matricola unica con formato specifico (P12345 per professori, S12345 per studenti, T12345 per tecnici).

RF2 – Autenticazione

Il sistema deve consentire l'autenticazione degli utenti registrati tramite matricola e password.

RF3 – Accesso a dashboard personalizzata

Dopo l'autenticazione, ogni utente accede ad una dashboard diversa a seconda del ruolo:

- I professori visualizzano i propri progetti sperimentali e possono crearne di nuovi.
- Gli studenti accedono a contenuti didattici o prenotazioni (in futuro).
- I tecnici visualizzano le prenotazioni da gestire.

RF4 – Creazione progetto sperimentale (Professore)

Un professore autenticato può creare nuovi progetti sperimentali inserendo titolo, descrizione, obiettivi, date, ecc.

RF5 – Visualizzazione ed eliminazione dei progetti

I professori possono visualizzare l'elenco dei propri progetti e scegliere di eliminarne uno se necessario.

RF6 – Creazione esperimenti associati a un progetto

Per ogni progetto, il professore può creare uno o più esperimenti, specificando descrizione, obiettivi, materiali e date.

RF7 – Prenotazione laboratorio e attrezzature

Durante la creazione di un esperimento, il sistema consente al professore di prenotare un laboratorio e selezionare una o più attrezzature, evitando conflitti di orario o doppie prenotazioni.

RF8 – Verifica disponibilità risorse

Il sistema deve impedire la prenotazione di laboratori o attrezzature già occupati nella stessa fascia oraria.

RF9 – Partecipazione agli esperimenti (Studente)

Il sistema deve consentire agli **studenti autenticati** di visualizzare l'elenco dei **progetti disponibili** e di **richiedere la partecipazione** agli esperimenti collegati. L'accesso a queste funzionalità è riservato solo agli utenti con ruolo "studente" che hanno effettuato il login.

RF10 – Gestione limite partecipanti per progetto sperimentale

Ogni progetto sperimentale deve avere un campo che indica il **numero massimo di studenti ammessi**. Il sistema deve impedire ulteriori iscrizioni quando il limite è stato raggiunto

RF11 – Logout

Il sistema deve permettere in ogni momento il logout dell'utente.

RF12 - Gestione Attrezzatura

Il tecnico deve poter aggiungere e/o modificare lo stato delle attrezzature

Modello Concettuale (E-R)

Il servizio web è pensato per la gestione dei progetti e delle attività sperimentali nei laboratori universitari.

1. I **professori** possono creare progetti, definire esperimenti e prenotare laboratori e attrezzature.
2. Gli **studenti**, una volta autenticati, possono visualizzare i progetti disponibili, prenotarsi e monitorare le attività a cui partecipano.
3. I **tecnici** hanno accesso a un pannello dedicato per gestire la disponibilità delle attrezzature e supervisionare le prenotazioni.

Ogni utente accede con credenziali univoche e visualizza solo le funzionalità legate al proprio ruolo. Il sistema garantisce un'organizzazione efficiente, evitando sovrapposizioni e semplificando la comunicazione tra studenti, docenti e tecnici.

- Attributi comuni: matricola, nome, cognome, email
- Attributi privati: password

Gestione della generalizzazione:

Accorpamento in padre

Nel progetto **UniLabManager**, per la gestione delle diverse tipologie di utenti (Professore, Studente, Tecnico) si è deciso di adottare una strategia di **generalizzazione con accorpamento nel padre**, che come rappresentato nel modello E-R, consiste nel mappare l'entità genitore **Utente** e tutte le sue specializzazioni in un'unica tabella fisica. La distinzione dei ruoli è affidata all'attributo omonimo **Ruolo**.

Motivazioni della scelta:

1. Semplificazione architetturale

L'utilizzo di un'unica tabella **Utente** rende lo schema del database semplice, lineare e facile da comprendere. Si evita la complessità di dover gestire e mantenere più tabelle correlate per rappresentare una singola gerarchia, riducendo i possibili errori e facilitando la manutenzione.

2. Efficienza

Poiché tutti i dati degli utenti sono in un'unica tabella, non è necessario eseguire operazioni di JOIN per recuperare le informazioni complete di un **Utente**. Ad esempio, per ottenere il nome di uno studente (attributo comune) e il suo anno di corso (attributo specifico), è sufficiente una singola e veloce.

Svantaggi:

1. Presenza di Valori **NULL**:

La tabella **Utente** contiene colonne specifiche per certi ruoli che rimarranno NULL per altri. Ad esempio, la colonna `area_competenza` sarà NULL per gli studenti.

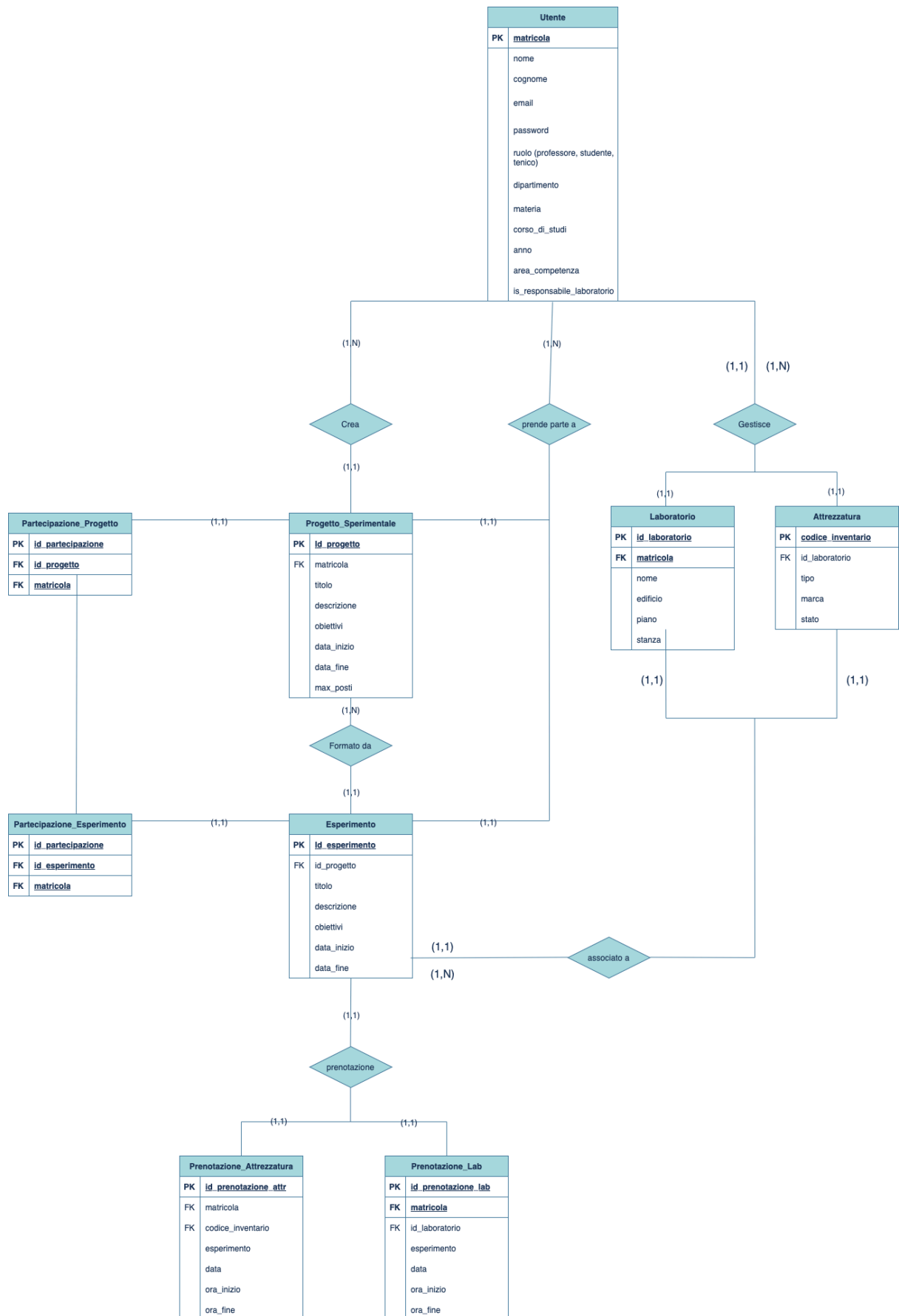
2. Mitigazione:

Nel contesto del progetto, il numero di attributi specifici per ciascun ruolo è limitato. Lo spreco di spazio è trascurabile. L'integrità dei dati (ad esempio, assicurarsi che uno studente abbia sempre un anno definito) viene gestita a livello di logica applicativa o tramite vincoli di controllo a livello di codice.

Aspetti implementativi e vincoli:

- Un **Utente** può essere solo o professore, o studente, o tecnico . Quindi non può essere contemporaneamente professore e studente. Questo vincolo è garantito dal fatto che l'attributo **Ruolo** può contenere un solo valore.
- **Generalizzazione totale:**
Ogni **Utente** nel sistema deve obbligatoriamente appartenere a un ruolo. Questo viene imposto definendo l'attributo **ruolo** come obbligatorio (NOT NULL) al momento della creazione di un nuovo utente.
- Le relazioni come Crea (tra **Utente** e **Progetto_Sperimentale**), partono dall'unica tabella Utente. Sarà compito della logica applicativa verificare che il ruolo dell'utente che avvia la relazione sia appropriato (es. solo un 'professore' può creare un progetto)

Modello E-R



Modello logico

1) UTENTE

Nome Campo	Descrizione	Tipo	Lunghezza	Vincoli
Matricola	Identificativo dell'utente	Char	6	Primary Key, Not null, Unique
Nome	Nome dell'utente	Varchar	100	Not null
Cognome	Cognome dell'utente	Varchar	100	Not null
Email	Email dell'utente	Varchar	150	Not null, Unique
Password	Password dell'utente	Varchar	128	Not null
Ruolo	Tipologia di utente (Professore, Studente, Tecnico)	Enum	-	Not null, valori ammessi ['Professore', 'Studente', 'Tecnico']
Dipartimento	Dipartimento di appartenenza	Varchar	100	Not null se ruolo = 'Professore'
Materia	Materia insegnata	Varchar	100	Not null se ruolo = 'Professore'
Corso di Studi	Corso di Studi di appartenenza	Varchar	100	Not null se ruolo = 'Studente'
Anno	Anni di corso	Int $1 < x < 5$	-	Not null se ruolo = 'Studente'
Area di competenza	Competente in	Varchar	100	Not null se ruolo = 'Tecnico'
Is_responsabile	Responsabilità laboratorio	Bool	-	Not null se ruolo = 'Tecnico'

Ruolo

Un utente deve essere o un Professore o uno Studente o un Tecnico:

- Se l'utente è un professore si compileranno i campi matricola, nome, cognome, email, password, ruolo, **dipartimento** e **materia**; i campi restanti saranno di tipo Null;
- Se l'utente è uno studente si compileranno i campi matricola, nome, cognome, email, password, ruolo, **corso di studi** e **anno**; i campi restanti saranno di tipo Null;
- Se l'utente è un tecnico si compileranno i campi matricola, nome, cognome, email, password, ruolo, **area di competenza** e **is_responsabile**; i campi restanti saranno di tipo Null;

Vincoli di tupla

Un utente non può registrarsi due volte con la stessa matricola o mail.

2) LABORATORIO

Nome Campo	Descrizione	Tipo	Lunghezza	Vincoli
Id Laboratorio	Codice identificativo del laboratorio	Int	5	Primary key, not null, unique
Matricola	Matricola del tecnico responsabile	Char	6	Foreign Key → Utente(matricola), Not null, unique
Nome	Nome del laboratorio	Varchar	100	Not null
Edificio	Edificio in cui si trova il laboratorio	Varchar	100	Not null
Piano	Piano dell'edificio	Integer	–	Not null
Stanza	Numero della stanza	Varchar	20	Not null

Vincoli di tupla

- Ad un tecnico può essere associato un solo laboratorio;
- Non possono esserci due laboratori con lo stesso Id;

3) ATTREZZATURA

Nome Campo	Descrizione	Tipo	Lunghezza	Vincoli
Codice_inventario	Codice inventario identificativo	Int	5	Primary Key, Unique, Not null
Laboratorio	Id laboratorio a cui è assegnata	Int	5	Foreign Key → Laboratorio(id), Not null
Tipo	Tipo di attrezzatura	Varchar	100	Not null
Marca	Marca dell'attrezzatura	Varchar	100	Not null
Stato	Stato dell'attrezzatura (es. disponibile)	Enum	-	Not null, valori ammessi ['Funzionante', 'In manutenzione', 'Non disponibile']

4) PRENOTAZIONE LABORATORIO

Nome Campo	Descrizione	Tipo	Lunghezza	Vincoli
Id prenotazione laboratorio	Identificativo prenotazione	Int	10	Primary key, not null, unique
Matricola	Matricola del professore che ha effettuato la prenotazione	Char	6	Foreign Key → Utente(matricola), Not null
Laboratorio	Id del laboratorio prenotato	Int	5	Foreign Key → Laboratorio(id), Not null
Esperimento	ID dell'esperimento associato	Int	5	Foreign Key → Esperimento (Id_esperimento), Not null
Data	Data della prenotazione	Date	–	Not null
Ora_inizio	Ora di inizio della prenotazione	Time	–	Not null
Ora_fine	Ora di fine della prenotazione	Time	–	Not null

Gestione prenotazioni

- Un professore può effettuare più prenotazioni;
- Non è possibile prenotare un laboratorio se questo è già stato prenotato;

5) PRENOTAZIONE ATTREZZATURA

Nome Campo	Descrizione	Tipo	Lunghezza	Vincoli
Id prenotazione attrezzatura	Identificativo prenotazione	Int	10	Primary key, not null, unique
Matricola	Matricola del professore che ha effettuato la prenotazione	Char	6	Foreign Key → Utente(matricola), Not null
Codice Inventario	Codice inventario attrezzatura prenotata	Int	5	Foreign Key → Attrezzatura (codice_inventario), Not null
Esperimento	Id dell'esperimento associato	Intr	5	Foreign Key → Esperimento (Id_esperimento), Not null
Data	Data della prenotazione	Date	–	Not null
Ora_inizio	Ora di inizio della prenotazione	Time	–	Not null
Ora_fine	Ora di fine della prenotazione	Time	–	Not null

Gestione prenotazione

- Un professore può prenotare una o più attrezzature;
- Un professore non può prenotare attrezzature in stato 'In manutenzione' o 'Non disponibile'
- Un professore non può prenotare attrezzature già prenotate da altri professori

6) PROGETTO SPERIMENTALE

Nome Campo	Descrizione	Tipo	Lunghezza	Vincoli
Id progetto	Identificativo univoco del progetto	Int	5	Primary Key, Unique, Not null
Matricola	Matricola del professore responsabile	Char	6	Foreign Key → Utente(matricola), Not null
Titolo	Titolo del progetto	Varchar	100	Not null
Descrizione	Descrizione dettagliata	Text	–	Not null
Obiettivi	Obiettivi del progetto	Text	–	Not null
Data_inizio	Data di inizio del progetto	Date	–	Not null
Data_fine	Data di fine del progetto	Date	–	Not null
Max_posti	Numero massimo di partecipanti	Integer	default = 1 1 < x < 30	Not null, Check > 0

7) PARTECIPAZIONE PROGETTO

Nome Campo	Descrizione	Tipo	Lunghezza	Vincoli
Id partecipazione	Identificativo della partecipazione al progetto	Int	5	Primary key, unique, not null
Id progetto	ID del progetto a cui si partecipa	Int	5	Foreign Key → Progetto_Sperimentale (Id_progetto), Not null
Matricola	Matricola dello studente partecipante	Char	6	Foreign Key → Utente(matricola), Not null

Gestione partecipazione

- Uno studente può effettuare una o più partecipazione a progetti sperimentali

- Uno studente non può prenotarsi ad un progetto che ha raggiunto il numero massimo di studenti

8) ESPERIMENTO

Nome Campo	Descrizione	Tipo	Lunghezza	Vincoli
Id_esperimento	Identificativo univoco dell'esperimento	Int	5	Primary Key, Unique, Not null
Id progetto	ID del progetto associato	Intr	5	Foreign Key → Progetto_Sperimentale (Id_progetto), Not null
Titolo	Titolo dell'esperimento	Varchar	100	Not null
Descrizione	Descrizione dettagliata	Text	200	Not null
Obiettivi	Obiettivi specifici	Text	100	Not null
Data_inizio	Data di inizio dell'esperimento	Date	–	Not null
Data_fine	Data di fine dell'esperimento	Date	–	Not null

Vincoli di tupla

- Non può esserci più di un esperimento in contemporanea per lo stesso progetto

9) PARTECIPAZIONE ESPERIMENTO

Nome Campo	Descrizione	Tipo	Lunghezza	Vincoli
Id partecipazione	Identificativo della partecipazione all'esperimento	Int	5	Primary key, unique, not null
Id esperimento	ID dell'esperimento prenotato	Int	–	Foreign Key →Esperimento (Id_esperimento), Not null
Matricola	Matricola dello studente	Char	6	Foreign Key →Utente(matricola), Not null

Gestione partecipazione

- Uno studente può partecipare ad un esperimento solo se fa parte del progetto sperimentale associato