

---

# 02NPYPD - LINGUAGGI E AMBIENTI MULTIMEDIALI A

## SIGN UP

*Registrazione degli utenti al sito web*

Lo scopo di questa esercitazione è quello di realizzare le pagine web necessarie per la registrazione di un utente al nostro social network, grazie al modello implementato durante la scorsa esercitazione. Prerequisito per questa esercitazione è il set di slide “*Lab 4 - Warm Up*”.

### WARM UP

Per iniziare, generiamo un controller chiamato `Users` con una vista denominata `new` e adeguiamola al contenuto e allo stile delle altre pagine. Inoltre, creiamo una `named route` in modo tale che la pagina `new` sia raggiungibile come `/signup` (es. `http://localhost:3000/signup`). Aggiorniamo, poi, il link presente nella homepage in modo che, cliccando sul bottone, si raggiunga la pagina appena creata.

Dopo aver seguito quanto mostrato in aula sulle slide denominate “*Lab 4 - Warm Up*”, aggiungiamo una nuova route per gestire le risorse messe a disposizione dal modello degli utenti (creato nella scorsa esercitazione) e dal suo controllore:

```
resources :users
```

Come mostrato nelle slide, tale route aggiunge tutta una serie di `named route` per generare URL legati agli utenti. Per iniziare a “vedere” gli effetti dell’aggiunta di tale route, creiamo un utente (agendo direttamente sul database o generandolo via codice) e aggiungiamo la pagina `show.html.erb` tra le viste associate al controllore degli utenti con la relativa azione nel controllore corrispondente, per mostrare alcuni dati dell’unico utente presente nel sito.

Questa parte di esercitazione è resa disponibile come progetto di partenza sul repository GitHub. Nel README sono indicate le modifiche rispetto alla soluzione della scorsa esercitazione.

### ESERCIZIO 1

Nel progetto di partenza trovate un file denominato `user_helper.rb` che contiene al suo interno l’helper `gravatar_for`. Tale helper non è completo; provvedete a completarlo, sapendo che:

- Gravatar è un servizio gratuito che permette agli utenti di caricare immagini e di associarle con gli indirizzi mail da loro controllati;
- gli ID dei Gravatar sono basati sulla codifica hash MD5 dell’indirizzo mail dell’utente, tutto minuscolo, e in Ruby l’algoritmo di hashing MD5 viene implementato dal metodo `Digest::MD5::hexdigest(stringa_da_utilizzare)`;
- l’helper ritorna un `image_tag` che è il modo Rails di implementare il tag HTML `<img>`;
- la pagina `show.html.erb` contiene già al suo interno la chiamata all’helper, in una riga commentata... rimuovete il commento prima di provare il funzionamento dell’helper!

## ESERCIZIO 2

Possiamo passare, ora, alla pagina di *sign up*. Se volete, potete cancellare l'utente presente sul database, utilizzando il comando `db:reset` di `rake` e rimuovere il codice per crearlo dall'azione `new` del controllore degli utenti.

Aiutandovi con le slide e utilizzando l'helper di Rails `form_for`<sup>1</sup> create, ora, il form necessario alla registrazione di un utente nel sito web, dando alla pagina uno stile coerente con il resto del sito e ricordandovi di inizializzare la variabile `@user` all'interno dell'azione `new` nel controllore degli utenti.

Dopo aver creato correttamente il form, visualizzate il codice HTML della pagina e guardate cosa Rails aggiunge in automatico. Valutare se tutto ciò che inserisce è necessario e cercare di capire, quando possibile, da dove Rails deriva le informazioni utilizzate per far funzionare correttamente il tutto.

## ESERCIZIO 3

Se, dopo aver realizzato il form di registrazione, provate a creare un nuovo utente premendo sul pulsante per inviare il form, otterrete un errore: l'azione `create` non esiste nel controllore `UserController`. Questo perché, facendo riferimento alla tabella delle route riportata nelle slide, è l'azione `create` che crea un nuovo utente (anche nel database).

Definite tale azione nel controllore degli utenti in modo da gestire sia form compilati in maniera corretta, sia form compilati in maniera scorretta (inserendo una password troppo corta, per esempio). L'azione `create`, pertanto, dovrà:

- creare un nuovo utente utilizzando tutti i dati passati nel form e memorizzando il tutto nella variabile `@user`
- se il salvataggio dei dati dell'utente nel DB va a buon fine, rimandare alla pagina del profilo utente (quella che risponde all'azione `show`, ad esempio `/users/1`)
- altrimenti richiamare l'azione `new` (cioè tornare alla pagina di registrazione).

## ESERCIZIO 4 (FACOLTATIVO)

Se la registrazione non va a buon fine, visualizzate dei messaggi di errore all'interno della pagina di registrazione. In questo caso, i messaggi di errore scatenati da Rails si possono recuperare utilizzando il metodo `user.errors.full_messages`. È consigliabile mettere tale metodo in un *partial* in una cartella chiamata `shared`, dato che il *partial* potrebbe anche essere utilizzato in form appartenenti a diversi controllori.

Aggiungete poi, un messaggio di benvenuto quando la registrazione dell'utente va a buon fine, utilizzando la variabile speciale (una hash, in realtà) messa a disposizione da Rails e chiamata `flash`<sup>2</sup>.

---

<sup>1</sup> <http://api.rubyonrails.org/classes/ActionView/Helpers/FormHelper.html>

<sup>2</sup> <http://api.rubyonrails.org/classes/ActionDispatch/Flash.html>