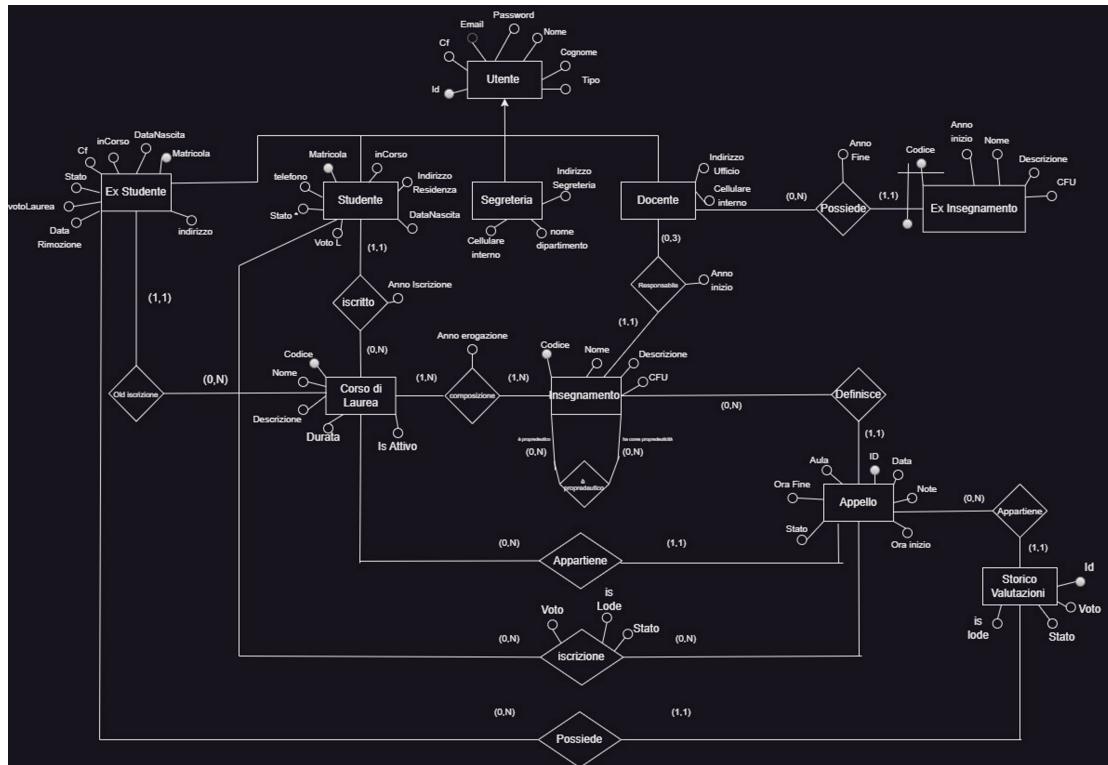


## Documentazione

### Schema E/R:



#### - Entità Utente :

L'entità utente rappresenta un generico utente che utilizza la base di dati. Utente è il padre della generalizzazione, nella quale, identifichiamo tre possibili categorie di utenti :

- Studente
- Segreteria
- Docente
- Ex Studente

La generalizzazione è pertanto totale ed esclusiva.

L'entità Utente, inoltre ,contiene una serie di caratteristiche/proprietà comuni a tutti gli utenti.

Utente inoltre contiene anche un attributo stato, il suo dominio sarà ( studente, docente, segretario), permettendo di smistare e identificare gli accessi degli utenti all'atto del login. Un ex studente, invece, accederà alla piattaforma come un normale studente, il suo stato di ex studente sarà verificato dopo il login e gli verranno revocate alcune funzionalità.

#### - Entità Studente :

L'entità, rappresenta uno studente iscritto attualmente ad un determinato corso di laurea.

- Viene identificato univocamente dalla sua matricola;
- L'attributo inCorso è un booleano e determina se lo studente è in corso, in base alla durata del corso di laurea e all'anno attuale. Esisteranno delle procedure in grado di aggiornarlo automaticamente.

- Attributo stato = L'attributo stato sarà inizializzato a null, i valori che potrà assumere saranno :

  - ◆ Ritirato
  - ◆ Laureato

- Voto di laurea = Tale attributo sarà inizializzato a null, assumerà un interno nel dominio [60,110], quando lo stato sarà laureato.

- Troviamo inoltre una serie di informazioni di contorno, come ad esempio : il numero di telefono, la data di nascita, l'indirizzo di residenza.

Uno studente è iscritto ad un solo corso di laurea.

Lo studente, inoltre ,sarà in grado di iscriversi agli appelli di un esame del suo corso di laurea.

#### - Entità segreteria :

L'entità rappresenta un segretario che lavora per l'università.

L'entità contiene una serie di attributi quali :

- Indirizzo della segreteria

- Nome del dipartimento
- Numero di telefono interno

Le azioni che un segretario potrà compiere saranno implementate a livello di applicazione web.

#### - Entità Docente

L'entità rappresenta un docente che lavora per l'università.

Un docente può tenere da 0 ( docente inattivo che non viene cancellato) fino a 3 insegnamenti, verrà tenuto traccia anche della data di inizio della docenza per un particolare corso.

Un docente può definire una serie di appelli per gli insegnamenti che tiene.

L'entità inoltre contiene i seguenti attributi :

- Indirizzo dell'ufficio del docente
- Numero di cellulare interno

Un docente inoltre possiederà anche uno storico di insegnamenti che ha tenuto nei vari anni.

Ipotizzo dunque che un docente NON venga mai cancellato dalla base di dati a meno che sia responsabile di nessun insegnamento.

Per questo motivo, all'atto della creazione di un docente non è necessario assegnarlo come responsabile ad un corso.

#### - Ex-Insegnamento

Un docente può possedere diversi ex-insegnamenti ovvero insegnamenti che ha tenuto in precedenza che ora sono affidati ad altri docenti.

L'entità viene identificata univocamente dal codice dell'insegnamento e dal docente che lo ha tenuto,  
nella base di dati vado dunque a supporre che un certo insegnamento tolto ad un docente non possa più essere assegnato a quel docente.

Troviamo inoltre nell'entità i seguenti attributi :

- Data di inizio e di fine della docenza
- Nome del corso
- Descrizione del corso
- Numero di Cfu.

Anche se gli insegnamenti non possono essere cancellati, decido di duplicare gli attributi e salvarli all'interno dell'entità ex insegnamento, in quanto alcuni di loro, come la descrizione, possono essere modificati.

#### - Corso di laurea

L'entità rappresenta un corso di laurea tenuto dall'università.

Ogni corso di laurea presenta un attributo booleano isAttivo, il suo dominio sarà:

- True := Il corso di laurea è attivo, il piano di studi presenta degli insegnamenti.
- False := il corso di laurea è inattivo, il piano di studi non presenta alcun insegnamento. Dei nuovi studenti non possono iscriversi ad un corso di laurea inattivo.

Suppongo, inoltre, che all'interno della mia base di dati anche se un corso è inattivo NON venga cancellato.

Un corso di laurea può essere composto da un insegnamento fino ad un massimo di N insegnamenti.

Un corso di laurea viene identificato univocamente dal suo codice.

All'interno dell'entità troviamo i seguenti attributi :

- Nome del corso
- Descrizione del corso
- Durata ( triennale o magistrale)

#### - Insegnamento

Un insegnamento è identificato da un codice (incrementale), esso potrà essere condiviso da più corsi di laurea.

Esiste per tanto una relazione N ad N tra corsi di laurea ed insegnamenti, essa prenderà il nome di piano di studi e conterrà :

- L'id dell'insegnamento
- Il cdl in cui è inserito
- L'anno di erogazione dell'insegnamento all'interno del cdl.

Un insegnamento inoltre presenta un certo responsabile, ovvero il docente che tiene il corso.

Un insegnamento non potrà essere creato senza specificare il docente responsabile.

Presenta i seguenti attributi :

- Nome
- Descrizione
- CFU

Un insegnamento può avere da 0 fino ad N propedeuticità e può essere propedeutico per 0 o N insegnamenti (nello schema, tale scenario viene identificato attraverso una relazione ricorsiva).

Uno studente dovrà svolgere gli esami propedeutici prima di poter conseguire un certo esame.

Suppongo che una volta inseriti gli insegnamenti NON vengano eliminati.

#### - Appello

Tale entità, rappresenta un appello di un certo esame. Un appello verrà identificato da un id auto-incrementale, conterrà anche i seguenti attributi :

- Data = data in cui si svolge l'appello
- Ora Inizio = ora di inizio dell'appello
- Ora fine = ora di fine dell'appello
- Note = note facoltative inserite da un docente riguardanti l'appello.
- Stato = il suo dominio può essere aperto o chiuso :
  - ◆ Aperto = Un appello è in tale stato se la data non è passata e se manca più di un'ora prima del suo inizio.
  - ◆ Chiuso = Quando un appello è in tale stato, gli studenti non possono iscriversi, né cancellare la loro iscrizione.
- Codice dell'insegnamento a cui fa riferimento.
- Id del docente responsabile.

Tutta via a differenza degli insegnamenti che sono condivisi, gli appelli avvengono per cdl. Uno studente si può iscrivere ad un appello solo se riguardante il suo cdl.  
Vado a supporre inoltre che gli appelli non vengano mai eliminati

- Esisto di un esame

Uno studente iscritto ad un certo appello, può visionare l'esito dell'esame nella relazione iscrizione.

L'iscrizione ad un esame sarà identificata dalla matricola dello studente e dall'id dell'appello, di conseguenza uno studente si potrà iscrivere a più appelli di un certo esame.

Nella relazione esito troviamo i seguenti attributi :

- **Voto esame** = Può essere null quando lo stato dell'iscrizione ha come valore "Iscritto", "in attesa", "ritirato". Altrimenti contiene un intero nel dominio [18,30];
  - **IsIode** = booleano, sarà possibile inizializzarlo a true, solamente quando il voto dell'esame è 30.
  - **Stato** = Lo stato dell'iscrizione è un valore nel seguente dominio :
    - ◆ Accettato = lo studente ha accettato il voto assegnatoli dal docente responsabile dell'appello
    - ◆ Rifiutato = lo studente ha rifiutato il voto assegnatoli dal docente responsabile dell'appello
    - ◆ Assente = lo studente è risultato assente all'appello
    - ◆ Iscritto = lo studente è iscritta all'appello
    - ◆ In attesa = lo studente ha ricevuto un esito da parte del docente responsabile, tale esito deve essere ancora accettato o rifiutato
    - ◆ Bocciato = lo studente ha ottenuto una valutazione < 18

Esisteranno delle procedure che saranno in grado di aggiornare automaticamente lo stato in base alle azioni compiute all'interno della applicazione web dagli utenti.

#### **- Ex studente e storico valutazioni**

Quando uno studente consegna una laurea, oppure decide di effettuare una rinuncia agli studi, le informazioni riguardanti il suo profilo da studente e la sua carriera, vengono trasferiti nelle entità Ex studente e Storico valutazioni.

**Al fine di permettere ad un ex-studente di iscriversi ad un nuovo corso di laurea** senza utilizzare un email diversa per la registrazione del suo account, ho deciso di mantenere in ex studente una chiave esterna al suo account situata nella relazione utente. Di conseguenza le credenziali di accesso saranno mantenute e gli verrà assegnata una nuova matricola. Inoltre dato che ho supposto la non cancellazione dei corsi di laurea anche se inattivi, ho deciso di mantenere un riferimento al corso di laurea frequentato precedentemente.

**Un ex-studente sarà identificato dalla matricola** (univoca) che possedeva mentre era iscritto ad un cdl. **Tale scelta, porta ai seguenti scenari:**

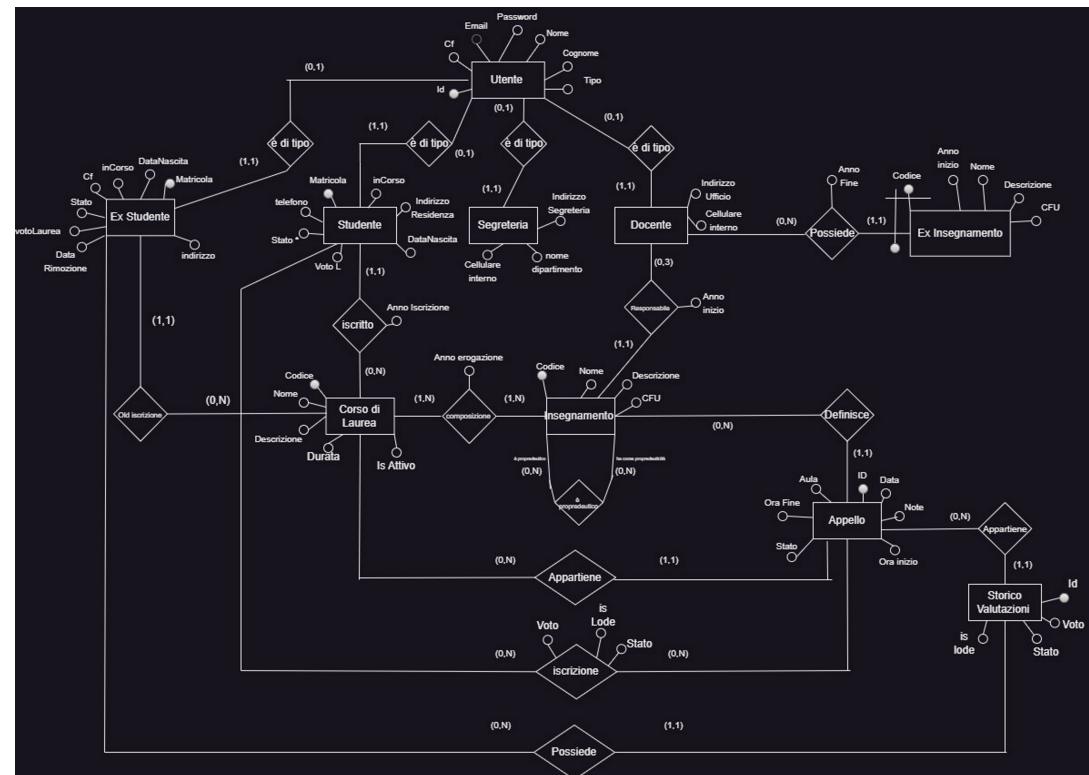
- Uno studente potrebbe riscriversi ad un cdl per cui ha già ottenuto una laurea in passato, tale possibilità sarà bloccata tramite un opportuno trigger.

- Tutta via ho lasciato che uno studente abbia la libertà di iscriversi ad un corso di laurea, per qui aveva effettuato una rinuncia agli studi in passato.

Un ex-studente può possedere da 0 ad N valutazioni. Ogni tupla all'interno dell'entità storico\_valutazioni è identificata da un id auto-incrementale. All'interno dell'entità, troviamo anche seguenti attributi :

- Voto
  - isLode
  - Stato
  - Id appello = si tratta di una chiave esterna che fa riferimento all'appello in cui è stato conseguito l'esito. Tale relazione è possibile in quanto avevo supposto la non cancellazione degli appelli.

## Ristrutturazione E/R :



Nella ristrutturazione dello schema E/R sono andato ad appiattire la generalizzazione tra Utente, Studente, Segreteria, Docente e ex studente. Ho deciso di andare a creare 4 associazioni una per ogni entità figlia della generalizzazione, in questo modo all'atto del login in base all'attributo tipo contenuto in utente, possono indirizzare l'accesso verso il corretto dominio di funzionalità. Vado dunque ad inserire una chiave esterna del id dell'utente in ognuna delle entità figlie.

L'altra possibilità sarebbe stata quella di accoppare le entità figlie nell'entità padre, ho deciso di evitare questa soluzione, in quanto il numero dei campi di ogni tupla della relazione utente sarebbe lievitato notevolmente aumentando il costo del singolo accesso.

## Schema Logico

Nome tabella	Attributi	Tipo	Chiave	Vincoli
Utente	idUtente	serial	PK	
	Nome	varchar (50)		Not null
	Cognome	varchar (100)		Not null
	Email	varchar (50)		Not null unique
	Password	varchar (32)		Not null
	Cf	varchar (16)		Not null unique
	Tipo	tipoUtente		Not null
Segretario	idUtente	integer	PK - FK	
	indirizzoSegreteria	varchar (100)		Not null
	nomeDipartimento	varchar (50)		Not null
	cellulareInterno	varchar (10)		Not null
Docente	idUtente	integer	PK - FK	
	indirizzoUfficio	varchar (100)		Not null
	cellulareInterno	varchar (10)		Not null
Insegnamento	Codice	serial	PK	
	Nome	varchar (50)		Not null
	Descrizione	varchar (200)		
	Cfu	integer		Not null
	annoInizio	integer		Not null
ExInsegnamento	codiceInsegnamento	integer	PK	
	idDocente	integer	FK - PK	
	Nome	varchar (50)		Not null
	Cfu	integer		Not null
	annoInizio	integer		Not null
Propedeuticità	annoFine	integer		Not null
	Esame	integer	FK - PK	
	Prop	integer	FK - PK	
	codiceLaurea	varchar (10)	FK - PK	
Appello	idAppello	serial	PK	
	codiceInsegnamento	integer	FK	
	Aula	varchar (20)		Not null
	Note	varchar (200)		
	dataEsame	DATE		Not null
	oralInizio	TIME		Not null
	oraFine	TIME		Not null
	statoAppello	tipostatoAppello		Not null
CorsoDiLaurea	Cdl	varchar (10)	FK	
	Codice	varchar (10)	PK	
	Nome	varchar (50)		Not null
	Descrizione	varchar (200)		
	isAttivo	bool		Not null
Studente	Durata	tipocorsoLaurea		Not null
	Matricola	serial	PK	
	Telefono	varchar (20)		Not null
	indirizzoResidenza	varchar (100)		Not null
	dataNascita	DATE		Not null
	annolscrizione	DATE		Not null
	inCorso	bool		Not null
	idUtente	integer	FK	
Stato	idCorso	varchar	FK	
	Tipo	tipostatoExStudente		

	votoLaur	votolaurea		
IscrizioneEsame	Matricola	integer	FK - PK	
	Id	integer	FK - PK	
	votoEsame	voto		
	Stato	tipoStatoVoto		
	isLode	bool		
ExStudente	Matricola	integer	PK	
	Nome	varchar (100)		Not null
	Cognome	varchar (100)		Not null
	Telefono	varchar (20)		Not null
	indirizzoResidenza	varchar (100)		Not null
	dataNascita	DATE		Not null
	annoscrizione	DATE		Not null
	inCorso	bool		
	Stato	tipoStatoExStudente		
	dataRimozione	DATE		Not null
	votoLaurea	votolaurea		
	codiceCorso	varchar (10)	FK	
	idUtente	integer	FK	
StoricoValutazioni	Id	serial	PK	
	votoEsame	voto		
	Stato	tipoStatoVoto		
	isLode	bool		
	idAppello	integer	FK	
	Matricola	integer	FK	
PianoStudi	codiceCorso	Varchar (10)	FK - PK	
	codiceInsegnamento	integer	FK - PK	
	annoErogazione	annoEsame		

Vincoli di integrità referenziale :

- Studente.idCorso è in vincoli di integrità referenziale con CorsoDiLaurea.Codice
- Studente.idUtente è in vincoli di integrità referenziale con Utente.id
- Docente.idutente è in vincoli di integrità referenziale con Utente.id
- Insegnamento.idDocente è in vincolo di integrità referenziale con Docente.idUtente
- Propredeuticità.Codice1 è in vincolo di integrità referenziale con Insegnamento.Codice
- Propredeuticità.Codice2 è in vincolo di integrità referenziale con Insegnamento.Codice
- ExInsegnamento.idutente è in vincolo di integrità referenziale con Docente.idUtente
- Appello.Codice\_insegnamento è in vincolo di integrità referenziale con Insegnamento.Codice
- Appello.idDocente è in vincolo di integrità referenziale con Docente.idUtente
- Iscrizione.Matricola è in vincolo di integrità referenziale con Studente.Matricola
- Iscrizione.idAppello è in vincolo di integrità referenziale con Appello.id
- StoricoValutazioni.Ex Matricola è in vincolo di integrità referenziale con Appello.id
- StoricoValutazioni.Ex Matricola è in vincolo di integrità referenziale con ExStudente.Matricola
- ExStudente.CodiceCorso è in vincolo di integrità referenziale con CorsoDiLaurea.Codice
- PianoStudi.CodiceCorso è in vincolo di integrità referenziale con CorsoDiLaurea.codice
- PianoStudi.CodiceInsegnamento è in vincolo di integrità referenziale con Insegnamento.codice

## Script Significativi

Lato database, alcuni degli script più significativi, sono i seguenti :

- [Query che permette ritorna tutti gli studenti che possono laurearsi per un certo cdl.](#)

Uno studente può laurearsi, se presenta una valutazione sufficiente accettata per tutti gli esami presenti nel piano di studi del suo corso di laurea.  
Tale condizione è facilmente implementabile tramite una divisione.

```
select s.matricola , u.nome , u.cognome , u.indirizzoResidenza , s.datanascita , s.annoscrizione , s.incorso , c.nome , s.idcorso
from "UniNostra".studente s inner join "UniNostra".utente u on s.idutente = u.idutente inner join "UniNostra".corsodilaurea c on c.codice = s.idcorso
where s.idcorso = idCdl and not exists (
    select *
    from "UniNostra".pianostudi p
    where p.codicecorso = s.idcorso and not exists (
        select *
        from "UniNostra".iscrizioneesame i inner join "UniNostra".appello a on a.idappello = i.id
        where i.matricola = s.matricola and a.codiceinsegnamento = p.codiceinsegnamento and a.cdl = s.idcorso and i.stato = 'Accettato'
    )
);
```

#### - Trigger per il controllo delle propedeuticità cicliche

Tale trigger permette di controllare che l'inserimento di una particolare propedeuticità, non crea cicli con altre propedeuticità dello stesso cdl.  
Per semplicità indico la relazione di propedeuticità con il simbolo ">". Esame A ha come propedeuticità esame B (Esame A -> Esame B)

Caso base :

Esame A -> Esame B e Esame B -> Esame A

Esame A -> Esame B, Esame B -> Esame C Esame C -> Esame A

Passo :

Supponiamo di avere i seguenti record nella tabella delle propedeuticità :

- Esame A -> Esame B
- Esame B -> Esame C

Supponiamo di voler inserire la propedeuticità Esame C -> Esame A, attraverso una query ricorsiva vado a risalire la "gerarchia di propedeuticità", andando a cercare , nel nostro esempio, un record che abbia l'Esame A propedeutico ad un altro Esame. Continuo tale procedimento, fino a quando non trovo un percorso che mi porti ad un record che abbia come esame propedeutico C.

```
create or replace function "UniNostra".controllaProp()
returns trigger as $$ 
declare
    lv integer;
    c1 integer;
    c2 integer;
    co varchar(10);
begin
    perform * from "UniNostra".propedeuticita p where new.prop = p.esame and new.esame = p.prop and new.codicelaurea = p.codicelaurea;
    if found then
        raise exception 'impossibile inserire la propedeucita % -> % causerebbe un ciclo', new.esame ,new.prop ;
    end if;

    with recursive sos (LEVEL, corso1, corso2, codicelaurea) as (
        select 1, p.esame as corso1, p.prop as corso2 , p.codicelaurea  from "UniNostra".propedeuticita p
        where p.esame = new.esame and p.codicelaurea = new.codicelaurea
        union
        select o.LEVEL + 1, next_p.esame, next_p .prop , next_p.codicelaurea
        from "UniNostra".propedeuticita as next_p, sos as o
        where o.corso2 = next_p.esame  and o.codicelaurea = next_p.codicelaurea and o.corso2 <> new.esame
    )
    select LEVEL, corso1,corso2, codicelaurea into lv,c1,c2,co from sos order by level desc limit 1;
    if c2 = new.esame then
        raise exception 'Non è possibile inserire la propedeuticità, altrimenti causerebbe un ciclo';
    end if;
    return new;
end;
$$ language plpgsql;
```

```
CREATE OR REPLACE TRIGGER controllaPropedeuticita after insert on "UniNostra".propedeuticita
FOR EACH ROW EXECUTE FUNCTION "UniNostra".controllaProp();
```

#### - Trigger che disiscrive uno studente dagli appelli di un certo insegnamento quando accetta un voto per quest'ultimo

Tale trigger ha come evento scatenante l'aggiornamento di un record della tabella riguardante le iscrizioni agli esami degli studenti.

In particolare, mi interessa lo scenario in cui uno studente accetta un voto assegnatoli da un certo docente ad un esame a cui era iscritto. Tale realtà viene catturata dal cambio di valore dell'attributo stato, il quale passa da "In attesa" a "Accettato".

Dopo di che, vengono eliminati tutti i record riguardanti le iscrizioni dello studente ad altri appelli (aperti), dello stesso insegnamento.

```
create or replace function "UniNostra".disiscriviStudente()
returns trigger as $$ 
declare
    codIns "UniNostra".appello.codiceinsegnamento %type;
    idApp "UniNostra".iscrizioneesame.id%type;
begin
    select a.codiceinsegnamento into codIns from "UniNostra".appello a where a.idappello = new.id;

    if old.stato = 'In attesa' and new.stato = 'Accettato' then
        for idApp in select i.id from "UniNostra".iscrizioneesame i inner join "UniNostra".appello a on i.id = a.idappello
        where i.id <> new.id and a.codiceinsegnamento = codIns and i.matricola = new.matricola and (i.stato = 'Iscritto' or i.stato = 'In attesa')
            loop
                delete from "UniNostra".iscrizioneesame i where i.matricola = new.matricola and i.id = idApp;
            end loop;
    end if;
    return new;
end;
$$ language plpgsql;
```

```
CREATE OR REPLACE TRIGGER eliminaIscrizioniPassate after update on "UniNostra".iscrizioneesame
FOR EACH ROW EXECUTE FUNCTION "UniNostra".disiscriviStudente();
```

#### - Trigger che permette di archiviare uno studente

Tale trigger ha come evento scatenante l'eliminazione di uno studente dalla relativa tabella.

Le informazioni dello studente vengono acquisite e copiate nei campi della tabella ex-studente, inoltre ogni esame presente nella sua carriera viene inserito nella tabella ex-valutazioni.

```

CREATE OR REPLACE FUNCTION "UniNostra".storicoStudente()
RETURNS TRIGGER AS $$ 
DECLARE
    iscrizioneApp "UniNostra".iscrizioneesame%rowtype;
    oldUtente "UniNostra".utente%rowtype;
BEGIN

    select u.idutente ,u.nome ,u.cognome ,u.email ,u."password" ,u.tipo ,u.cf into oldUtente
    from "UniNostra".studente s inner join "UniNostra".utente u on s.idutente = u.idutente
    where s.matricola = old.matricola ;

    insert into
    "UniNostra".exstudente(matricola,nome,cognome,telefono,indirizzoResidenza,dataNascita,annoIscrizione,incorso,stato,datarimozione,"votolaurea",codicecorso,idutente)
    values(old.matricola,oldUtente.nome,oldUtente.cognome,old.telefono,old.indirizzoResidenza,old.dataNascita,old.annoIscrizione,old.incorso,old.stato,CURRENT_DATE,old.votoLaur,old.idCorso,old.idUtente);

    for iscrizioneApp in select * from "UniNostra".iscrizioneesame i where i.matricola = old.matricola
    loop
        insert into "UniNostra".storicovalutazioni (votoesame,stato,idAppello,matricola,isLode)
        values(iscrizioneApp.votoEsame,iscrizioneApp.stato,iscrizioneApp.id,iscrizioneApp.matricola,iscrizioneApp.isLode);

    end loop;
    delete from "UniNostra".iscrizioneesame i where i.matricola = old.matricola;

    RETURN OLD;
END;
$$ LANGUAGE plpgsql;
CREATE OR REPLACE TRIGGER storicoStud before delete on "UniNostra".studente
FOR EACH ROW EXECUTE function "UniNostra".storicoStudente();

```

- Per quanto riguarda l'applicazione web, ho cercato di riutilizzare gli stessi elementi in varie pagine, **creando così dei componenti astratti**.

Un esempio di componente può essere ad esempio la creazione delle tabella. La quale può essere personalizzata in base al numero di righe e di colonne che si vogliono avere e in base ai campi che si vogliono disporre all'interno di essa.

```

function creaIntestazione($valori) {
    echo "<thead class='thead-dark' id = 'testaTabella'><tr>";
    foreach ($valori as $titolo) {
        echo "<th scope='col'>".$titolo."</th>";
    }
    echo "</tr></thead>";
}
function creaColonne($valori,$cont){
    echo "<tr>";
    echo "<th scope='row'>".$cont."</th>";
    foreach ($valori as $titolo) {
        echo "<td>".$titolo."</td>";
    }
    echo "</tr>";
}
function eliminaTabella($id){
    echo "<script language='JavaScript' type='text/javascript'>
        var element = document.getElementById('".$id."');
        element.remove();
    </script>";
}

```

## **Lista e Descrizione di ogni procedura**

Nome procedura	Parametri	Descrizione	Eccezioni Sollevate
<a href="#">inserisciCorsolaurea</a>	<ul style="list-style-type: none"> <li>• Codice del Corso (varchar)</li> <li>• Nome del corso (varchar)</li> <li>• Descrizione(varchar) *</li> <li>• Durata (magistrale o triennale)</li> </ul>	Aggiungi un nuovo corso di laurea	<ul style="list-style-type: none"> <li>• Se la durata non è magistrale o triennale.</li> <li>• Se esiste già un corso con lo stesso codice o con lo stesso nome e la stessa durata</li> </ul>
<a href="#">aggiornaStatoCorso</a>	<ul style="list-style-type: none"> <li>• id del corso di laurea (varchar)</li> <li>• stato (boolean)</li> </ul>	Funzione che permette di attivare e disattivare un corso di laurea, sarà utilizzata internamente	<ul style="list-style-type: none"> <li>• Se il corso di laurea inserito come parametro non esiste</li> </ul>
<a href="#">inserisciInsegnamento</a>	<ul style="list-style-type: none"> <li>• Nome dell'insegnamento (varchar)</li> <li>• Descrizione(varchar) *</li> <li>• Cf (integer)</li> <li>• IdDocente(integer)</li> </ul>	Aggiunta di un insegnamento	<ul style="list-style-type: none"> <li>• Se il docente inserito non esiste.</li> <li>• Se il docente tiene già il corso che si vuole inserire.</li> <li>• Se esiste già un insegnamento con li stessi crediti e nome</li> </ul>
<a href="#">updateDocenza</a>	<ul style="list-style-type: none"> <li>• id dell'insegnamento che si vuole aggiornare (integer)</li> <li>• id del nuovo docente (integer)</li> </ul>	Aggiornamento responsabile di un corso	<ul style="list-style-type: none"> <li>• Se l'insegnamento inserito non esiste</li> <li>• Se il nuovo responsabile tiene già 3 insegnamenti</li> </ul>
<a href="#">aggiungiUtente</a>	<ul style="list-style-type: none"> <li>• Nome(varchar)</li> <li>• Cognome(varchar)</li> <li>• Password(varchar)</li> <li>• Email(varchar)</li> <li>• tipo (tipoUtente)</li> <li>• Cf (varchar)</li> </ul>	Aggiungi un nuovo utente, procedura interna utilizzata per aggiungere studenti, segretari o docenti.	
<a href="#">aggiornaCredenzialiUtente</a>	<ul style="list-style-type: none"> <li>• email (varcahr)</li> <li>• password(varchar)</li> <li>• newEmail *(varchar)</li> </ul>	<ul style="list-style-type: none"> <li>• Modifica delle credenziali di accesso di un utente. L'utente viene riconosciuto attraverso</li> </ul>	<ul style="list-style-type: none"> <li>• Se l'email non esiste nel db</li> <li>• La password associata all'email inserita non esiste</li> </ul>

	• newPassword *(varchar).	l'email e la password • i campi che non vogliono essere modificati andranno lasciati a null.	
<a href="#">aggiungiDocente</a>	• Nome (varchar) • Cognome (varchar) • Password (varchar) • Email (varchar) • Tipo (tipoUtente) • Cf (varchar) • indirizzoUfficio (varchar) • cellulareInterno (varchar)	Aggiunta di un docente	• Email già esistente nel database
<a href="#">aggiungiStudente</a>	• nome (varchar) • Cognome (varchar) • Email (varchar) • Password (varchar) • Cf (varchar) • Telefono (varchar) • indirizzo residenza (varchar) • dataNascita (date) • annoIscrizione (integer) • idCorso (varchar)	Inserimento di uno studente nella base di dati, saranno richieste le informazioni dello studente e il corso di laurea a cui si vuole iscrivere. Procedura utilizzata internamente dalla segreteria	• Se il corso di laurea non è attivo lo studente non può iscriversi. • Se il corso di laurea inserito non esiste. • Se il corso di laurea non è attivo. • Se lo studente risulta già iscritto ad un corso di laurea
<a href="#">updateInfoStudente</a>	• Matricola (integer) • telefono varchar(20) • DataNascita date • residenza varchar(100).	Procedura che permette il cambio di alcune informazioni anagrafiche per uno studente.	• Matricola inesistente
<a href="#">inserisciSegretario</a>	• nomeU varchar(50) • cognomeU varchar(100) • emailU varchar(50) • passwordU varchar(32) • cfU varchar(16) • indirizzoSegreteria varchar(100) • nomeDipartimento varchar(50) • cellulareInterno varchar(10)	Funzione per l'inserimento di un segretario, può essere utilizzata solamente da altri segretari	• Segretario già registrato
<a href="#">inserisciPianoStudi</a>	• Codice corso di laurea (varchar) • id dell'insegnamento (integer) • annoErogazione (annoEsame)	Funzione per inserimento insegnamenti nei corsi di laurea, gli insegnamenti sono condivisi tra corsi di laurea	• Se il codice del corso di laurea non esiste • Se il codice dell'insegnamento non esiste • Se l'anno del insegnamento non è valido
<a href="#">inserisciPropedeuticità</a>	• Codice Insegnamento 1 (integer) • Codice insegnamento 2 propedeutico (integer) • codiceCDL (varchar)	• Funzione per inserimento propedeuticità tra insegnamenti. Viene inserito per prima l'insegnamento a cui ci si vuole riferire e successivamente l'insegnamento propedeutico per il superamento del primo. • La propedeuticità vale per il corso di laurea specificato, in cui devono far parte entrambi gli insegnamenti.	• Insegnamento 1 non esistente, insegnamento 2 non esistente. • Se Codice1 = Codice2, gli insegnamenti devono essere diversi. • Gli insegnamenti devono appartenere allo stesso corso di laurea.
<a href="#">eliminaPropedeuticità</a>	• idCorso (Integer) • idCorsoPropedeutica(integer) • idCdl varchar(10)	Procedura che permette di eliminare una certa propedeuticità da un certo cdl.	• Nessun insegnamento con id inserito • Nessun insegnamento con id propedeuticità inserito • Nessun corso con id cdl inserito • Nel manifesto di studi del cdl inserito non esistono gli esami con id inseriti • Non esiste la propedeuticità idCorso -> idCorsoPropedeutico nel cdl inserito
<a href="#">inserimentoAppello</a>	• idInsegnamento (integer) • idDocente (integer) • aula (varchar) • note (varchar) • dataEsame (date) • oraInizio (time) • oraFine (time)	• Funzione per i docenti responsabili degli insegnamenti, permette di inserire un appello in una certa data per gli insegnamenti che detiene specificando il corso di laurea. • L'appello deve essere programmato almeno un giorno prima	• Se il docente non è responsabile dell'insegnamento. • Se il codice di laurea inserito non esiste. • Se l'ora di fine non viene dopo l'ora di inizio. • L'esame deve durare almeno 15 minuti. • Se esiste già un appello dello stesso insegnamento e dello stesso cdl che si sovrappone. • Se l'aula è già occupata a quel ora.
<a href="#">aggiornaStatoAppello</a>	• idAppello (integer)	• Funzione che permette di aggiornare lo stato di una appello, uno studente si può iscrivere ad un appello solo se esso è aperto, ovvero fino a un ora prima dell'ora di inizio dell'esame. • Update stato appello da chiamare ogni volta che lo studente refresha la pagine delle iscrizioni	• Se non esiste nessun appello con l'id inserito
<a href="#">inserisciliscrizioneEsame</a>	• idAppello(integer) • matricola(integer)	• Funzione per l'iscrizione di uno studente ad un esame. L'utente può iscriversi solo se l'appello non è chiuso (data passata) e fino ad un ora prima dell'esame. • Un utente inoltre, può iscriversi solo agli esami previsti dal suo cdl.	• Nessun studente con matricola inserita • Nessun appello con id inserito

<u>accettaVoto</u>	<ul style="list-style-type: none"> <li>• idAppello (integer)</li> <li>• matricola(integer)</li> <li>• accettato (bool)</li> </ul>	<ul style="list-style-type: none"> <li>• Funzione che permette ad uno studente di accettare o rifiutare l'esito di un appello per un certo insegnamento.</li> <li>• L'accettazione viene richiesta all'utente tramite una variabile booleana se è false l'esito è accettato altrimenti l'esito è rifiutato.</li> </ul>	<ul style="list-style-type: none"> <li>• Lo studente non è iscritto all'appello d'esame</li> <li>• La sua iscrizione deve essere inAttesa e l'appello deve essere chiuso</li> </ul>
<u>eliminaliscrizione</u>	<ul style="list-style-type: none"> <li>• matricola(integer)</li> <li>• idAppello(integer)</li> </ul>	Funzione che permette ad uno studente di disiscriversi da un appello a cui risulta iscritto.	<ul style="list-style-type: none"> <li>• Se lo studente non è iscritto all'appello inserito</li> <li>• Se l'appello è chiuso o l'utente si disiscrive a meno di un ora dall'inizio.</li> </ul>
<u>registraVotoEsame</u>	<ul style="list-style-type: none"> <li>• idDocente(integer)</li> <li>• matricola(integer)</li> <li>• idAppello(integer)</li> <li>• voto* (votoEsame)</li> <li>• lode (bool)</li> </ul>	Funzione che permette ad un docente di registrare il risultato di un esame.	<ul style="list-style-type: none"> <li>• Docente non responsabile dell'appello inserito.</li> <li>• L'appello deve essere chiuso.</li> <li>• Lo studente deve essere iscritto all'appello.</li> <li>• La lode non può essere true se il voto non è 30.</li> <li>• Lo studente ha già un voto registrato</li> </ul>
<u>aggiornaInCorso</u>	<ul style="list-style-type: none"> <li>• Matricola(integer)</li> </ul>	Funzione che permette di aggiornare la variabile inCorso di uno studente.	<ul style="list-style-type: none"> <li>• La matricola inserita non esiste</li> </ul>
<u>registraLaurea</u>	<ul style="list-style-type: none"> <li>• Matricola (integer)</li> <li>• status (tipoSatoExStudente)</li> <li>• votoLaurea "UniNostra".votoLaurea</li> </ul>	Funzione che permette ad un segretario di registrare la laurea o il ritiro di uno studente.	<ul style="list-style-type: none"> <li>• La matricola inserita non esiste</li> <li>• Lo studente inserito non ha ancora passato tutti gli esami previsti dal suo cdl</li> </ul>
<u>rimuoviPianoStudi</u>	<ul style="list-style-type: none"> <li>• idInsegnamento (integer)</li> <li>• cdl(varchar(10))</li> </ul>	Procedura che dato l'id di un insegnamento e un cdl di cui fa parte, lo elimina dal piano di studi.	<ul style="list-style-type: none"> <li>• L'insegnamento non esiste</li> <li>• Il cdl non esiste</li> <li>• L'insegnamento non è previsto per quel cdl</li> <li>• Ci sono appelli aperti per il l'insegnamento che si vuole eliminare dal piano studi del cdl inserito</li> </ul>
<u>updateInsegnamento</u>	<ul style="list-style-type: none"> <li>• idInsegnamento (integer)</li> <li>• Descrizione (varchar(200))</li> <li>• cfu integer</li> <li>• idDoc (integer)</li> </ul>	Procedura che dato l'id di un insegnamento aggiorna le seguenti informazioni, se modificate : Descrizione, CFU, Responsabile insegnamento	<ul style="list-style-type: none"> <li>• idInsegnamento non valido</li> </ul>

### **Lista e Descrizione di ogni trigger**

Nome Procedura	Action	Descrizione	Eccezioni Sollevate
<u>eliminaProp</u>	<ul style="list-style-type: none"> <li>• Delete di insegnamento dal piano studi</li> </ul>	Trigger che agisce all'eliminazione di un insegnamento dal piano di studi di un cdl, eliminando tutte le propedeuticità assegnate e per l'esame eliminato	
<u>attivaDisattivaCdl</u>	<ul style="list-style-type: none"> <li>• Insert or delete sulla tabella PianoStudi</li> </ul>	Trigger che permette di disattivare un corso di laurea se non presenta insegnamenti o attivarlo se ne presenta almeno uno.	
<u>countDocentiResponsabili</u>	<ul style="list-style-type: none"> <li>• Insert o update sulla tabella degli insegnamenti</li> </ul>	Trigger che permette di assegnare un insegnamento ad un docente solo se è responsabile di meno di 3 corsi.	<ul style="list-style-type: none"> <li>• Se il docente è già responsabile di 3 corsi</li> </ul>
<u>storicoInsegnamentiDocenti</u>	<ul style="list-style-type: none"> <li>• Update della tabella insegnamenti</li> </ul>	Trigger che al cambio di docenza vada ad aggiungere nello storico del docente l'insegnamento che teneva.	<ul style="list-style-type: none"> <li>• Se il docente ha tenuto il corso in passato non li può essere riassegnato</li> </ul>
<u>controllaProp</u>	<ul style="list-style-type: none"> <li>• Inserimento di dati nella tabella delle propedeuticità</li> </ul>	Trigger che permette di evitare inserimenti di propedeuticità che causerebbero un ciclo, ad esempio : A -> B, B -> C, C -> A	<ul style="list-style-type: none"> <li>• Inserimento propedeuticità non consentita</li> </ul>
<u>controllaAppelli</u>	<ul style="list-style-type: none"> <li>• Inserimento di un appello nella tabella degli appelli.</li> </ul>	Trigger per verificare che non esistano già appelli di esami previsti nello stesso anno accademico di un cdl, previsti nella stessa data.	<ul style="list-style-type: none"> <li>• Se esiste già un altro appello nella stessa data per un corso previsto nello stesso anno accademico di un certo cdl</li> </ul>
<u>controlloAppello</u>	<ul style="list-style-type: none"> <li>• Inserimento su iscrizioneEsame</li> </ul>	<ul style="list-style-type: none"> <li>• Trigger per il controllo delle iscrizioni ad un appello da parte degli studenti.</li> <li>• Uno studente si può iscrivere ad un appello fino ad un'ora prima dell'esame ed a solo esami del suo cdl</li> </ul>	<ul style="list-style-type: none"> <li>• L'appello risulta chiuso</li> <li>• Iscrizione un'ora prima dell'inizio dell'esame</li> <li>• Lo studente si sta iscrivendo ad un appello di un insegnamento non del suo cdl</li> <li>• Lo studente non ha i requisiti di propedeuticità dell'esame</li> </ul>
<u>controllaPassatoVoti</u>	<ul style="list-style-type: none"> <li>• Insert sulla tabella iscrizioneEsami</li> </ul>	Trigger che controlla che uno studente all'atto dell'iscrizione ad un certo appello non abbia già un voto registrato come Accettato per quel insegnamento.	<ul style="list-style-type: none"> <li>• Se lo studente possiede un esito pendente (non ancora accettato o rifiutato) non si potrà iscrivere ad un nuovo appello.</li> <li>• Se lo studente ha già accettato un voto per lo stesso insegnamento</li> </ul>
<u>controllaTurni</u>	<ul style="list-style-type: none"> <li>• Inserimento nella tabella iscrizioni esami</li> </ul>	Trigger che controlla che uno studente si possa iscrivere solo ad un turno di un esame durante la stessa giornata.	<ul style="list-style-type: none"> <li>• Se lo studente risulta già iscritto ad un altro turno dello stesso esame durante la giornata</li> </ul>
<u>disiscriviStudente</u>	<ul style="list-style-type: none"> <li>• Update di iscrizione studente</li> </ul>	Trigger, quando uno studente	

		accetta un voto di un certo esame, viene automaticamente disiscritto da tutti gli altri appelli di quel esame a cui si era iscritto.	
<a href="#">storicoStudente</a>	• Delete sulla tabella studente	Trigger che all'eliminazione di uno studente (laureato o ritirato), sposta lo studente in ex studente e archivia la sua carriera.	

#### **Lista e Descrizione di ogni funzione, utilizzate dalla applicazione web**

Nome Funzione	Parametri	Descrizione	Eccezioni Sollevate
		<b>STUDENTI</b>	
<a href="#">login</a>	• email (text) • psw(text)	Funzione che permette di effettuare il login per un certo utente.	• Se l'email o la password risultano incorretti
<a href="#">profiloStudente</a>	• idUtente (integer)	Funzione che dato l'id di uno studente ritorna alcune sue informazioni.	• Se l'id inserito non esiste o non è uno studente
<a href="#">appelliAperti</a>	• idUtente(integer)	Funzione che, dato un id di uno studente, mostra gli appelli aperti per il suo cdl.	• L'id inserito non appartiene ad uno studente o non esiste • Eccezioni sollevate dal controllo degli appelli
<a href="#">iscrizioniAppelli</a>	• idUtente(integer)	Funzione che permette di far visionare ad un certo utente gli appelli d'esame a cui risulta iscritto.	• L'id utente inserito non appartiene ad uno studente
<a href="#">visualizzaEsamiPassati</a>	• idUtente (integer)	Funzione che, dato l'id utente di uno studente, restituisce tutti gli esami che ha accettato e superato.	• Se l'id utente inserito non appartiene ad uno studente
<a href="#">visualizzaCarriera</a>	• idUtente (integer)	Funzione che, dato l'id utente di uno studente, restituisce tutta la sua carriera.	• Se l'id utente inserito non appartiene ad uno studente
<a href="#">accettaVoti</a>	• idUtente(integer)	Funzione che dato l'id di un utente studente mostra tutti i voti che può rifiutare o accettare.	• Se l'id inserito non appartiene ad uno studente
<a href="#">visualizzaExCarriere</a>	• idUtente (integer)	Funzione che dato l'id utente di uno studente ritorna tutte le sue carriere passate	• Se l'id inserito non è di uno studente
<a href="#">visualizzaExVotiValidi</a>	• matricola (integer)	Funzione che, data la matricola di un ex-studente, ritorna la sua carriera passata.	• La matricola inserita non appartiene ad un ex-studente
<a href="#">visualizzaExVoti</a>	• matricola (integer)	Funzione che, data la matricola di un ex-studente, ritorna la sua carriera passata completa.	• La matricola inserita non appartiene ad un ex-studente
<a href="#">isStudente</a>	• idutente (integer)	Funzione che permette di indirizzare nelle corrette pagine uno studente oppure un ex-studente.	
<a href="#">visualizzaProfiloEx</a>	• idUtnete (integer)	Funzione che visualizza il profilo di un ex studente.	• L'id inserito non è di un ex studente
<a href="#">visualizzaCdl</a>	• idUtente(integer)	Funzione che dato l'id di uno studente permette di visualizzare gli insegnamenti del suo cdl.	• Se l'id inserito non è di uno studente
<a href="#">visualizzaInfoCdl</a>	• idUtente(integer)	Funzione che, dato l'id di uno studente, restituisce informazioni sul suo corso di laurea.	• Se l'id inserito non è di uno studente
<a href="#">visualizzaProp</a>	• idCdl (varchar(10)) • idIns (integer)	Funzione che dato l'id di un insegnamento e del cdl a cui fa riferimento restituisce tutte le propedeuticità per quel corso.	• Se il cdl non esiste • Se l'insegnamento non è in quel cdl
<a href="#">visualizzaTuttiCdl</a>	• idCdl (varchar(10))	Funzione che, dato il codice di un corso di laurea, restituisce tutti i suoi insegnamenti.	• L'id del corso di laurea inserito non è valido
<a href="#">visualizzaidCdl</a>	nessuno	Funzione che restituisce tutti gli id dei corsi di laurea presenti.	
		<b>DOCENTE</b>	
<a href="#">profiloDocente</a>	idUtente (integer)	Funzione che dato l'id di un docente permette di visualizzarne le informazioni.	• Se l'id inserito non appartiene ad un docente
<a href="#">cdlDocente</a>	• idUtente(integer)	Funzione che dato l'id di un docente ritorna i cdl in cui opera.	• Se l'id inserito non appartiene ad un docente
<a href="#">visualizzaCdlDoc</a>	• idCdl (varchar(10))	Funzione che, dato il codice di un corso di laurea, restituisce tutti i suoi insegnamenti tenuti dal docente selezionato.	• L'id del corso di laurea inserito non è valido
<a href="#">numInsegnamenti</a>	• idDocente (integer)	Funzione che dato l'id di un docente ritorna il numero dei suoi	• Se l'id inserito non è di un docente

		insegnamenti.	
<a href="#">idCorsiDoc</a>	• idUtente (integer)	Funzione che, dato l'id di un docente, restituisce gli id degli insegnamenti che detiene.	• L'id utente inserito non è di un docente
<a href="#">appelliApertiDoc</a>	• idDocente (integer)	Funzione che dato l'id di un docente restituisce gli appelli aperti.	• Se l'id inserito non appartiene ad un docente
<a href="#">numIscrittiA</a>	• idAppello (integer)	Funzione che dato l'id di un appello ritorna il numero di iscritti.	• Se l'id della appello non esiste
<a href="#">iscrittiAppello</a>	• idApp (integer)	Funzione che dato l'id di un appello ritorna tutti gli studenti iscritti.	• Se l'id dell'appello non esiste
<a href="#">appelliChiusiDoc</a>	• idDocente (integer)	Funzione che permette di visualizzare gli appelli chiusi di un docente.	• Se l'id inserito non appartiene al docente
<a href="#">numPartecipantiApp</a>	• idAppello (integer)	Funzione che dato l'id di un appello ritorna il numero di studenti che lo hanno fatto .	• L'id dell'appello inserito non esiste
<a href="#">studPartecipanti</a>	• idAppello (integer)	Funzione che dato l'id di un appello ritorna gli studenti partecipanti.	
<a href="#">studPartecipantiStorico</a>	• idAppello (integer)	Funzione che dato l'id di un appello ritorna gli studenti partecipanti nello storico.	
<a href="#">storicoInsegnamenti</a>	• idDocente (integer)	Funzione che dato l'id di un docente ritorna gli insegnamenti che ha tenuto in precedenza.	• L'id inserito non appartiene ad un docente
		<b>SEGRETARIO</b>	
<a href="#">profiloSegretario</a>	• idSegretario	Funzione che, dato l'id di un segretario, ritorna alcune informazioni riguardanti il suo profilo.	• L'id inserito non appartiene ad un segretario
<a href="#">visualizzaTuttiCorsi</a>	• idUtente (integer)	Funzione che ritorna tutti i corsi di laurea presenti nel db.	• Id utente non esistente
<a href="#">numIscrizioniCdl</a>	• idCdl (varchar(10))	Funzione che, dato l'id di un cdl, restituisce il numero di studenti iscritti ad esso.	• Se id cdl non esiste
<a href="#">insegnamentiDisponibili</a>	• idCdl (varchar(10))	Funzione che, dato l'id di un cdl, ritorna tutti gli insegnamenti che possono essere aggiunti a quel cdl (non ancora presenti).	• Il cdl inserito non esiste
<a href="#">cdlIsAttivo</a>	• idCdl (varchar(10))	Funzione che dato il codice di un corso di laurea ritorna se è attivo o meno.	• Il corso di laura inserito non esiste
<a href="#">infoIns</a>	• idInse (integer)	Funzione che dato l'id di un insegnamento ne ritorna le sue informazioni.	• L'insegnamento non esiste
<a href="#">anniPossibili</a>	• idCdl(varchar 10)	Funzione che dato l'id di un cdl ritorna gli anni possibili per gli esami.	
<a href="#">docentiDisponibili</a>	• idUtente(integer)	Funzione che dato l'id di un utente ritorna i docenti a cui può essere assegnato un insegnamento.	• Se non esiste nessun utente con tale id
<a href="#">propDisponibili</a>	• idCdl Varchar(10) • idInsegnamento (integer)	Funzione che, dato l'id di un insegnamento e il cdl di cui fa parte, restituisce tutti gli insegnamenti di quel corso per inserire le propedeuticità.	• Nessun corso con il codice specificato come parametro • Nessun insegnamento con il codice specificato come parametro
<a href="#">visualizzaTuttiInsegnamenti</a>	• idUtente (integer)	Funzione che, dato l'id di un utente (per verificare che sia un accesso autorizzato), ritorna tutti gli insegnamenti presenti.	• Se l'utente inserito non è autorizzato
<a href="#">idCdlPerInsegnamento</a>	• idInsegnamento integer	Funzione che dato l'id di un insegnamento ritorna i cdl di cui fa parte.	• Id non esistente
<a href="#">visualizzaStudentiCdl</a>	• idCdl	Funzione che dato l'id di un cdl ritorna tutti gli studenti iscritti.	• Non esiste nessun corso di laurea con il codice inserito
<a href="#">visualizzaInfoDisiscrizione</a>	• matricola studente	Funzione che data la matricola di uno studente ritorna le sue informazioni.	• La matricola inserita non esiste
<a href="#">idAssociatoMatricola</a>	• matricola (integer)	Funzione che dato l'id di una matricola restituisce l'id utente associato.	• La matricola inserita non esiste
<a href="#">appelliDatoUtente</a>	• idUtente integer	Funzione che restituisce tutti gli appelli, dato l'id di un utente.	• Utente non autorizzato
<a href="#">esitoStud</a>	• matricola intger • idAppello integer	Funzione che dato la matricola e l'appello di un esame a cui ha partecipato ritorna le sue informazioni.	• La matricola non ha partecipato all'appello inserito

<a href="#"><u>cambiacvotoaccettato</u></a>	• matricola (integer) • idAppello (integer) • votoEsame ("UniNostra".voto)	Procedura che data la matricola e l'id ad un appello a cui è iscritta permette di cambiare il voto.	• Matricola non iscritta all'appello • Matricola iscritta ma con stato != Accettato • Voto < 18
<a href="#"><u>studentiprontilaurea</u></a>	• idUtente	Funzione che ritorna gli studenti che si possono laureare.	• Utente non autorizzato
<a href="#"><u>mediaStudente</u></a>	• matricola (integer)	Funzione che data la matricola di uno studente ritorna la sua media.	• Studente non esistente
<a href="#"><u>tuttiigliexstudenti</u></a>	• idSegretario (integer)	Funzione che ritorna tutti gli ex-studenti.	• Utente non autorizzato
<a href="#"><u>tuttiidocenti</u></a>	• idSegretario (integer)	Funzione che, dato l'id di un segretario, restituisce tutti i docenti presenti nel database.	• Utente non autorizzato
<a href="#"><u>numexdocenze</u></a>	• idDocente (integer)	Funzione che dato l'id di un docente ritorna il numero di docenze.	• L'id inserito non appartiene ad un docente
<a href="#"><u>updateinfodocente</u></a>	• idDocente (integer) • indirizzo varchar(100) • cellulare varchar(10)	Procedura che permette di aggiornare le informazioni di un docente.	• Non esiste nessun docente con id inserito
<a href="#"><u>tuttiisegretari</u></a>	• idSegretario (integer)	Funzione che, dato l'id di un segretario, ritorna tutti i segretari presenti nella base di dati.	• Utente non autorizzato, l'id inserito non è di un segretario.
<a href="#"><u>updateinfosegretario</u></a>	• idSegretario (integer) • nome diartimento varchar(50) • indirizzo varchar(100) • cellulare varchar(10)	Procedura che permette di aggiornare le informazioni di un segretario.	• Non esiste nessun segretario con id inserito
<a href="#"><u>tuttiigliutenti</u></a>	• idSegretario	Funzione che permette di visualizzare tutti gli utenti tranne i segretari.	• Utente non autorizzato
<a href="#"><u>infoutente</u></a>	• idUtente	Funzione che dato l'id di un utente ritorna alcune sue informazioni.	• Utente inesistente
<a href="#"><u>recuperopsw</u></a>	• idSegretario (integer) • idUtente(integer) • nuovaPsw varchar(32)	Procedura che dato l'id di un utente e la nuova password modifica le sue credenziali d'accesso.	• Utente non autorizzato • Utente non esistente • L'utente a cui si vuole cambiare password è un segretario

## Prove di funzionamento

### - Schermata di login

Per prima cosa, un utente che vuole accedere all'applicazione web si ritroverà davanti ad una schermata di login.

Dovrà immettere alcune credenziali :

- Nome utente
- Password
- Tipo utente

Verrà visualizzato un errore in caso di credenziali errate. Se le credenziali sono corrette, in base al tipo di utenza sarà indirizzato al corretto pool di funzionalità.

### - Studente

Uno studente potrà eseguire le seguenti funzionalità :

- Visualizzare il proprio profilo
- Iscriversi agli appelli aperti del suo cdl
- Visualizzare e disiscriversi dagli appelli
- Accettare o rifiutare voti
- Visualizzare la carriera
- Visualizzare la carriera passata
- Visualizzare gli insegnamenti del proprio cdl e quelli di altri cdl
- Cambiare la propria password

The screenshot shows the UniNostra student dashboard with a blue header bar. On the left is the UniNostra logo. In the center, the user's name 'luca.corradin@studenti.UniNostra' is displayed, along with a 'Logout' button. Below the header is a grid of six cards:

- Profilo**: Visualizza profilo studente. Button: Visualizza.
- Iscriviti agli appelli**: Iiscrizione agli appelli del proprio cdl. Button: Visualizza.
- Visualizza iscrizioni**: Visualizza iscrizioni confermate agli appelli. Button: Visualizza.
- Valutazioni esami**: Accetta o rifiuta le valutazioni degli esami. Button: Visualizza.
- Carriera**: Visualizza la carriera e la propria media. Button: Visualizza.
- Storico Carriera**: Visualizza la carriera passata. Button: Visualizza.
- Insegnamenti Cdl**: Visualizza gli insegnamenti del tuo cdl. Button: Visualizza.
- Insegnamenti vari**: Visualizza gli insegnamenti di altri cdl. Button: Visualizza.
- Cambio Password**: Cambia la password dell'account. Button: Visualizza.

#### - Iscrizione agli appelli

L'iscrizione ad un particolare appello sarà bloccata nei seguenti casi :

- Lo studente ha già un voto accettato in carriera per quello specifico esame.
- Lo studente ha una valutazione pendente (né accettata, né rifiutata) per quello specifico esame.
- Lo studente non ha passato un esame propedeutico all'esame a cui si vuole iscrivere.
- L'appello risulta chiuso.
- Lo studente è già iscritto ad un altro turno dello stesso insegnamento tenuto durante la stessa giornata.

The screenshot shows the UniNostra registration page with a blue header bar. On the left is the UniNostra logo. In the center, the user's name 'giacomo.comitani@studenti.UniNostra' is displayed, along with a 'Logout' button. Below the header is a table showing one course entry:

#	Codice Insegnamento	Nome Insegnamento	Cfu	Docente	Data	Ora Inizio	Aula	Cdl	Stato	Iscritti
1	12	Medicina Generale	12	Violetta Lonati	22-09-2023	10:00:00	Gamma	MD101	aperto	<b>Iscritti</b>



In caso di iscrizione avvenuta correttamente, lo studente sarà in grado di visualizzare l'iscrizione nell'apposita pagina e se l'appello è aperto potrà disiscriversi.

The screenshot shows the UniNostra disinscription page with a blue header bar. On the left is the UniNostra logo. In the center, the user's name 'giacomo.comitani@studenti.UniNostra' is displayed, along with a 'Logout' button. Below the header is a table showing one course entry:

#	Codice Insegnamento	Nome Insegnamento	Cfu	Docente	Data	Ora Inizio	Aula	Stato Appello	Stato Studente	Disiscritti
1	12	Medicina Generale	12	MD101 Violetta Lonati	22-09-2023	10:00:00	Gamma	aperto	Iscritto	<b>Disiscritti</b>



#### - Accetta e rifiuta voto, carriera studente

Lo studente, inoltre, per gli appelli chiusi potrà accettare o rifiutare le valutazioni inserite dal docente responsabile dell'appello.

- Voto accettato := il voto verrà visualizzato tra gli esami passati e contribuirà alla media.
- Voto rifiutato := il voto sarà visibile nella carriera completa dello studente.

Quando uno studente accetta un voto, inoltre, sarà disiscritto da tutti gli altri appelli aperti dello stesso insegnamento a cui si era iscritto.


**UniNostra**
andrea.galliano@studenti.UniNostra
Logout

Codice # Insegnamento	Nome Insegnamento	Cfu	Cdl	Docente	Data	Voto	Stato Studente	Accetta	Rifiuta
1 4	Algoritmi e strutture dati	12	FX101	Giovanni Pighizzini	16-09-2023	30 e lode	In attesa	<span style="background-color: green; color: white; border-radius: 5px; padding: 2px 5px;">Accetta</span>	<span style="background-color: red; color: white; border-radius: 5px; padding: 2px 5px;">Rifiuta</span>



Via Festa del Perdono 7 - 20122 Milano | Assistenza Unimia

### Esami Passati

# Codice Insegnamento	Nome Insegnamento	Cfu	Cdl	Docente	Data	Voto	Stato Studente
1 4	Algoritmi e strutture dati	12	FX101	Giovanni Pighizzini	16-09-2023	30 e lode	Accettato
2 10	Programmazione 3	12	FX101	Giovanni Pighizzini	30-08-2023	30 e lode	Accettato
3 6	Programmazione 1	12	FX101	Fabio Scotti	04-08-2023	29	Accettato

Media: 29.67

### Carriera Studente

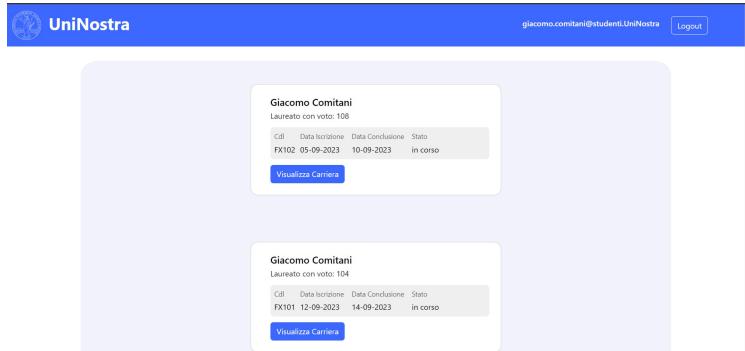
# Codice Insegnamento	Nome Insegnamento	Cfu	Cdl	Docente	Data	Voto	Stato
1 4	Algoritmi e strutture dati	12	FX101	Giovanni Pighizzini	16-09-2023	30 e lode	Accettato
2 4	Algoritmi e strutture dati	12	FX101	Giovanni Pighizzini	15-09-2023	23	Rifiutato
3 10	Programmazione 3	12	FX101	Giovanni Pighizzini	30-08-2023	30 e lode	Accettato
4 6	Programmazione 1	12	FX101	Fabio Scotti	04-08-2023	29	Accettato

#### - Storico Carriera

Se uno studente ha una carriera pregressa in un altro corso di laurea di UniNostra, potrà visualizzare alcune informazioni :

- Se ha conseguito la laurea e con quale voto
- Stato iscrizione (in corso o fuori corso)
- Anno di inizio studi e anno di fine studi

Inoltre, potrà visualizzare la sua carriera pregressa.



**Giacomo Comitani**  
Laureato con voto: 108

Cdl	Data Iscrizione	Data Conclusione	Stato
FX102	05-09-2023	10-09-2023	in corso

Visualizza Carriera

**Giacomo Comitani**  
Laureato con voto: 104

Cdl	Data Iscrizione	Data Conclusione	Stato
FX101	12-09-2023	14-09-2023	in corso

Visualizza Carriera

#### - Visualizzazione insegnamenti di altri corsi

Lo studente potrà visualizzare gli insegnamenti sia del suo cdl sia quelli di altri cdl.



**Selezione Cdl:**

Codice # Insegnamento	Nome Insegnamento	Cfu	Cdl	Anno Erogazione	Docente	Propedeuticità
1 13	Fisica 1	12	MD101	1	primo approccio alla fisica	Vincenzo Puri Nessuna
2 11	Anatomia	18	MD101	1	Si studia il corpo umano	Vincenzo Puri Nessuna
3 12	Medicina Generale	12	MD101	2	si studiano cose	Violetta Lonati Nessuna
4 32	Fisica 2	6	MD101	2	Esame di fisica 2	Vincenzo Puri Fisica 1

MD101 ▾

- FX101
- FX102
- MD101
- MD102

#### - Cambio Password

Lo studente, inoltre, come tutti gli utenti, è in grado di cambiare la password relativa al proprio account

The screenshot shows the 'Cambio Password' (Change Password) page. At the top, there is a success message: 'Cambio Password avvenuto con successo!'. The form fields include 'Password Attuale' (Current Password), 'Password Nuova' (New Password), and 'Ripeti la password' (Repeat Password). Below the form is a 'Invia' (Send) button. The footer contains the university logo, the address 'Via Festa del Perdono 7 - 20122 Milano - Assistenza Unimia', and a 'Logout' button.

#### - Docente

Un docente all'interno dell'applicativo web dispone delle seguenti funzionalità :

- Visionare il proprio profilo e gli insegnamenti di cui è responsabile nei vari cdl
- Creare appelli per gli insegnamenti di cui è responsabile
- Visualizzare gli appelli aperti e gli studenti iscritti
- Valutare gli studenti per gli appelli chiusi
- Visualizzare tutti gli appelli chiusi che ha tenuto con gli studenti che vi hanno partecipato, sia ex che attuali.
- Visualizzare gli insegnamenti tenuti in precedenza
- Cambiare la password.

The screenshot shows the teacher dashboard with several buttons: 'Profilo' (Profile), '+ Crea Appelli' (Create Appeals), 'Visualizza Appelli' (View Appeals), 'Immetti Valutazioni' (Enter Evaluations), 'Storico Appelli' (Appeal History), 'Storico Insegnamenti' (Teaching History), and 'Cambia Password' (Change Password). The 'Cambia Password' button is highlighted with a blue border.

#### - Creazione degli appelli e visualizzazione

Un docente può creare gli appelli per gli insegnamenti di cui è responsabile. La creazione di un appello verrà bloccata con un messaggio di errore se :

- Sono già presenti nella stessa giornata appelli di altri insegnamenti erogati nello stesso anno accademico per un certo cdl
- L'aula o le aule specificate dal docente risultano occupate da un altro appello
- Un docente può creare più turni dello stesso appello durante una giornata, tutta via essi non si devono sovrapporre per quanto riguarda l'orario.
- Se la data specificata è passata o con troppo poco preavviso, un appello deve essere creato almeno con un giorno di anticipo.

The screenshot shows the 'Appello inserito con successo' (Appeal inserted successfully) message at the top. Below it is a form for creating a new appeal. The form fields include: 'Codice Insegnamento' (Course Code) set to '12 - Medicina Generale', 'Cdl' (CDL) set to 'MD101', 'Aula/e Aula1+Aula2...' (Auditorium), 'Selezione Data' (Select Date) set to '21-09-2023', 'Ora Inizio: hh:mm:ss' (Start Time), 'Ora Fine: hh:mm:ss' (End Time), 'Descrizione' (Description), and a 'Crea' (Create) button.

Se l'appello è stato creato correttamente sarà possibile visionarlo assieme alla lista degli studenti partecipanti.

**Sarà inoltre possibile filtrare i risultati nella tabella , attraverso l'opportuna barra di ricerca**

## Appelli Aperti

Filtro 

Codice # Insegnamento	Nome Insegnamento	Cfu	Cdl	Anno Erogazione	Data Esame	Ora Inizio	Ora Fine	Aula	note	stato	iscritti	lista iscritti
1 12	Medicina Generale	12	MD101	2	22-09-2023	10:00:00	13:00:00	Gamma	Appello di medicina generale	aperto	1	<a href="#">Iscrizioni</a>



## Studenti Iscritti

Giacomo Comitani

Matricola 10011 Stato Iscritto Cdl MD101

**- Inserire valutazioni relative ad appelli chiusi**

Una volta che l'appello è chiuso (data passata), il docente responsabile potrà valutare gli studenti regolatamente iscritti.

- Se lo studente è stato promosso ( voto > 18), lo stato della sua iscrizione passerà al valore "In attesa". Egli potrà quindi accettare o rifiutare il voto
- Se il voto viene lasciato vuoto, lo studente viene segnato come ritirato
- Se il voto inserito è < 18, lo studente sarà bocciato

## Valutazioni in sospeso

Codice # Insegnamento	Nome Insegnamento	Clu	Cdl	Anno Erogazione	Data Esame	Ora Inizio	Ora Fine	Aula	note	stato	iscritti	lista iscritti
1 4	Algoritmi e strutture dati	12	FX101	2	15-09-2023	10:00:00	11:00:00	lambda+gamma	appello di algoritmi	chiuso	1	<a href="#">Iscrizioni</a>
2 4	Algoritmi e strutture dati	12	FX101	2	16-09-2023	15:00:00	17:00:00	omega+lambda	Appello Algoritmi	chiuso	1	<a href="#">Iscrizioni</a>



## Iscrizioni appello id: 85

Filtro 

Progressivo	Matricola	Nome	stato	cdf	valuta
1	3	Andrea Galliano	Iscritto	FX101	<a href="#">Valuta</a>



Voto Inserito con successo

Voto Inserito con successo

Matricola	Codice Appello
11	85
Voto, campo vuoto per studente ritirato	
Lode <input type="checkbox"/>	
<input type="button" value="Invia"/>	

Università degli studi di milano  Via Festa del Perdono 7 - 20122 Milano - Assistenza Unimia

#### - Storico Appelli

Il docente, potrà inoltre visionare tutti gli appelli che ha tenuto in passato, ciascuno con tutti gli studenti che vi hanno partecipato (sia attuali che ex -studenti).

 **UniNostra** giovanni.pighizzini@docenti.UniNostra

Filtra

#	Codice Insegnamento	Nome Insegnamento	Cfu	Cdl	Anno Erogazione	Data Esame	Ora Inizio	Ora Fine	Aula	note	stato	iscritti	lista iscritti
1	10	Programmazione 3	12	FX101	1	30-08-2023	11:40:00	13:00:00	omega	bho	chiuso	2	<input type="button" value="Iscrizioni"/>
2	10	Programmazione 3	12	FX102	1	30-08-2023	10:00:00	11:00:00	omega	bho	chiuso	1	<input type="button" value="Iscrizioni"/>
3	10	Programmazione 3	12	FX102	1	31-08-2023	10:50:00	15:15:00	omega	bho	chiuso	0	Nessun iscritto
4	4	Algoritmi e strutture dati	12	FX102	2	31-08-2023	10:00:00	11:00:00	omega	bho	chiuso	1	<input type="button" value="Iscrizioni"/>
5	10	Programmazione 3	12	FX101	1	01-09-2023	08:00:00	11:00:00	gamma+lambda	bho	chiuso	0	Nessun iscritto
6	4	Algoritmi e strutture dati	12	FX101	2	03-09-2023	12:00:00	15:00:00	gamma+lambda	bho	chiuso	1	<input type="button" value="Iscrizioni"/>
7	10	Programmazione 3	12	FX101	1	04-09-2023	16:00:00	18:00:00	gamma+lambda	bho	chiuso	0	Nessun iscritto

 **UniNostra** giovanni.pighizzini@docenti.UniNostra

**Partecipanti appello id: 11**

Filtra

#	Matricola	Nome	Cdl	voto	stato
1	1	Luca Corradini	FX101	20	Accettato
2	3	Andrea Galliano	FX101	30 e lode	Accettato

Università degli studi di milano  Via Festa del Perdono 7 - 20122 Milano - Assistenza Unimia

#### - Profilo Docente

Il docente può visualizzare alcune informazioni biografiche e gli insegnamenti che tiene nei vari cdl.

Insegnante Docente	Regolamenti e strutture dati Indirizzo Ufficio Via celoria 18 piano 8 393020220
--------------------	--

**Insegnamenti di cui è responsabile**

Selezione Cdl:

#	Codice Insegnamento	Nome Insegnamento	Cfu	Cdl	Anno Erogazione	Descrizione	Docente	Propedeuticità
1	10	Programmazione 3	12	FX101	1	non esiste	Giovanni Pighizzini	Nessuna
2	4	Algoritmi e strutture dati	12	FX101	2	Esame che riguarda pighizzini	Giovanni Pighizzini	Programmazione 1

#### - Segreteria

Un segretario, dato che è l'autorità di massimo livello , potrà compiere il maggior numero di operazioni sulla base di dati. In particolare potrà :

- Visualizzare il proprio profilo, con alcune informazioni personali
- Visualizzare i cdl esistenti con i relativi piani di studi .
- Creare nuovi corsi di laurea
- Visualizzare, modificare e aggiungere nuovi insegnamenti
- Visualizzare gli studenti per corso di laurea, modificare le loro informazioni e aggiungerne di nuovi
- Visualizzare appelli aperti e chiusi
- Laureare studenti
- Visualizzare ex-studenti
- Visualizzare, modificare e aggiungere nuovi docenti
- Visualizza, modifica e aggiungere nuovi segretari
- Recuperare le password di altri utenti

#### - Visualizza e modifica piani studi dei cdl

Un segretario per ogni corso di laurea, potrà :

- Aggiungere o rimuovere insegnamenti da un piano di studi.  
Un insegnamento non può essere rimosso da un manifesto degli studi se esistono degli appelli ancora aperti o il docente deve assegnare delle valutazioni in sospeso. Inoltre, quando un esame viene rimosso dal piano di studi tutte le propedeuticità da e per altri insegnamenti di quel cdl verranno eliminate.  
Uno studente che ha passato un esame rimosso da un piano di studi non lo visualizzerà tra gli esami che fanno media, ma nella sua carriera.
- Aggiungere o rimuovere propedeuticità. Quando un esame viene reso propedeutico entra in vigore solamente per coloro che non hanno ancora passato quello stesso esame.

Un corso di laurea inattivo (senza insegnamenti nel suo piano di studi) passerà ad essere attivo quando verrà aggiunto almeno un insegnamento al suo piano di studi.

#	Codice Cdl	Nome	Descrizione	Durata	Attivo	Numero Iscritti	Piano Studi
1	MD102	Medicina	Magistrale in medicina	Magistrale	Non attivo	0	<button>Visualizza</button>
2	FX101	Informatica	Corso di laurea triennale in informatica	Triennale	Attivo	2	<button>Visualizza</button>
3	MD101	Medicina	triennale di medicina	Triennale	Attivo	2	<button>Visualizza</button>
4	FX102	Informatica	Corso di laurea Magistrale in informatica	Magistrale	Attivo	0	<button>Visualizza</button>



## Piano Studi Cdl : FX101

Filtro:

Ricerca...

Aggiungi Insegnamento al piano di studi: [Aggiungi](#)

Codice # Insegnamento	Nome Insegnamento	Cfu	CdL	Anno Erogazione	Descrizione	Docente	Propedeuticità	Rimuovi dal manifesto	Modifica Propedeuticità
1 7	Programmazione 2	12	FX101	1	pighizzini	Fabio Scotti	Nessuna	<a href="#">Rimuovi</a>	<a href="#">Modifica</a>
2 6	Programmazione 1	12	FX101	1	pighizzini	Fabio Scotti	Nessuna	<a href="#">Rimuovi</a>	<a href="#">Modifica</a>
3 10	Programmazione 3	12	FX101	1	non esiste	Giovanni Pighizzini	Nessuna	<a href="#">Rimuovi</a>	<a href="#">Modifica</a>
4 4	Algoritmi e strutture dati	12	FX101	2	Esame che riguarda pighizzini	Giovanni Pighizzini	Programmazione 1	<a href="#">Rimuovi</a>	<a href="#">Modifica</a>



Esame inserito con successo

Insegnamento	Anno Corso
12 - Medicina Generale	1
Cfu	Docente
12	Violetta Lonati
<a href="#">Invia</a>	



## Aggiungi propedeuticità per l'insegnamento : 6

Codice Insegnamento	Nome Insegnamento
6	Programmazione 1
Propedeuticità Attuali :	
<b>Nessuna</b>	
Propedeuticità disponibili	
10 - Programmazione 3 - Anno erogazione: 1	
<a href="#">Aggiungi</a>	<a href="#">Rimuovi</a>
<a href="#">Torna a visualizza insegnamenti:</a> <a href="#">Indietro</a>	

**- Visualizza e modifica piani studi dei cdl**

Un segretario può inserire un nuovo corso di laurea. Inizialmente il corso di laurea sarà inattivo (in quanto senza insegnamenti nel piano studio), di conseguenza non si potranno iscrivere nuovi studenti.

La creazione del corso di laurea darà errore se :

- Esiste già un cdl con lo stesso codice
- Esiste già un cdl con lo stesso nome e la stessa durata

**UniNostra**

luigipepe@segreteria.UniNostra Logout

Corso di laurea inserito!

### Crea Corso di Laurea

Codice Corso di laurea, massimo 10 caratteri	Nome Corso di laurea
Descrizione:	
Durata CdL	
Triennale	
<a href="#">Aggiungi</a>	

Università degli studi di milano  Via Festa del Perdono 7 - 20122 Milano Assistenza Unimia

**UniNostra**

luigipepe@segreteria.UniNostra Logout

### Corsi di laurea

Filtra Ricerca...

#	Codice CdL	Nome	Descrizione	Durata	Attivo	Numero Iscritti	Piano Studi
1	MD102	Medicina	Magistrale in medicina	Magistrale	Non attivo	0	<a href="#">Visualizza</a>
2	FX101	Informatica	Corso di laurea triennale in informatica	Triennale	Attivo	2	<a href="#">Visualizza</a>
3	CO202	chimica organica	chimica organica	Triennale	Non attivo	0	<a href="#">Visualizza</a>
4	MD101	Medicina	triennale di medicina	Triennale	Attivo	2	<a href="#">Visualizza</a>
5	FX102	Informatica	Corso di laurea Magistrale in informatica	Magistrale	Attivo	0	<a href="#">Visualizza</a>

Università degli studi di milano  Via Festa del Perdono 7 - 20122 Milano Assistenza Unimia

#### - Aggiunta di nuovi insegnamenti

Il segretario potrà aggiungere nuovi insegnamenti. All'atto della creazione sarà obbligatorio inserire il responsabile del corso.  
Al segretario verranno mostrati solamente i docenti disponibili, ovvero coloro che sono responsabili di meno di 3 insegnamenti.

La creazione darà errore se :

- Esiste già un insegnamento con lo stesso nome e gli stessi cfu
- Il docente selezionato è già responsabile di quel corso.

Sarà inoltre possibile modificare alcune informazioni relative all'insegnamento.

**UniNostra**

luigipepe@segreteria.UniNostra Logout

### Insegnamenti

Filtra Ricerca...

Aggiungi Insegnamento: [Aggiungi](#)

#	Id Insegnamento	Nome	Descrizione	Cfu	Docente	Cdl	Modifica
1	4	Algoritmi e strutture dati	Esame che riguarda piglizzini	12	Giovanni Piglizzini	FX101 FX102	<a href="#">Modifica</a>
2	6	Programmazione 1	pighizzini	12	Fabio Scotti	FX101 FX102	<a href="#">Modifica</a>
3	7	Programmazione 2	pighizzini	12	Fabio Scotti	FX102	<a href="#">Modifica</a>
4	10	Programmazione 3	non esiste	12	Giovanni Piglizzini	FX101 FX102	<a href="#">Modifica</a>
5	11	Anatomia	Si studia il corpo umano	18	Vincenzo Piuri	MD101	<a href="#">Modifica</a>
6	12	Medicina Generale	si studiano cose	12	Violetta Lonati	MD101	<a href="#">Modifica</a>
7	13	Fisica 1	primo approccio alla fisica	12	Vincenzo Piuri	MD101	<a href="#">Modifica</a>
8	32	Fisica 2	Esame di fisica 2	6	Vincenzo Piuri	MD101	<a href="#">Modifica</a>

**UniNostra**

luigipepe@segreteria.UniNostra Logout

Insegnamento inserito!

### Crea Insegnamento

Nome Insegnamento	Cfu
anatomia	9
Descrizione:	
anatomia	
Docente	
Fabio Scotti	
<a href="#">Aggiungi</a>	

Università degli studi di milano  Via Festa del Perdono 7 - 20122 Milano Assistenza Unimia

**Insegnamenti**

Filtra Ricerca... Aggiungi Insegnamento: [Aggiungi](#) Logout

#	Id Insegnamento	Nome	Descrizione	Cfu	Docente	Cdl	Modifica
1	4	Algoritmi e strutture dati	Esame che riguarda pighizzini	12	Giovanni Pighizzini	FX101 FX102	<a href="#">Modifica</a>
2	6	Programmazione 1	pighizzini	12	Fabio Scotti	FX101 FX102	<a href="#">Modifica</a>
3	7	Programmazione 2	pighizzini	12	Fabio Scotti	FX102	<a href="#">Modifica</a>
4	10	Programmazione 3	non esiste	12	Giovanni Pighizzini	FX101 FX102	<a href="#">Modifica</a>
5	11	Anatomia	Si studia il corpo umano	18	Vincenzo Piuri	MD101	<a href="#">Modifica</a>
6	12	Medicina Generale	si studiano cose	12	Violetta Lonati	MD101	<a href="#">Modifica</a>
7	13	Fisica 1	primo approccio alla fisica	12	Vincenzo Piuri	MD101	<a href="#">Modifica</a>
8	32	Fisica 2	Esame di fisica 2	6	Vincenzo Piuri	MD101	<a href="#">Modifica</a>
9	33	anatomia	anatomia	9	Giovanni Pighizzini	Nessun Cdl	<a href="#">Modifica</a>

- [Aggiunta di nuovi studenti](#)

Il segretario, inoltre potrà aggiungere nuovi studenti ai cdl attivi :

- Può essere registrato anche un ex studente che non abbia una laurea per il cdl a cui si vuole iscrivere
- La registrazione darà errore se esiste un username nome.cognome con ruolo diverso da studente
- Se esiste già uno studente registrato con quella email

Il segretario può anche disiscrivere gli studenti, quando ciò accade lo studente andrà nello storico.

Possono inoltre essere modificate alcune informazioni personali dello studente.

**Corsi di laurea**

Filtra Ricerca... luigipepe@segreteria.UniNostra Logout

#	Codice Cdl	Nome	Descrizione	Durata	Attivo	Numero Iscritti	Piano Studi
1	MD102	Medicina	Magistrale in medicina	Magistrale	Non attivo	0	<a href="#">Visualizza</a>
2	FX101	Informatica	Corso di laurea triennale in informatica	Triennale	Attivo	2	<a href="#">Visualizza</a>
3	CO202	chimica organica	chimica organica	Triennale	Non attivo	0	<a href="#">Visualizza</a>
4	MD101	Medicina	triennale di medicina	Triennale	Attivo	2	<a href="#">Visualizza</a>
5	FX102	Informatica	Corso di laurea Magistrale in informatica	Magistrale	Attivo	0	<a href="#">Visualizza</a>

Università degli studi di milano  Via Festa del Perdono 7 - 20122 Milano Assistenza Unimia

**Studenti per il cdl : FX101**

Filtra Ricerca... Aggiungi Studente: [Aggiungi](#) luigipepe@segreteria.UniNostra Logout

#	Matricola	Nome	Telefono	Data di Nascita	Residenza	Data Iscrizione	Stato	Modifica	Carriera	Discrim
1	1	Luca Corradini	222222	05-08-2002	Busto Arsizio Va	08-05-2019	Fuori Corso	<a href="#">Modifica</a>	<a href="#">Carriera</a>	<a href="#">Discriva</a>
2	11	Andrea Galliano	3930582002	03-11-2002	gallarabia mi	15-09-2023	In Corso	<a href="#">Modifica</a>	<a href="#">Carriera</a>	<a href="#">Discriva</a>

Università degli studi di milano  Via Festa del Perdono 7 - 20122 Milano Assistenza Unimia

**Aggiungi Studente al cdl : FX101**

Studente iscritto con successo!

Nome:	francesco	Cognome:	capuzzo
User: nome.cognome	francesco.capuzzo	cf: max 16 caratteri	CPZ
Password: minimo 4 caratteri	....	Conferma Password:	....
Telefono:	3481749850	Residenza:	Busto Arsizio

## Studenti per il cdl : FX101

Filtra Ricerca... [Aggiungi Studente](#) [Aggiungi](#)

#	Matricola	Nome	Telefono	Data di Nascita	Residenza	Data Iscrizione	Stato	Modifica	Carriera	Discrim.
1	1	Luca Corradini	222222	05-08-2002	Busto Arsizio Va	08-05-2019	Fuori Corso	<a href="#">Modifica</a>	<a href="#">Carriera</a>	<a href="#">Discrim.</a>
2	24	francesco capuzzo	3481749850	06-06-2000	Busto Arsizio	21-09-2023	In Corso	<a href="#">Modifica</a>	<a href="#">Carriera</a>	<a href="#">Discrim.</a>
3	11	Andrea Galliano	3930582002	03-11-2002	gallarabia mi	15-09-2023	In Corso	<a href="#">Modifica</a>	<a href="#">Carriera</a>	<a href="#">Discrim.</a>

Università degli studi di milano



Via Festa del Perdono 7 - 20122 Milano Assistenza Unimia

- [Visualizza appelli](#)

Un segretario può anche visionare tutti gli appelli, passati e presenti, per ogni corso di laurea.

Gli appelli comprendono la lista degli studenti che vi hanno partecipato, sia attuali che ex studenti.

Per gli studenti attualmente iscritti ad un cdl che hanno accettato un esito per un certo esame sarà possibile modificarne il voto .

Filtra Ricerca...

#	Codice Insegnamento	Nome Insegnamento	Cfu	Cdl	Anno Erogazione	Data Esame	Ora Inizio	Ora Fine	Aula	note	stato	iscritti	lista iscritti
1	4	Algoritmi e strutture dati	12	Giovanni Pighizzini	16-09-2023	15:00:00	17:00:00	omega+lambda	FX101	chiuso	1	<a href="#">Iscrizioni</a>	
2	4	Algoritmi e strutture dati	12	Giovanni Pighizzini	15-09-2023	18:00:00	19:00:00	sigma	FX101	chiuso	0	Nessun iscritto	
3	4	Algoritmi e strutture dati	12	Giovanni Pighizzini	15-09-2023	10:00:00	11:00:00	lambda+gamma	FX101	chiuso	2	<a href="#">Iscrizioni</a>	
4	4	Algoritmi e strutture dati	12	Giovanni Pighizzini	13-09-2023	18:00:00	19:30:00	omega	FX101	chiuso	1	<a href="#">Iscrizioni</a>	
5	4	Algoritmi e strutture dati	12	Giovanni Pighizzini	12-09-2023	18:00:00	19:50:00	omega	FX101	chiuso	0	Nessun iscritto	
6	4	Algoritmi e strutture dati	12	Giovanni Pighizzini	12-09-2023	18:00:00	19:50:00	bertoni	FX101	chiuso	1	<a href="#">Iscrizioni</a>	
7	4	Algoritmi e strutture dati	12	Giovanni Pighizzini	12-09-2023	18:00:00	19:50:00	lambda	FX101	chiuso	0	Nessun iscritto	

Filtra Ricerca...

Partecipanti appello id: 85

#	Matricola	Nome	Cdl	voto	stato	modifica voto
1	11	Andrea Galliano	FX101	30 e lode	Accettato	<a href="#">Modifica</a>

Università degli studi di milano



Via Festa del Perdono 7 - 20122 Milano Assistenza Unimia

- [Laurea Studenti](#)

Il segretario, per ogni corso, potrà anche visualizzare gli studenti che si possono laureare. Uno studente si può laureare se ha passato tutti gli esami previsti dal cdl del suo corso di studi. Il segretario potrà inserire il voto di laurea, il quale in base alla media dello studente sarà compreso in un range possibile. Ad esempio se lo studente ha una media del 18, il voto di laurea potrà essere compreso tra 60 e 70.

Selezione Cdl:										
#	Matricola	Nome	Residenza	Data di nascita	Anno iscr.	Nome Cdl	Id Cdl			
1	11	Andrea Galliano	gallarabia mi	2002-11-03	15-09-20	FX101 CO102 FX101 FX102 MD101 MD102	Informatica	FX101	29	Laurea

- [Visualizza e aggiungi docenti](#)

Un segretario potrà anche aggiungere nuovi docenti. All'atto della creazione il docente può essere creato anche senza assegnargli degli insegnamenti. La procedura di aggiunta di un nuovo docente darà errore se all'interno della base di dati esiste già un utente con lo stesso username, oppure un docente con la stessa email. Sarà inoltre possibile :

- Modificare alcune informazioni personali di un docente
- Visualizzare lo storico dei suoi insegnamenti

Docenti							
Filtra	Ricerca...	Aggiungi Docente: <a href="#">Aggiungi</a>					
#	Nome	Indirizzo Ufficio	Cellulare interno	Numero Docenze	Docenze	Ex Docenze	Modifica
1	Vincenzo Piscitelli	via celoria 18	3331888772	3	Anatomia, Fisica 1, Fisica 2	Nessun ex insegnamento	<a href="#">Modifica</a>
2	Violetta Lonati	via celoria 18	3331888772	1	Medicina Generale	<a href="#">Visualizza</a>	<a href="#">Modifica</a>
3	Giovanni Pighizzini	Via celoria 18 piano 8	393020220	3	Programmazione 3, Algoritmi e strutture dati, anatomia	<a href="#">Visualizza</a>	<a href="#">Modifica</a>
4	Fabio Scotti	Via celoria	393020220	2	Programmazione 1, Programmazione 2	<a href="#">Visualizza</a>	<a href="#">Modifica</a>
5	Masimo Santini	Celoria 18	2838383	0	-	Nessun ex insegnamento	<a href="#">Modifica</a>

Storico Insegnamenti						
#	Codice Insegnamento	Nome Insegnamento	Cfu	Anno Inizio	Anno Fine	Docente
1	13	Fisica 1	12	2023	2023	Violetta Lonati
2	32	Fisica 2	6	2023	2023	Violetta Lonati



- [Visualizza e aggiungi docenti](#)

Un segretario potrà visualizzare gli ex-studenti per ogni cdl e per ognuno di essi visualizzare la sua carriera passata

## Ex-Studenti

Filtra

#	Matricola	Nome	Telefono	Residenza	Data nascita	Data iscrizione	Data rimozione	Stato	Stato	Voto laurea	Codice cdl	Nome Cdl	Carriera
1	4	Giacomo Comitani	3930582002	vimodrone mi	03-11-2002	12-09-2023	14-09-2023	In corso	Laureato	104	FX101	Informatica	<a href="#">Visualizza</a>
2	23	francesco capuzzo	3481749850	Busto Arsizio	06-06-2000	21-09-2023	21-09-2023	In corso	Ritirato	-	FX101	Informatica	<a href="#">Visualizza</a>
3	2	Giacomo Comitani	3930582002	vimodrone mi	03-11-2002	05-09-2023	10-09-2023	In corso	Laureato	108	FX102	Informatica	<a href="#">Visualizza</a>
4	17	Mattia DelleDonne	272727	Segrate, Varese	18-02-1999	19-09-2023	19-09-2023	In corso	Laureato	60	MD101	Medicina	<a href="#">Visualizza</a>
5	14	Mattia DelleDonne	272727	Busto	19-09-2023	19-09-2023	19-09-2023	In corso	Ritirato	-	MD101	Medicina	<a href="#">Visualizza</a>
6	15	Mattia DelleDonne	9999999	Milano.segrate	19-09-2023	19-09-2023	19-09-2023	In corso	Ritirato	-	MD101	Medicina	<a href="#">Visualizza</a>

### - Aggiungere nuovi segretati

Infine un segretario può aggiungere nuovi account di segretari, tale procedura fallirà se :

- Esiste già un username nome.cognome all'interno della base di dati
- Esiste già un segretario con l'email inserita.

## Segretari

Filtra

[Aggiungi Segretario](#) [Aggiungi](#)

#	Nome	Email	Indirizzo	Dipartimento	Cellulare	Modifica
1	Luigi Pepe	luigipepe@segreteria.UniNostra	Via celoria 8	Informatica	1234567891	<a href="#">Modifica</a>
2	Francesco Rossi	francesco.rossi@segreteria.UniNostra	Via celoria 18	Fisica	328229	<a href="#">Modifica</a>



### - EX - Studente

Un ex-studente, avrà meno funzionalità rispetto ad uno studente normale. Esso infatti potrà :

- Visualizzare il proprio profilo
- Visualizzare la sua carriera passata
- Modificare la propria password.

<b>Profilo</b> Visualizza profilo studente <a href="#">Visualizza</a>	<b>Visualizza Carriera</b> Visualizza carriera passata <a href="#">Visualizza</a>	<b>Cambia Password</b> Cambia la password del tuo account <a href="#">Visualizza</a>
---	---	--



## Profilo Utente



Mattia  
DelleDonne  
Ritratto

Informazioni	
Matricola	Corso di laurea
1101	MD101
Stato	Data Iscrizione
in corso	19-09-2023
Indirizzo	Telefono
Milano, Segrate	3838383

