



UNIVERSITÀ  
DEGLI STUDI  
DI BERGAMO

# Web app Agile

*Web app per la gestione di progetti  
sviluppati con metodologia agile Scrum*

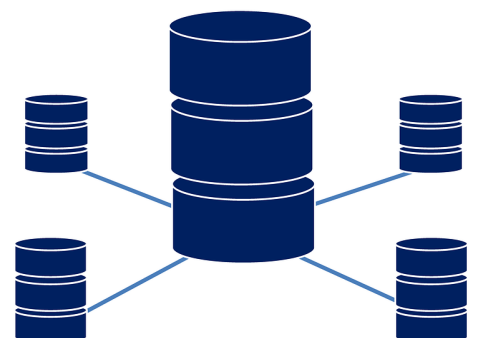
Basi di dati e Web - 9 CFU  
Anno Accademico 2021 ÷ 2022

## Indice

<b>1. Introduzione</b>	<b>2</b>
<b>2. Requisiti dell'applicazione</b>	<b>2</b>
<b>3. Modello ER</b>	<b>3</b>
<b>4. Schema logico</b>	<b>5</b>
<b>5. Popolazione della base di dati</b>	<b>5</b>
<b>6. Stampa delle pagine web</b>	<b>6</b>
Pagina Sign In / Sign Up	6
Pagina Active Sprint	6
Pagina Backlog	7
Pagina Epiche	7
Pagina Roadmap	8
Pagina Timesheet	8
Pagina Profilo	9

## Gruppo

- *De Duro Federico*    Mat. 1073477
- *Soldini Matteo*     Mat. 1074601
- *Zanotti Paolo*       Mat. 1074166



## 1. Introduzione

La web app sviluppata include il software di base per la gestione di progetti sviluppati con metodologie agili, la quale controlla in modo efficace tutti gli aspetti necessari per una buona gestione del ciclo di sviluppo del software, iterativo ed incrementale, concepito per gestire progetti e prodotti software o applicazioni di sviluppo. L'app si basa principalmente sul framework Scrum, ideato per la prima volta da Ken Schwaber e Jeff Sutherland a partire dalla fine degli anni '90. In maniera molto sintetica, Scrum è un framework di processo che prevede di dividere il progetto in blocchi rapidi di lavoro (Sprint), alla fine di ciascuno dei quali creare un incremento del software. Esso indica come definire i dettagli del lavoro da fare nell'immediato futuro e prevede vari meeting con caratteristiche precise per creare occasioni di ispezione e controllo del lavoro svolto. Lo sprint viene costruito a partire dalle Task presenti all'interno del Backlog (elenco di attività da svolgere al fine di migliorare il prodotto). Ogni task (Nota anche come User Story) può far parte di un Epica (macro-tema, come ad esempio "Documentazione", "Nuovo prodotto"... ) e può essere associata a diverse sottoattività. Esse vengono pesate in "punti fibonacci", corrispondenti a dieci ore di lavoro (esempio: Attività pesata 2 p.f., sarà eseguita in circa 20 ore di elaborazione). Ogni task può essere assegnata ad uno sviluppatore.

Si definiscono quindi diversi ruoli all'interno del team:

- Sviluppatori: coloro che fanno parte del team e lavorano in modo costante all'interno di ogni sprint (Sviluppatori software, più in generale i dipendenti).
- Esterni: coloro che segnalano le attività (Product Owner, Project manager...)
- Scrum master: responsabile della rimozione degli ostacoli che limitano la capacità del team di raggiungere l'obiettivo dello Sprint e i deliverable previsti.

## 2. Requisiti dell'applicazione

La web app è stata sviluppata utilizzando il framework Flask (basato su Python), mentre la struttura delle pagine dinamiche è stata scritta con HTML, JavaScript e **Jinja2**.

Per quanto riguarda lo stile abbiamo optato per il framework **TailwindCSS**, unito alla sua libreria **DaisyUI** per migliorare la UI del progetto.

Abbiamo incluso anche la libreria **jQuery UI** per gestire alcune animazioni come la funzione di Drag & Drop all'interno dei task.

Per gestire i Task ed i Subtask abbiamo utilizzato **AJAX**, questo ci ha permesso di usare delle chiamate asincrone e di conseguenza di aggiornare lo stato degli elementi nel database evitando il ricaricamento della pagina stessa.

La base di dati invece è stata gestita tramite **SQLAlchemy**, un toolkit SQL che ci ha permesso di gestire in maniera semplice ma efficace il tutto. In più questo ci ha permesso di evitare grossi problemi di sicurezza legati a delle nostre possibili sviste durante la fase di sviluppo.

E' stata implementata anche la funzione di autenticazione per gli utenti tramite il plugin **Flask-Login**, usando come supporto la [guida](#) scritta da DigitalOcean. Tutte le password vengono crittografate tramite l'algoritmo SHA-256 in modo da preservare la sicurezza degli utenti.

Per quanto riguarda la gestione delle immagini profilo abbiamo scelto di caricarle direttamente sul server e di tenere traccia nel database solo del nome del file.

Tutte le dipendenze dell'applicazione si trovano all'interno del file "*requirements.txt*", una volta aperto il progetto infatti basterà eseguire il comando "*pip3 install -r requirements.txt*" per installarle.

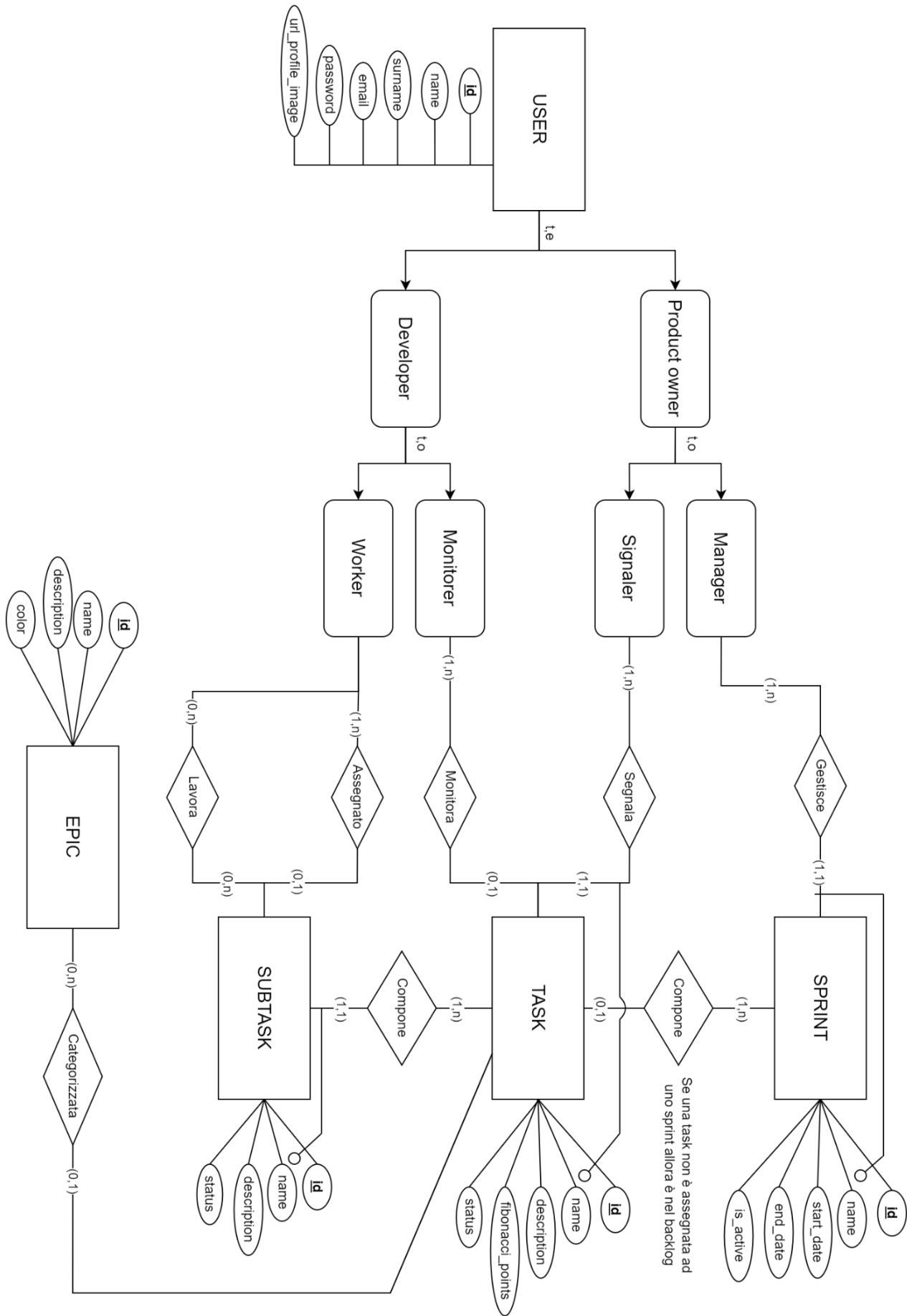
N.B. E' consigliato usare un ambiente virtuale in cui far girare il software.

Durante lo sviluppo dell'applicazione è stato usato un sistema di controllo di versione quale Git, in particolare tramite la piattaforma GitHub. Il repository è pubblico ed è stato rilasciato sotto licenza MIT.

 [Repository Github](#)

### 3. Modello ER

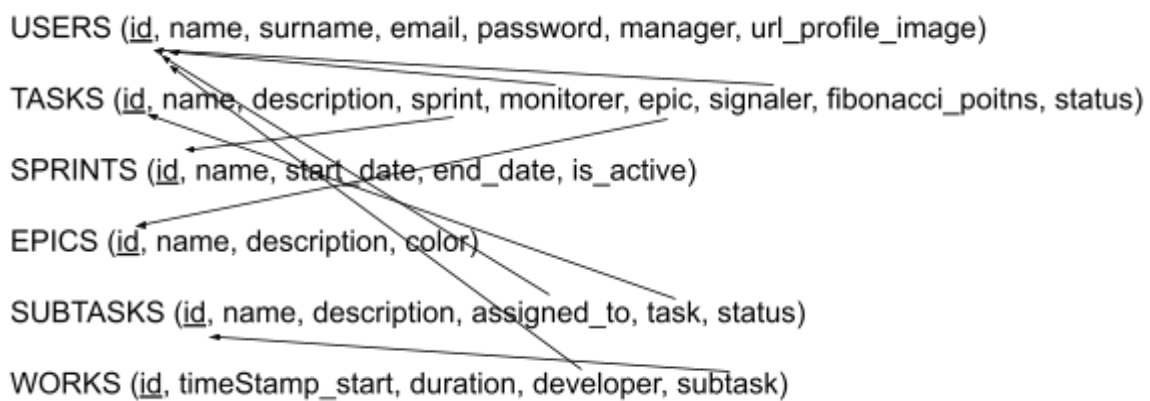
Come prima attività, è stato sviluppato il modello entità-relazione, sfruttando la strategia a macchia d'olio, partendo dunque con specifiche generali per poi passare progressivamente, ai requisiti, fino al massimo dettaglio e avanzando per sottoproblemi.



## 4. Schema logico

Una volta affrontato il modello ER, è stato trasformato in un modello logico, comprensibile dal DBMS utilizzato per la costruzione dell'applicazione, seguendo le procedure:

- Eliminazione delle gerarchie
- Selezione delle chiavi primarie, eliminazione delle identificazioni esterne
- Normalizzazione degli attributi composti o multipli
- Traduzione di entità e associazioni in schemi di relazioni
- Verifica di normalizzazione



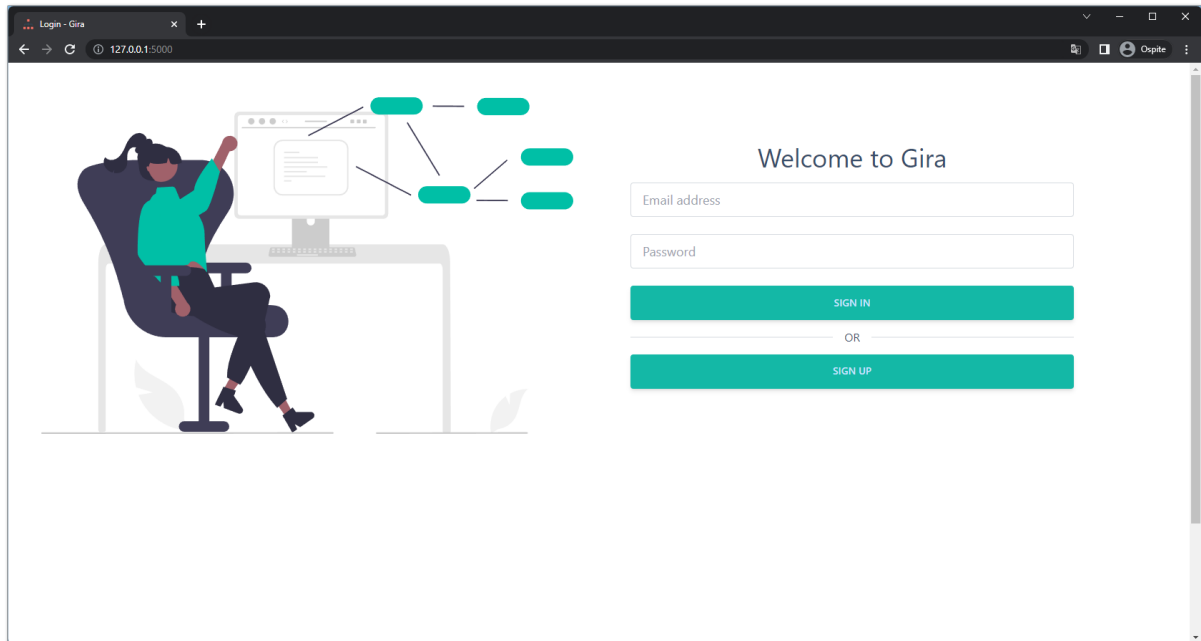
## 5. Popolazione della base di dati

Come esempio dimostrativo, la base di dati è stata popolata con utenti fittizi e con un progetto di un sistema embedded, il quale potrebbe essere tranquillamente sviluppabile e organizzato tramite la web app.

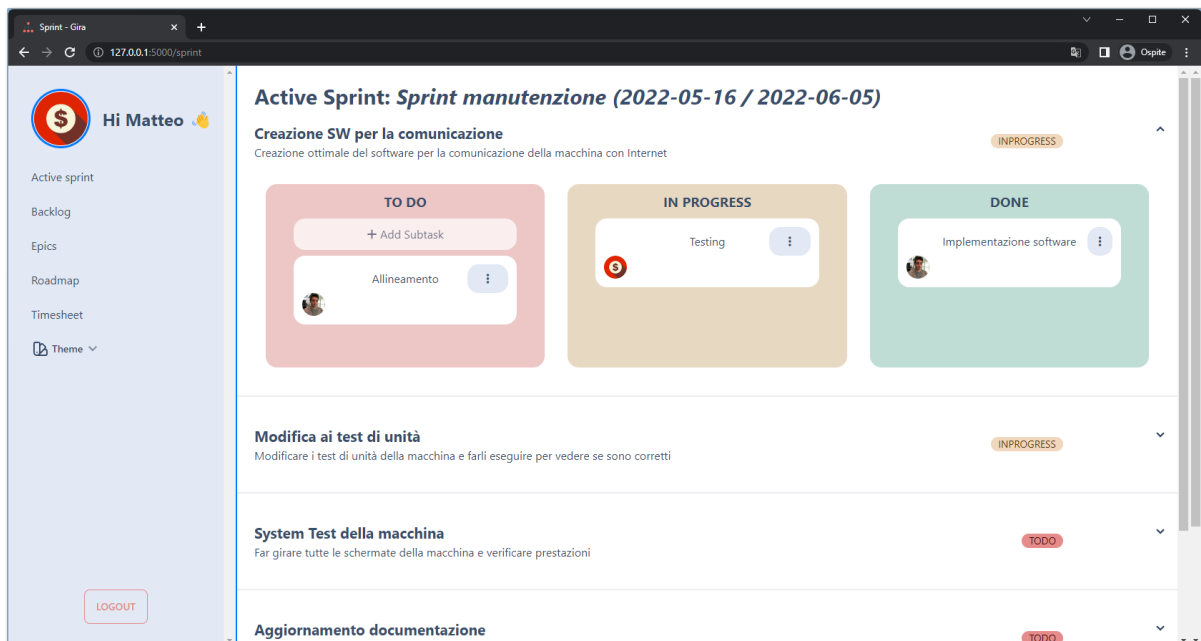
 [DATABASE FILE](#)

## 6. Stampa delle pagine web

### Pagina Sign In / Sign Up



### Pagina Active Sprint



## Pagina Backlog

**Sprint** Total points: 100

**Active** Sprint manutenzione (2022-05-16 / 2022-06-05)

Status	Task	Category	Priority	FB
INPROGRESS	Creazione SW per la comunicazione	Nuovo prodotto	1	0 FB
INPROGRESS	Modifica ai test di unità	Manutenzione	5	0 FB
TODO	System Test della macchina	Manutenzione	2	0 FB
TODO	Aggiornamento documentazione	Documentazione	13	1 FB
TODO	Partecipazione Corso DB	Manutenzione	14	0 FB

Close sprint (8 days remaining)

**Backlog** 4 issues Add Task Add epic

Status	Task	Category	Priority	FB
TODO	Definizione Budget		6	0 FB
TODO	Creazione versione Beta		9	4 FB
TODO	Aggiornamento menu	Manutenzione	10	0 FB
TODO	Pulizia codice	Manutenzione	11	0 FB

## Pagina Epiche

**Epics**

**Manutenzione**  
Manutenzione del codice

Related tasks:

- System Test della macchina
- Modifica ai test di unità
- Aggiornamento menu
- Pulizia codice
- Partecipazione Corso DB

**EDIT**

**Nuovo prodotto**  
Nuovo prodotto da creare

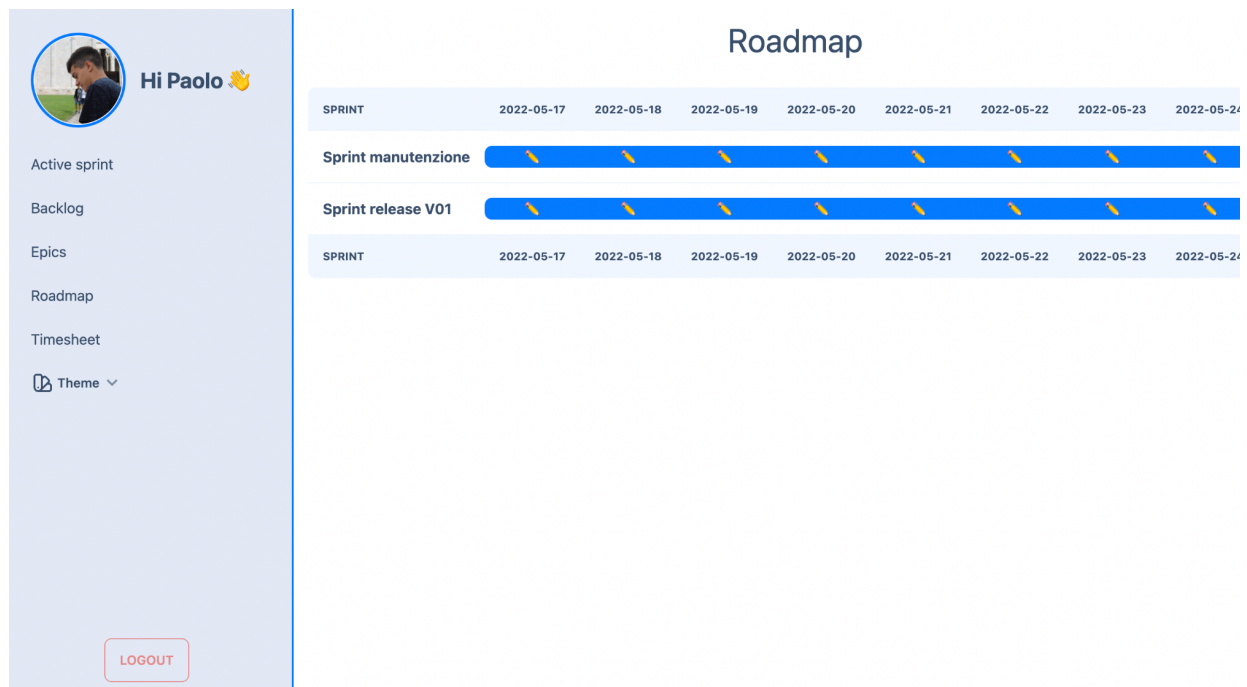
Related tasks:

- Creazione SW per la comunicazione

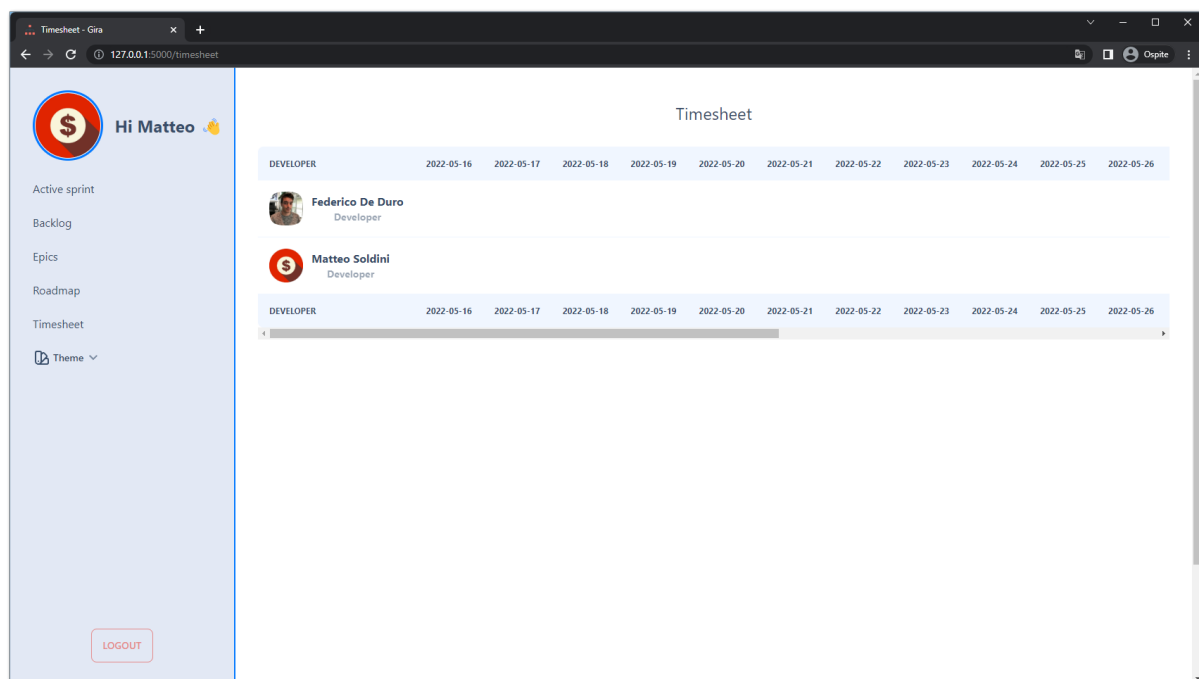
**EDIT**

**Documentazione**

## Pagina Roadmap



## Pagina Timesheet





## Pagina Profilo

The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/profile". The page is titled "Matteo Soldini's profile".

**Left Sidebar:**

- Hi Matteo (with a circular profile picture icon containing a dollar sign)
- Active sprint
- Backlog
- Epics
- Roadmap
- Timesheet
- Theme (with a dropdown arrow)
- LOGOUT (button)

**Main Content Area:**

**Matteo Soldini's profile**

**Information**

(Circular profile picture icon containing a dollar sign)

Matteo

Soldini

m.soldini1@studenti.uniibg.it

Manager? ☐

(Edit icon) (Delete icon)

**Password**

Current password

New password

(Edit icon)