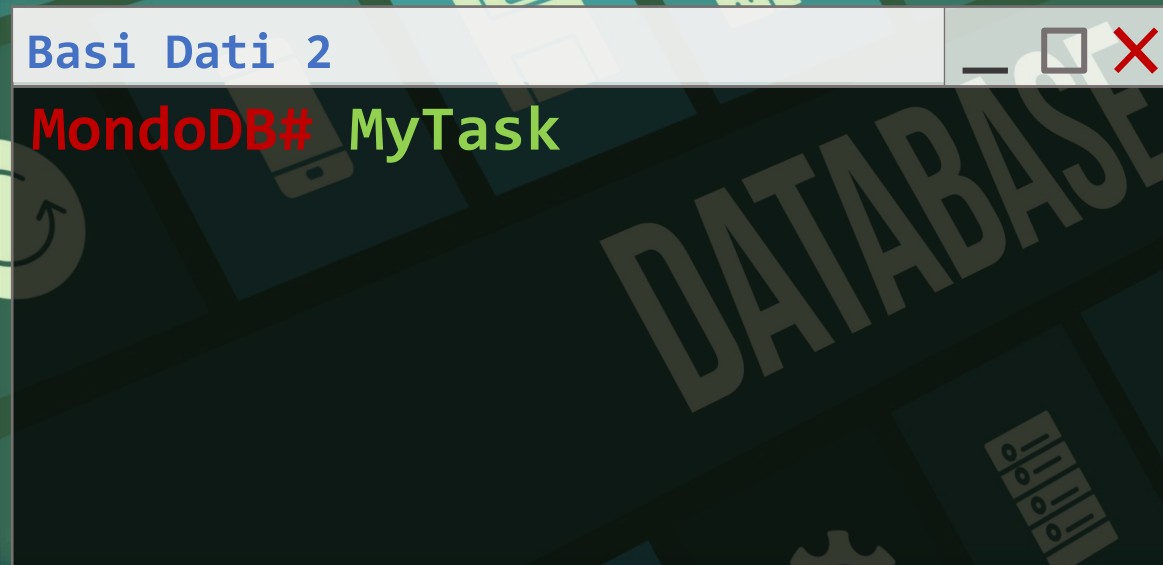


Università degli Studi di Salerno

Dipartimento di Informatica



Pasquale Matrone



DESCRIZIONE

MyTask

Applicazione in **Node.js**
che usa il DBMS
MongoDB per la gestione
dei task (o «*storie*») di un
progetto (o «*sprint*»)



Overview funzionalità applicazione

Overview funzionalità applicazione

Ruoli App

- L'applicazione prevede tre tipi di utente (**ruoli**): Sviluppatore, Project Manager e Manager

RUOLO	DESCRIZIONE
Sviluppatore	<ul style="list-style-type: none">• Visualizza, crea, assegna, si autoassegna task
Project Manager	<ul style="list-style-type: none">• Visualizza, crea, assegna, si autoassegna task• Visualizza, crea progetti
Manager	<ul style="list-style-type: none">• Visualizza, crea, assegna, si autoassegna task• Visualizza, crea progetti• Visualizza i «numeri» di un progetto

Overview funzionalità applicazione

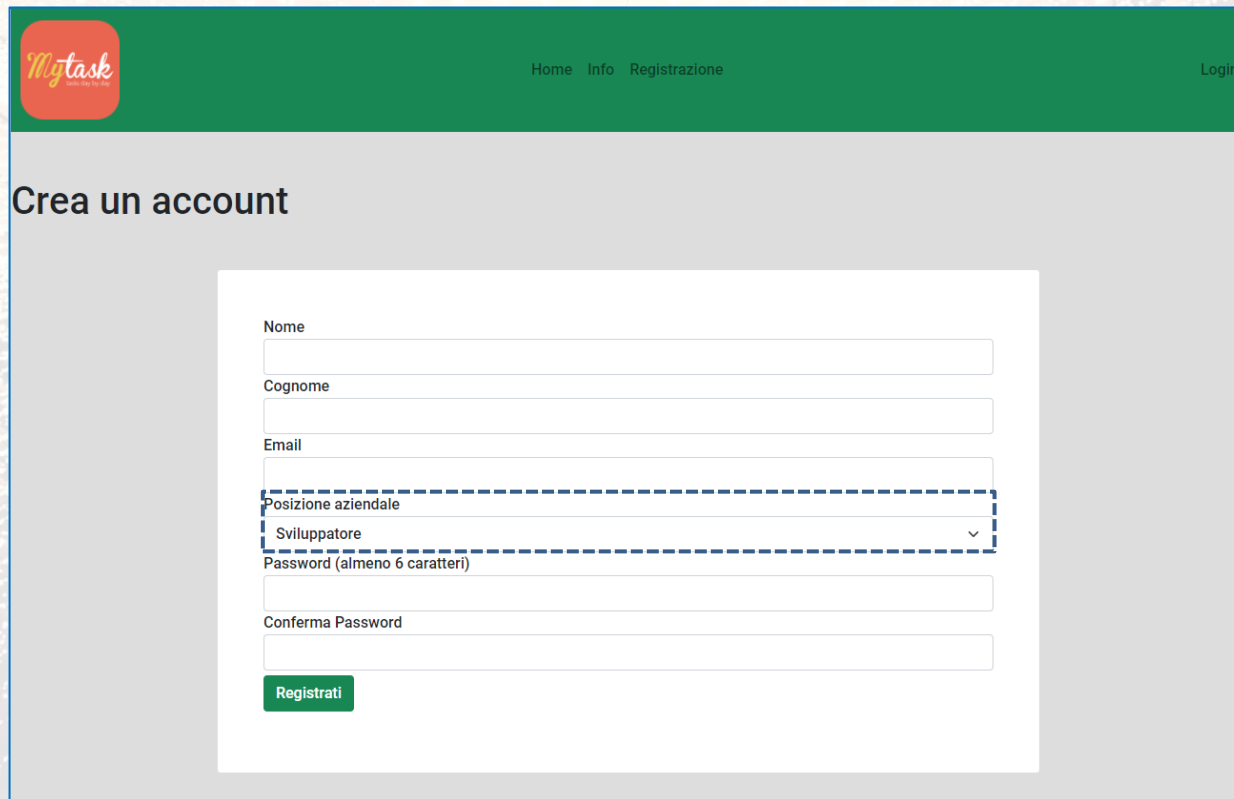
MyTask | 1/4

- Principali funzionalità App:
 - Registrazione nuovo utente, **login**, logout
 - **Creazione nuovo task**
 - **Visualizzazione task** (assegnati, creati, assegnati e creati)
 - Lavagna task per l'auto-assegnamento
 - Storico task (task già completati)
 - Ricerca testuale dei task (**pattern di ricerca testuale**)
 - Creazione nuovo progetto
 - Visualizzazione progetti (attivi e non attivi)
 - Visualizzazione dei «*numeri*» di un progetto
 - Numero di task per il progetto X in stato Y assegnati a Z

Overview funzionalità applicazione

MyTask | 2/4

- Registrazione **nuovo utente**: tre **posizioni aziendali** per i diversi ruoli nell'app (sviluppatore, pm, manager)



The screenshot shows the 'Crea un account' (Create an account) page of the MyTask application. The page has a green header with the MyTask logo on the left and navigation links 'Home', 'Info', 'Registrazione', and 'Login' on the right. The main content area is light gray and contains the title 'Crea un account'. Below the title is a white registration form with the following fields: 'Nome' (Name), 'Cognome' (Surname), 'Email', 'Posizione aziendale' (Company Position) which is a dropdown menu currently showing 'Sviluppatore' (Developer), 'Password (almeno 6 caratteri)' (Password (at least 6 characters)), and 'Conferma Password' (Confirm Password). A green 'Registrati' (Register) button is located at the bottom of the form.

Mytask

Home Info Registrazione Login

Crea un account

Nome

Cognome

Email

Posizione aziendale

Sviluppatore

Password (almeno 6 caratteri)

Conferma Password

Registrati

Overview funzionalità applicazione

MyTask | 3/4

- **Lista Task** Assegnati e Creati, Assegnati, Creati, Lavagna, Storico

The screenshot shows the MyTask application interface. At the top, there is a green header with the MyTask logo, navigation links (Home, Info, My Task, Progetti), and a user profile section (Logout pasqualematrone1994@gmail.com). Below the header is a search bar with the placeholder text 'Pattern di ricerca (solo lettere)...'. The main content area is titled 'Lista dei Task' and displays details for a specific task with ID 'AMP-61e02ea3ce07e1d8e9b95753'. The task details include: Progetto: Amplifon HR, Sprint: 3, Descrizione: Implementare item assunzione nuovo dipendente, Stato: In Lavorazione, Effort: 10 giorni, Creato da: pasqualematrone1994@gmail.com, Assegnatario: angelomariaesposito1995@gmail.com, and Commenti: Allegati inviati per email. A progress bar shows 36% completion. Below the progress bar is a link 'Contatta il creatore del task'. At the bottom, there is a yellow box with the delivery date 'Data Consegna: 23/01/2022' and two buttons: 'Modifica' (blue) and 'Cancella' (red). An illustration of a red pencil checking off three boxes is overlaid on the right side of the task details.

Overview funzionalità applicazione

MyTask | 4/4

- **Nuovo Task:** tutti i campi, tranne *Commenti*, sono obbligatori

The screenshot displays the 'MyTask' application interface. At the top, a green header bar contains the 'MyTask' logo on the left, navigation links 'Home', 'Info', 'My Task', and 'Progetti' in the center, and a 'Logout' link with the email 'pasqualematrone1994@gmail.com' on the right. A dropdown menu is open under 'My Task', listing 'Nuovo Task' (highlighted with a dashed border), 'Task Assegnati e Creati', 'Task Assegnati', 'Task Creati', 'Lavagna Task', and 'Storico Task'. Below the header, the main content area is titled 'Aggiungi una nuovo task'. The form includes several fields: 'Progetto' (a dropdown menu currently showing 'Enel Global Travel'), 'Sprint' (an empty text input), 'Descrizione' (a large empty text area), 'Stato' (a dropdown menu showing 'In Lavorazione'), 'Data Consegna' (a date input field showing 'gg/mm/aaaa'), 'Assegnatario' (a dropdown menu showing 'Lavagna'), and 'Commenti' (a large empty text area). A green 'Crea' button is located at the bottom left of the form.



Principali Tecnologie

Node.js

MongoDB

Handlebars



Principali Tecnologie

Node.js

MongoDB

Handlebars

Principali Tecnologie

Node.js | 1/3

- **Node.js** è un runtime system open source multiplatforma orientato agli eventi per l'esecuzione di codice **JavaScript**
 - Un runtime system è un software che fornisce i servizi necessari all'esecuzione di un programma
- Node.js consente di utilizzare JavaScript anche per scrivere codice da eseguire **lato server**
 - Ad esempio, per la produzione del contenuto delle pagine web dinamiche prima che la pagina venga inviata al browser dell'utente



Principali Tecnologie

Node.js | 2/3

- Node.js permette quindi di implementare il cosiddetto paradigma «**JavaScript everywhere**»
- **npm** (abbreviazione di *Node Package Manager*) è un gestore di **pacchetti** per il linguaggio di programmazione JavaScript
 - È il gestore di pacchetti predefinito di Node.js
 - Consiste in un client da linea di comando, chiamato anch'esso *npm*, e un database online di pacchetti pubblici e privati, chiamato *npm registry*
 - **Di seguito, un elenco dei pacchetti usati per l'implementazione dell'app MyTask**



Principali Tecnologie

Node.js | 3/3

PACCHETTO	DESCRIZIONE
Mongoose	<ul style="list-style-type: none">Fornisce una soluzione semplice e basata su schema per modellare i dati dell'applicazioneInclude il casting di tipo integrato, la validazione e la costruzione di query
Express	<ul style="list-style-type: none">Framework per applicazioni web per Node.js
Express-session	<ul style="list-style-type: none">Crea un middleware di sessione
Connect-flash	<ul style="list-style-type: none">Area speciale della sessione per memorizzare i messaggi
Passport	<ul style="list-style-type: none">Middleware di autenticazione
Bcrypt	<ul style="list-style-type: none">Funzione crittografica di hash
Body-parser	<ul style="list-style-type: none">Analizzare i body delle richieste in arrivo (proprietà <i>req.body</i>)

Principali Tecnologie

Node.js | 3/3

PACCHETTO	DESCRIZIONE
Mongoose	<ul style="list-style-type: none">Fornisce una soluzione semplice e basata su schema per modellare i dati.
Express	<ul style="list-style-type: none">Framework web integrato, la soluzione di query per le applicazioni web per Node.js.
Express-session	<ul style="list-style-type: none">Crea un middleware di sessione.
Connect-flash	<ul style="list-style-type: none">Area speciale della sessione per memorizzare i messaggi.
Passport	<ul style="list-style-type: none">Middleware di autenticazione.
Bcrypt	<ul style="list-style-type: none">Funzione crittografica di hash.
Body-parser	<ul style="list-style-type: none">Analizzare i body delle richieste in arrivo (proprietà <i>req.body</i>).

Maggiori dettagli in seguito...



Principali Tecnologie

Node.js | 3/3

PACCHETTO

Mongoose

Express

Express-session

Connect-flash

Passport

Bcrypt

Body-parser

- Per **Routing** si intende determinare come un'applicazione risponde ad una richiesta client per un **endpoint** particolare
- Un endpoint è formato da
 1. un URI (o percorso)
 2. un metodo di richiesta HTTP specifico (GET, POST, ecc)

Client

```
<li><a class="dropdown-item" href="/lista_progetti">Lista Progetti Attivi</a></li>
```

Server

```
//Route per pagina lista progetti attivi: http://localhost:3000/lista_progetti
app.get('/lista_progetti', accessoSicuro, function(req , res) {
  Progetti.find({attivo: true}).lean().sort({date: 'desc'})
    .then(function(docs) {
      res.render('lista_progetti', {
        progetti: docs
      });
    });
});
```




Principali Tecnologie

Node.js

MongoDB

Handlebars

Principali Tecnologie

MongoDB | 1/3

- **MongoDB** è un DBMS **non relazionale**, orientato ai **documenti**
 - Il nome deriva da «humongous», enorme
- Classificato come un database di tipo **NoSQL**, MongoDB si allontana dalla struttura tradizionale basata su tabelle dei database relazionali in favore di documenti in stile **JSON** con schema dinamico (**BSON**), rendendo l'integrazione di dati di alcuni tipi di applicazioni più facile e veloce



mongoDB®

Principali Tecnologie

MongoDB| 2/3

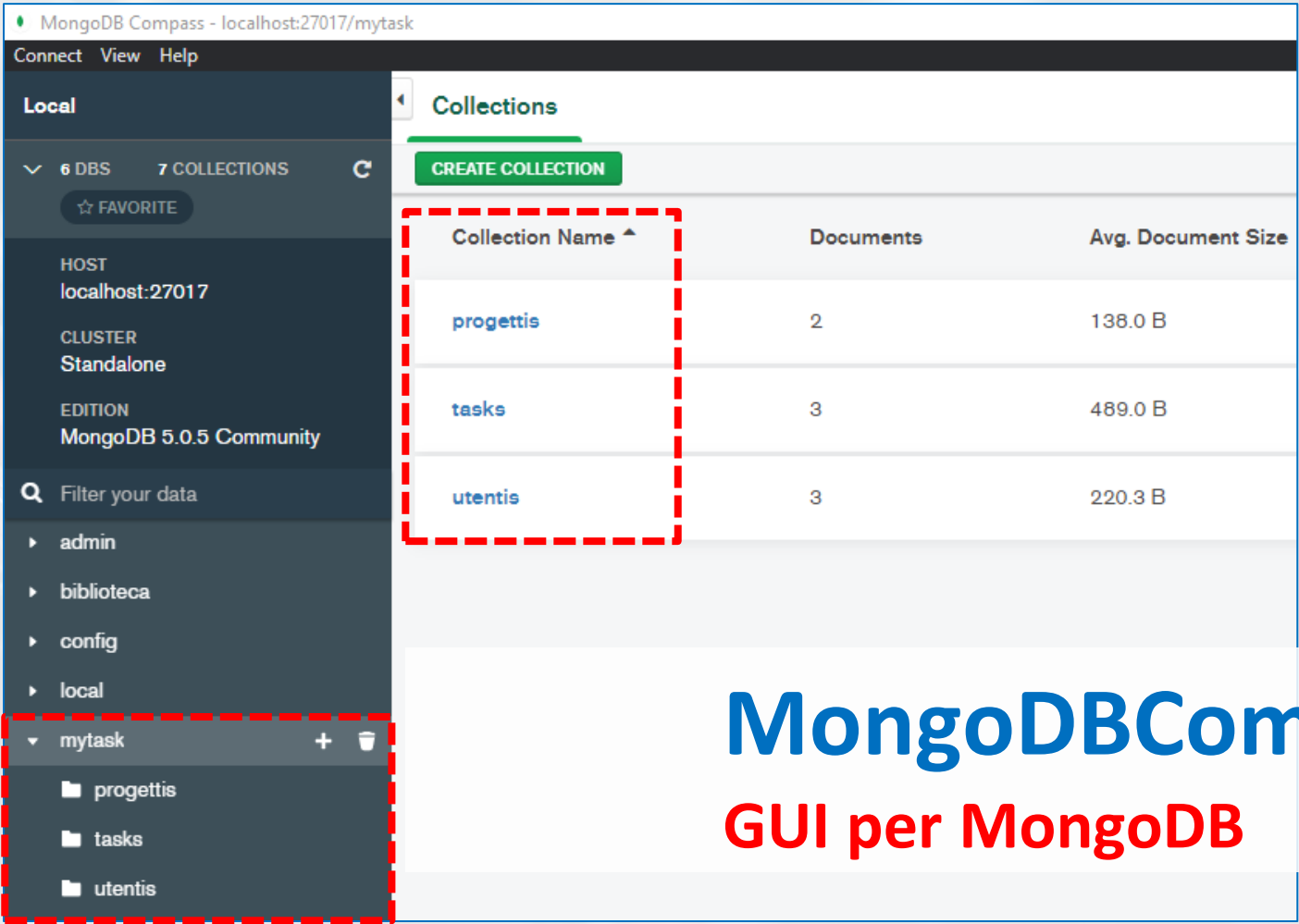
- Secondo db-engines.com, a gennaio 2022 MongoDB è al **5º** posto della **classifica** dei database più popolari del mondo e al 1º posto dei database NoSQL
- Alcuni degli utilizzatori più popolari di MongoDB sono
 - SAP
 - Forbes
 - The New York Times
 - Sourceforge
 - CERN
 - eBay
 - ...

The most popular database management systems

January 2022	Score
1. Oracle	1267
2. MySQL	1206
3. Microsoft SQL Server	945
4. PostgreSQL	607
5. MongoDB	489

Principali Tecnologie

MongoDB | 3/3



The screenshot displays the MongoDB Compass application interface. The title bar reads "MongoDB Compass - localhost:27017/mytask". The top navigation bar includes "Connect", "View", and "Help". The left sidebar, titled "Local", shows the connection details: "6 DBS", "7 COLLECTIONS", and "localhost:27017". It also lists the cluster as "Standalone" and the edition as "MongoDB 5.0.5 Community". A search bar "Filter your data" is present. Below, a list of databases is shown, with "mytask" selected and expanded to show collections: "progettis", "tasks", and "utentis". The main panel, titled "Collections", features a "CREATE COLLECTION" button and a table of existing collections. The table has columns for "Collection Name", "Documents", and "Avg. Document Size". The collections listed are "progettis" (2 documents, 138.0 B), "tasks" (3 documents, 489.0 B), and "utentis" (3 documents, 220.3 B). Red dashed boxes highlight the "mytask" database in the sidebar and the "Collections" table in the main panel.

Collection Name	Documents	Avg. Document Size
progettis	2	138.0 B
tasks	3	489.0 B
utentis	3	220.3 B

MongoDB Compass
GUI per MongoDB

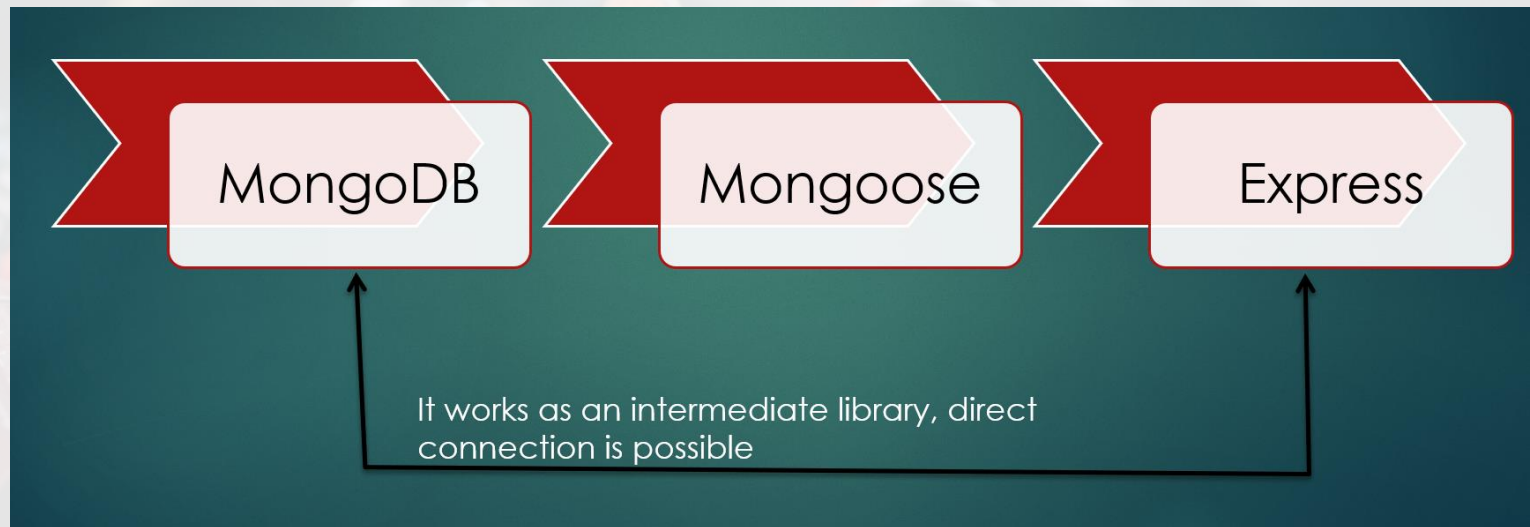
Principali Tecnologie

MongoDB | Mongoose | 1/3

PACCHETTO	DESCRIZIONE
Mongoose	<ul style="list-style-type: none">Fornisce una soluzione semplice e basata su schema per modellare i dati dell'applicazioneInclude il casting di tipo integrato, la validazione e la costruzione di query
Express	<ul style="list-style-type: none">Framework per applicazioni web per Node.js
Express-session	<ul style="list-style-type: none">Crea un middleware di sessione
Connect-flash	<ul style="list-style-type: none">Area speciale della sessione per memorizzare i messaggi
Passport	<ul style="list-style-type: none">Middleware di autenticazione
Bcrypt	<ul style="list-style-type: none">Funzione crittografica di hash
Body-parser	<ul style="list-style-type: none">Analizzare i body delle richieste in arrivo (proprietà <i>req.body</i>)

Principali Tecnologie

MongoDB | Mongoose | 2/3



Mongoose: libreria *JavaScript* orientata agli oggetti, che crea una **connessione** tra **MongoDB** ed **Express**

Principali Tecnologie

MongoDB | Mongoose | 3/3

mongoose

elegant **mongodb** object modeling for **node.js**

[read the docs](#)[discover plugins](#)[Star](#) 23,757

Version 6.1.6

[Fork](#) 3,350

Let's face it, **writing MongoDB validation, casting and business logic boilerplate is a drag**. That's why we wrote Mongoose.

```
const mongoose = require('mongoose');
mongoose.connect('mongodb://localhost:27017/test');

const Cat = mongoose.model('Cat', { name: String });

const kitty = new Cat({ name: 'Zildjian' });
kitty.save().then(() => console.log('meow'));
```

Mongoose provides a straight-forward, schema-based solution to model your application data. It includes built-in type casting, validation, query building, business logic hooks and more, out of the box.

Principali Tecnologie

MongoDB | Mongoose | 3/3

mongoose

elegant m

Mongoose «*semplifica*» lo sviluppo con MongoDB

Ecco perché esiste ed ho utilizzato Mongoose

Star 23,757

Version 6.1.6

Fork 3,350

Let's face it, writing MongoDB validation, casting and business logic boilerplate is a drag. That's why we wrote Mongoose.

```
const mongoose = require('mongoose');
mongoose.connect('mongodb://localhost:27017/test');

const Cat = mongoose.model('Cat', { name: String });

const kitty = new Cat({ name: 'Zildjian' });
kitty.save().then(() => console.log('meow'));
```

Mongoose provides a straight-forward, schema-based solution to model your application data. It includes built-in type casting, validation, query building, business logic hooks and more, out of the box.

Principali Tecnologie

MongoDB | Mongoose | 3/3

mongoose

elegant m

Mongoose «*semplifica*» lo sviluppo con MongoDB

Ecco perché esiste ed ho utilizzato Mongoose

Star 23,757

Version 6.1.6

Fork 3,350

Let's face it, writing MongoDB validation, casting and business logic boilerplate is a drag. That's why we wrote Mongoose.

Approfondiamo le caratteristiche principali di Mongoose



business logic hooks and more, out of the box.

Caratteristiche Mongoose

1

Schema

2

Modello

3

Query

4

Indici

Caratteristiche Mongoose

1

Schema

2

Modello

3

Query

4

Indici

Principali Tecnologie

MongoDB | Mongoose | Schema

- Ogni **schema** mappa una **collezione** MongoDB e definisce la forma dei **documenti** all'interno di quella collezione
 - Definisce il **tipo** di ogni **proprietà**
 - String
 - Number
 - Date
 - Boolean
 - Array
 - ...

```
import mongoose from 'mongoose';
const { Schema } = mongoose;

const blogSchema = new Schema({
  title: String, // String is shorthand for {type: String}
  author: String,
  body: String,
  comments: [{ body: String, date: Date }],
  date: { type: Date, default: Date.now },
  hidden: Boolean,
  meta: {
    votes: Number,
    favs: Number
  }
});
```


Caratteristiche Mongoose

1

Schema

2

Modello

3

Query

4

Indici

Principali Tecnologie

MongoDB | Mongoose | Modello

- I **modelli** sono responsabili della creazione e della lettura dei documenti dal sottostante database MongoDB
 - I modelli sono definiti a partire da uno **schema**
 - Il primo argomento della funzione *mongoose.model()* è il nome singolare della collezione
 - Un'**istanza** di un modello è chiamata **documento**

```
const schema = new mongoose.Schema({ name: 'string', size: 'string' });  
const Tank = mongoose.model('Tank', schema);
```


Caratteristiche Mongoose

1

Schema

2

Modello

3

Query

4

Indici

Principali Tecnologie

MongoDB | Mongoose | Query

- Alcune funzioni statiche di un modello Mongoose per le operazioni **CRUD** (**c**reate, **r**ead, **u**ppdate, **d**eleate) sono le seguenti:
 - Model.deleteMany()
 - Model.deleteOne()
 - Model.find()
 - Model.findById()
 - Model.findByIdAndDelete()
 - Model.findByIdAndUpdate()
 - E tante altre...
- Ciascuna di queste funzioni restituisce un oggetto *Query* di tipo *Mongoose*

Principali Tecnologie

MongoDB | Mongoose | Query

- Alcune funzioni statiche di un modello Mongoose per le operazioni **CRUD** (create, read, update, delete) sono le seguenti:

SEMPLICE ESEMPIO

```
/* Trovare i libri intitolati «I promessi sposi» */
Libro.find({titolo: "I promessi sposi"}, function
(err, docs){
    if(err){
        return console.log(err);
    }
    console.log('Found documents filtered by { autore:
"I promessi sposi" } =>', docs);
});
```

Caratteristiche Mongoose

1

Schema

2

Modello

3

Query

4

Indici

Principali Tecnologie

MongoDB | Mongoose | Indici

- Oltre all'indice di default sul campo «_id», MongoDB permette di creare **indici** su altre proprietà dello schema
 - In questo modo, i documenti, possono essere indicizzati in base alle nostre proprietà per un accesso più veloce
 - Possiamo anche creare indici composti da più di una proprietà (*compound indexes*)
 - **ModelSchema.index()**

```
//Indici custom per le mie query
taskSchema.index({utente: 1});
taskSchema.index({assegnatario: 1});
```



Principali Tecnologie

Node.js

MongoDB

Handlebars

Principali Tecnologie

Handlebars

- **Handlebars** è un semplice linguaggio di **template**
 - Un «espressione handlebars» viene sostituita con i valori di un oggetto di input
- A completare il pacchetto delle tecnologie **front-end** utilizzate nell'applicazione MyTask, troviamo
 - HTML
 - CSS
 - **Bootstrap**
 - Componenti interfaccia sito web
 - **jQuery**
 - Libreria JavaScript «*feature-rich*»





Principali Tecnologie

Considerazioni

Node.js

MongoDB

Handlebars

Principali Tecnologie

Considerazioni

- Node.js, Handlebars, ecc sono tecnologie in rapida evoluzione
 - L'utilizzo di siti come **Stack Overflow**, oltre alle documentazioni ufficiali, supportano tantissimo il programmatore
 - Stack Overflow è un sito web in cui si possono porre domande su argomenti di programmazione





Principali Tecnologie

Riassunto

Node.js

MongoDB

Handlebars

Principali Tecnologie

Database | ...



mongoDB®

Principali Tecnologie

Database | BE | ...



Principali Tecnologie

Database | BE | FE | ...



Principali Tecnologie

Database | BE | FE | Strumenti



App MyTask

Database

Esempio: Lista Task

Miglioramenti

App MyTask

Database

Esempio: Lista Task

Miglioramenti

MyTask

Database | MongoDB

The screenshot displays the MongoDB Compass interface. On the left, the 'Local' database is selected, showing 6 databases and 9 collections. The 'mytask' database is expanded, revealing three collections: 'progettis', 'tasks', and 'utentis'. The main panel shows the 'Collections' tab with a 'Create collection' button and a 'View' dropdown. Below, three collection cards are listed, each showing a storage size of 20.48 kB and a document count.

Collection Name	Storage size	Documents
progettis	20.48 kB	2
tasks	20.48 kB	3
utentis	20.48 kB	3

The screenshot displays the MongoDB Compass interface. On the left, the 'Local' database is selected, showing 6 DBS and 9 COLLECTIONS. The 'mytask' database is expanded, revealing three collections: 'progettis', 'tasks', and 'utentis'. The main content area shows the details for the 'progettis' collection, including its storage size (20.48 kB) and the number of documents (2). The 'tasks' and 'utentis' collections are also listed with their respective storage sizes and document counts.

Collection	Storage size	Documents
progettis	20.48 kB	2
tasks	20.48 kB	3
utentis	20.48 kB	3

Local

6 DBS 9 COLLECTIONS

☆ FAVORITE

HOST
localhost:27017

CLUSTER
Standalone

EDITION
MongoDB 5.0.5 Community

Filter your data

- admin
- biblioteca
- config
- local
- mytask +
- progettis
- tasks
- utentis

Collections

Create collection View

progettis

Storage size: 20.48 kB Documents: 2

tasks

Storage size: 20.48 kB Documents: 3

utentis

Storage size: 20.48 kB Documents: 3

```
_id: ObjectId("61e02e51ce07e1d8e9b95743")  
nome: "Enel Global Travel"  
cliente: "Enel SpA"  
descrizione: "Sistema gestione viaggi di lavoro"  
attivo: true  
__v: 0
```


The screenshot displays the MongoDB Compass web interface. On the left sidebar, under the 'Local' section, the 'mytask' database is expanded, showing three collections: 'progettis', 'tasks', and 'utentis'. The 'tasks' collection is highlighted with a red dashed rectangular box. The main panel on the right, titled 'Collections', shows details for the selected 'tasks' collection. It indicates a storage size of 20.48 kB and contains 3 documents. The 'progettis' and 'utentis' collections are also visible with the same storage size and 2 and 3 documents respectively.

Collection	Storage size	Documents
progettis	20.48 kB	2
tasks	20.48 kB	3
utentis	20.48 kB	3

```
_id: ObjectId("61e02ea3ce07e1d8e9b95753")
prefisso_task: "AMP"
progetto: "Amplifon HR"
sprint: 3
descrizione: "Implementare item assunzione nuovo dipendente"
commento: "Allegati inviati per email"
data_consegna: 2022-01-23T00:00:00.000+00:00
utente: "61e02dcdce07e1d8e9b9572c"
creato_da: "pasqualematrone1994@gmail.com"
assegnatario: "angelomariaesposito1995@gmail.com"
stato: "In Lavorazione"
percentuale: "35%"
valore_percentuale: "35"
data_consegna_toStr_ : "23/01/2022"
effort: 10
data: 2022-01-13T13:52:35.978+00:00
__v: 0
```

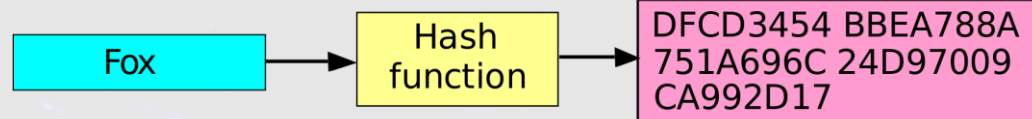

The screenshot displays the MongoDB Compass interface. On the left, the 'Local' sidebar shows the database 'mytask' selected, with a tree view containing collections 'progettis', 'tasks', and 'utentis'. The main panel shows the 'Collections' tab for 'mytask', listing three collections: 'progettis', 'tasks', and 'utentis'. Each collection entry shows its storage size (20.48 kB) and the number of documents (2 for 'progettis', 3 for 'tasks' and 'utentis'). The 'utentis' collection is highlighted with a red dashed border.

Collection	Storage size	Documents
progettis	20.48 kB	2
tasks	20.48 kB	3
utentis	20.48 kB	3

```
_id: ObjectId("61e02df7ce07e1d8e9b9572f")
nome: "Antonio"
cognome: "Marchesano"
email: "antoniomarchesano1989@gmail.com"
password: "$2a$10$S880MBmuCCE7nrc/aabJeuWR/.AzECzcWzdZB1cOzuPY19a6noSky"
pm: true
manager: false
data: 2022-01-13T13:49:43.584+00:00
__v: 0
```


- **bcrypt.js**
 - Libreria per eseguire l'**hashing** delle password
 - Una funzione crittografica di hash è un algoritmo che mappa un messaggio di lunghezza arbitraria in una stringa binaria di dimensione fissa chiamata **valore di hash**
 - Aumenta la sicurezza dell'app

```
cognome: "Marchesano"  
email: "antoniomarchesano1989@gmail.com"  
password: "$2a$10$S88oMBmuCCE7nrc/aabJeuWR/.AzECzcWzdZB1cOzuPY19a6noSky"  
pm: true  
manager: false  
data: 2022-01-13T13:49:43.584+00:00  
__v: 0
```

Input**Hash sum**

Database | MongoDB | Considerazioni finali

- I principali **vantaggi** riscontrati nell'utilizzare MongoDB come **database** nella progettazione ed **implementazione** dell'app MyTask sono stati i seguenti

1. Per i **documenti**, non è obbligatorio avere una struttura rigida fissa
2. Inserire nuove informazioni è facilissimo (**proprietà** dei documenti)
3. Ben **documentato** (documentazione ufficiale, Stack Overflow, Googling, ecc)

App MyTask

Database

Esempio: Lista Task

Miglioramenti

{ Esempio: Lista Task | Schema }

```
const mongoose = require('mongoose');

const taskSchema = mongoose.Schema({
  prefisso_task: {
    type: String,
    required: true
  },
  progetto: {
    type: String,
    required: true
  },
  sprint: {
    type: Number
  },
  descrizione: {
    type: String,
    required: true
  },
  commento: {
    type: String,
    required: false
  },
  data: {
    type: Date,
    default: Date.now
  },
  data_consegna: {
    type: Date,
    default: Date.now
  },
  utente: {
    type: String,
    required: true
  },
});
```



task.js

{ Esempio: Lista Task | Documento }

```
_id: ObjectId("61e02ea3ce07e1d8e9b95753")
prefisso_task: "AMP"
progetto: "Amplifon HR"
sprint: 3
descrizione: "Implementare item assunzione nuovo dipendente"
commento: "Allegati inviati per email"
data_consegna: 2022-01-23T00:00:00.000+00:00
utente: "61e02dcdce07e1d8e9b9572c"
creato_da: "[REDACTED]@gmail.com"
assegnatario: "ang[REDACTED]@gmail.com"
stato: "In Lavorazione"
percentuale: "35%"
valore_percentuale: "35"
data_consegna_toStr_ : "23/01/2022"
effort: 10
data: 2022-01-13T13:52:35.978+00:00
__v: 0
```

{ Esempio: Lista Task | Route | 1/2 }

```
{{#if user}}  
<div class="dropdown">  
  <button class="btn dropdown-toggle" type="button" id="dropdownMenuButton1" data-bs-toggle="dropdown"  
    aria-expanded="false">  
    My Task  
  </button>  
  <ul class="dropdown-menu" aria-labelledby="dropdownMenuButton1">  
    <li><a class="dropdown-item" href="/aggiungi_task">Nuovo Task</a></li>  
    <li><a class="dropdown-item" href="/lista_task">Task Assegnati e Creati</a></li>  
    <li><a class="dropdown-item" href="/assegnati_a_me">Task Assegnati</a></li>  
    <li><a class="dropdown-item" href="/creati_da_me">Task Creati</a></li>  
    <li><a class="dropdown-item" href="/lavagna">Lavagna Task</a></li>  
    <li><a class="dropdown-item" href="/storico">Storico Task</a></li>  
  </ul>  
</div>  
  
{{#if pm}}
```



{ Esempio: Lista Task | Route | 1/2 }

```
{{#if user}}  
<div class="dropdown">  
  <button class="btn dropdown-toggle" type="button" id="dropdownMenuButton1" data-bs-toggle="dropdown"  
    aria-expanded="false">  
    My Task  
  </button>  
  <ul class="dropdown-menu" aria-labelledby="dropdownMenuButton1">  
    <li><a class="dropdown-item" href="/aggiungi_task">Nuovo Task</a></li>  
    <li><a class="dropdown-item" href="/lista_task">Task Assegnati e Creati</a></li>  
    <li><a class="dropdown-item" href="/assegnati_a_me">Task Assegnati</a></li>  
    <li><a class="dropdown-item" href="/creati_da_me">Task Creati</a></li>  
    <li><a class="dropdown-item" href="/lavagna">Lavagna Task</a></li>  
    <li><a class="dropdown-item" href="/storico">Storico Task</a></li>  
  </ul>  
</div>  
  
{{#if pm}}
```



{ Esempio: Lista Task | Route | 2/2 }

```
//Route per pagina lista task: http://localhost:3000/lista\_task  
app.get('/lista_task', accessoSicuro, function(req , res) {  
  Task.find({$or:[{utente: req.user.id}, {assegnatario: req.user.email}],  
    stato: {$ne: "Completato"}}).lean().sort({date: 'desc'})  
  .then(function(docs) {  
    res.render('lista_task', {  
      task: docs,  
      search: true  
    });  
  });  
});
```



{ Esempio: Lista Task | Route | 2/2 }

Azionato da [Lista Task](/lista_task)

```
//Route per pagina lista task: http://localhost:3000/lista_task
app.get('/lista_task', accessoSicuro, function(req , res) {
  Task.find({$or:[{utente: req.user.id}, {assegnatario: req.user.email}],
    stato: {$ne: "Completato"}}).lean().sort({date: 'desc'})
    .then(function(docs) {
      res.render('lista_task', {
        task: docs,
        search: true
      });
    });
});
```

lista_task.handlebars




app.js

{ Esempio: Lista Task | View | 1/2 }

```
<h1 class="card-title">Lista dei Task</h1>
{{#each task}}
<div class="card card-body mb-5" id="divTaskID">
  <h3>{{prefisso_task}}-{{_id}}</h3>
  <p><strong>Progetto: </strong>{{progetto}}</p>
  <p><strong>Sprint: </strong>{{sprint}}</p>
  <p><strong>Descrizione: </strong>{{descrizione}}</p>
  <p><strong>Stato: </strong>{{stato}}</p>
  <p><strong>Effort: </strong>{{effort}} giorni</p>
  <p><strong>Creato da: </strong>{{creato_da}}</p>
  <p><strong>Assegnatario: </strong>{{assegnatario}}</p>
  {{#if commento}}
  <p><strong>Commenti: </strong>{{commento}}</p>
  {{/if}}
  <p>
    <div class="progress">
      <div class="progress-bar progress-bar-striped bg-success" role="progressbar" style="width: {{p}}%></div>
    </div>
  </p>
  <a href="mailto:{{creato_da}}">Contatta il creatore del task</a>
  <p>
    <div class="alert alert-warning" role="alert" style="text-align: center;">
      <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor" class="bi bi-exclamation-triangle">
        <path d="M3.5 0a.5.5 0 0 1 .5.5V1h8V.5a.5.5 0 0 1 1 0V1h2a2 2 0 0 1 2 2V8a2 2 0 0 1 -2 2H3.5a.5.5 0 0 1 -.5.5V8a.5.5 0 0 1 0-.5Z"/>
      </svg>
      Data Consegna: <strong>{{data_consegna_toString}}</strong>
    </div>
  </p>
  <a class="btn btn-primary btn-block mb-3" href="/modifica_task/{{_id}}">Modifica</a>
  <a class="btn btn-danger btn-block mb-3" href="/cancella_task/{{_id}}">Cancella</a>
</div>
{{else}}
<h4>Non c'è nessun task da completare</h4>
{{/each}}
```



{ Esempio: Lista Task | View | 2/2 }

Home Info My Task ▾ Progetti ▾ Logout *pasqualematrone1994@gmail.com*

Pattern di ricerca (solo lettere)...

Nuovo Task

Task Assegnati e Creati

Task Assegnati


Task Creati

Lavagna Task

Storico Task

Lista dei Task

AMP-61e02ea3ce07e1d8e9b95753
Progetto: Amplifon HR
Sprint: 3
Descrizione: Implementare item assunzione nuovo dipendente
Stato: In Lavorazione
Effort: 10 giorni
Creato da: pasqualematrone1994@gmail.com
Assegnatario: angelomariaesposito1995@gmail.com
Commenti: Allegati inviati per email



35%

[Contatta il creatore del task](#)

☐ Data Consegna: **23/01/2022**

Modifica

Cancella

App MyTask

Database

Esempio: Lista Task

Miglioramenti

The background of the slide is a dark, blurred image of green light patterns, resembling the 'Matrix' digital rain effect. The patterns are vertical and slightly tilted, creating a sense of depth and movement. A semi-transparent white horizontal bar is positioned across the middle of the image, containing the text.

Possibili miglioramenti?

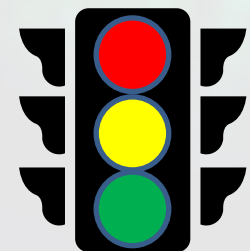


Possibili miglioramenti?

Tantissimi...

- **Possibili miglioramenti dell'App MyTask?**
 - **Workflow dei task**
 - Sistema di notifica via email
 - Inviare una email all'assegnatario del task se la scadenza è prossima
 - **Interfaccia utente**
 - Grafica meno «*spartana*» e più gradevole
 - Campi obbligatori/non obbligatori in base al contenuto di altri campi
 - Campi visibili/non visibili in base al contenuto di altri campi
 - «*Semaforo*» per evidenziare la prossimità della scadenza di un task

- **Possibili miglioramenti dell'App MyTask?**
 - **Ricerca testuale task**
 - Pattern che matcha su tutti i campi di un task
 - Numeri e stringhe
 - **Analisi dei task da parte di un manager**
 - Query più specifiche
 - Ad esempio, riportare i task consegnati in ritardo per assegnatario
 - **Registrazione utente anche tramite Facebook, Twitter, ecc**
 - Tramite la libreria Passport.js
 - ...



A photograph of a library interior. Tall wooden bookshelves filled with books line the walls. Several incandescent light bulbs hang from the ceiling, casting a warm glow. The perspective is looking down a long aisle between the shelves.

Link & Riferimenti

Link Progetto

Riferimenti Bibliografici

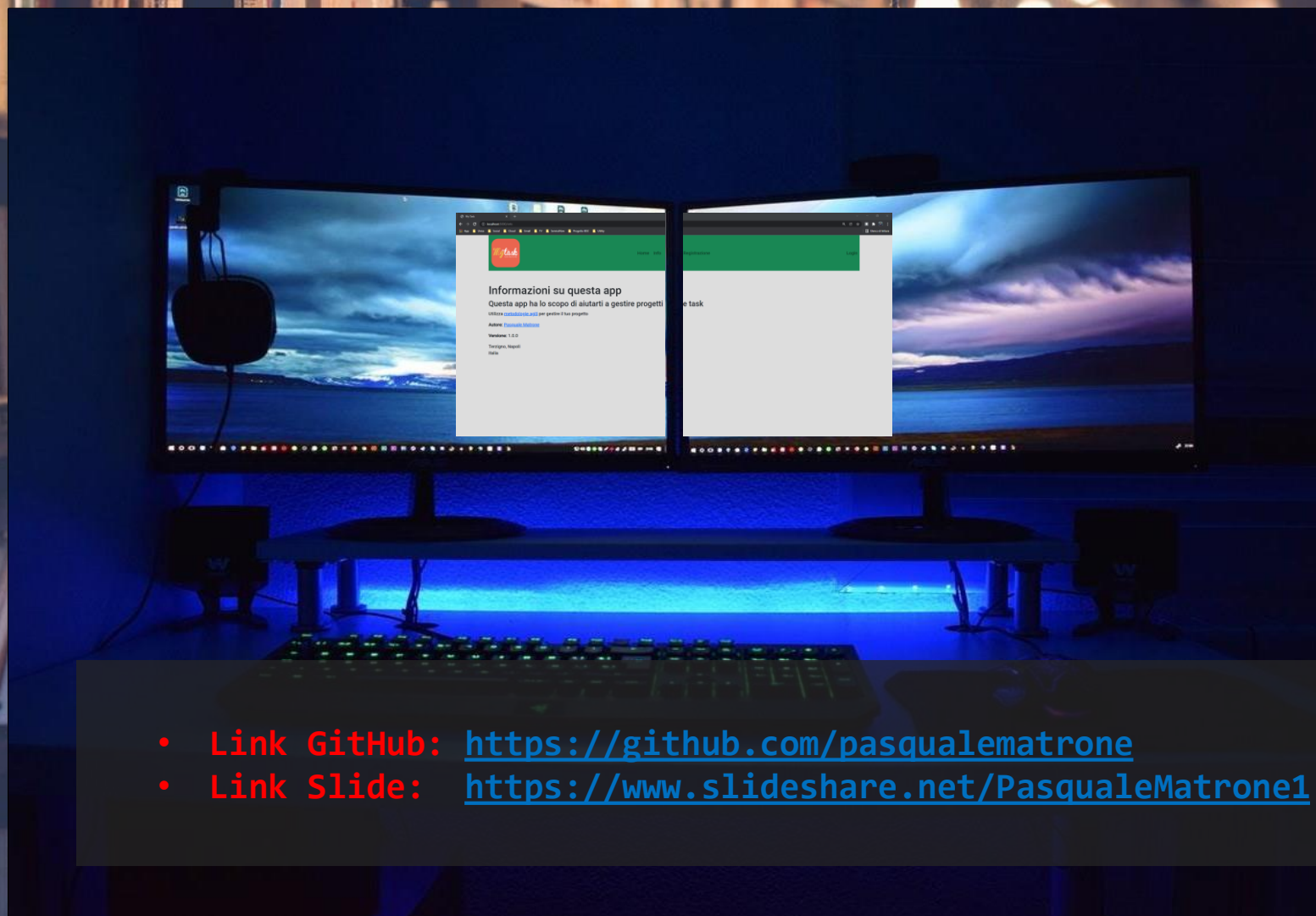
A photograph of a library interior. On the left, there are tall wooden bookshelves filled with books. On the right, several incandescent light bulbs hang from the ceiling, creating a warm, ambient glow. The background is slightly blurred, emphasizing the shelves and lights.

Link & Riferimenti

Link Progetto

Riferimenti Bibliografici

Link Progetto



- **Link GitHub:** <https://github.com/pasqualematrone>
- **Link Slide:** <https://www.slideshare.net/PasqualeMatrone1>

A photograph of a library interior. On the left, there are tall wooden bookshelves filled with books. On the right, several incandescent light bulbs hang from the ceiling, casting a warm glow. The background is slightly blurred, emphasizing the shelves and lights.

Link & Riferimenti

Link Progetto

Riferimenti Bibliografici

Riferimenti Bibliografici

- **Node.js®**
 - <https://nodejs.org/it/>
 - <https://it.wikipedia.org/wiki/Node.js>
- **MongoDB**
 - <https://www.mongodb.com/>
 - <https://it.wikipedia.org/wiki/MongoDB>
 - <https://db-engines.com/en/>
- **MongoDB Compass**
 - <https://www.mongodb.com/products/compass>
- **Npm**
 - <https://www.npmjs.com/>
 - [https://it.wikipedia.org/wiki/Npm \(software\)](https://it.wikipedia.org/wiki/Npm_(software))

Riferimenti Bibliografici

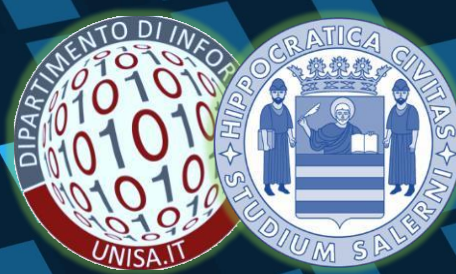
- **Mongoose.js**
 - <https://mongoosejs.com/>
 - <https://mongoosejs.com/docs/guide.html>
 - <https://itnext.io/performance-tips-for-mongodb-mongoose-190732a5d382>
 - [https://en.wikipedia.org/wiki/Mongoose \(MongoDB\)](https://en.wikipedia.org/wiki/Mongoose_(MongoDB))
- **Express.js**
 - <https://expressjs.com/it/>
 - <https://it.wikipedia.org/wiki/Express.js>
- **Express-session**
 - <https://www.npmjs.com/package/express-session>
- **Body-parser**
 - <https://www.npmjs.com/package/body-parser>

Riferimenti Bibliografici

- **Connect-flash**
 - <https://www.npmjs.com/package/connect-flash>
- **Passport**
 - <https://www.passportjs.org/>
- **Bcryptjs**
 - <https://www.npmjs.com/package/bcryptjs>
 - https://it.wikipedia.org/wiki/Funzione_crittografica_di_hash
- **Nodemon**
 - <https://www.npmjs.com/package/nodemon>
- **Handlebars**
 - <https://handlebarsjs.com/>

Riferimenti Bibliografici

- **Bootstrap**
 - <https://getbootstrap.com/>
- **JavaScript**
 - <https://developer.mozilla.org/en-US/docs/Web/JavaScript?retiredLocale=it>
- **jQuery**
 - <https://jquery.com/>
- **Stack Overflow**
 - <https://stackoverflow.com/>
 - [https://it.wikipedia.org/wiki/Stack Overflow \(sito\)](https://it.wikipedia.org/wiki/Stack_Overflow_(sito))



Università degli Studi di Salerno

Dipartimento di Informatica

```
Basi Dati 2
# echo Grazie per l'attenzione!
```

Pasquale Matrone