



MI

MANUALE

D'INSTALLAZIONE

ModuLink

Riferimento	NC08_MI_ver.1.0
Versione	1.0
Data	10/01/2026
Destinatario	Studenti di Ingegneria del Software 2025/26
Presentato da	Buzi Arjel, Carpentieri Daniele, Chikviladze Aleksandre, Cito Roberto.
Codice Gruppo	NC08
Approvato da	



## Revision History

Data	Versione	Descrizione	Autori
10/01/2026	0.1	Prima stesura	Arjel Buzi
11/01/2026	1.0	Stesura Definitiva e Revisione	Arjel Buzi



## Team members

---

Nome	Ruolo nel progetto	Acronimo	Informazioni di contatto
<b>Roberto Cito</b>	Team Member	RC	<a href="mailto:r.cito@studenti.unisa.it">r.cito@studenti.unisa.it</a>
<b>Daniele Carpentieri</b>	Team Member	DC	<a href="mailto:d.carpentieri8@studenti.unisa.it">d.carpentieri8@studenti.unisa.it</a>
<b>Aleksandre Chikviladze</b>	Team Member	AC	<a href="mailto:a.chikviladze@studenti.unisa.it">a.chikviladze@studenti.unisa.it</a>
<b>Arjel Buzi</b>	Team Member	AB	<a href="mailto:a.buzi@studenti.unisa.it">a.buzi@studenti.unisa.it</a>



## Sommario

Revision History .....	2
Team members .....	3
1. INTRODUZIONE.....	5
2. DEPLOYMENT CON JAVA .....	6
2.1 Requisiti .....	6
2.2 Variabili ambientali .....	6
2.3 Scaricare il progetto .....	7
2.4 Compilazione ed Esecuzione .....	8
3. DEPLOYMENT CON DOCKER.....	9
3.1 Docker Desktop.....	9
3.2 Docker da terminale .....	11
4. UTILIZZO .....	11



# 1. INTRODUZIONE

L'obiettivo di questo documento è illustrare in modo chiaro e completo le procedure necessarie per accedere e utilizzare la nostra web-app. Ogni fase del processo sarà descritta con attenzione, così da fornire all'utente tutte le informazioni utili per orientarsi fin dal primo utilizzo. Le varie sezioni saranno accompagnate da screenshot esplicativi che consentiranno di visualizzare le operazioni da eseguire, riducendo al minimo eventuali ambiguità e agevolando la comprensione del flusso di lavoro. Questo manuale si propone quindi come guida di riferimento sia per i nuovi utenti sia per coloro che desiderano approfondire funzionalità già note.



## 2. DEPLOYMENT CON JAVA

Questa guida illustra la procedura per configurare, compilare e avviare l'applicazione.

### 2.1 Requisiti

Per il corretto funzionamento dell'applicativo, l'ambiente di hosting deve disporre dei seguenti componenti:

- **Java SDK (JDK) 17 o superiore:** Necessario per il runtime e la compilazione.
- **Database MySQL:** Una istanza attiva (locale o remota).
- **Server SMTP/IMAP/POP:** Un servizio di posta elettronica attivo per la gestione delle notifiche e delle comunicazioni.
- **Apache Maven:** Per la gestione delle dipendenze e la build del progetto.
- **Git** (Opzionale ma consigliato): Per il versionamento e il recupero del codice.

### 2.2 Variabili ambientali

L'applicazione utilizza variabili d'ambiente per proteggere i dati sensibili e permettere la portabilità tra diversi ambienti (sviluppo, staging, produzione). Devono essere configurate sul sistema ospite prima dell'avvio:

Nome	Descrizione	Esempio / Note
<b>DB_URL</b>	URL di connessione JDBC.	<code>jdbc:mysql://localhost:3306/modulink</code>
<b>DB_USER</b>	Username del database.	<code>root</code>
<b>DB_PASS</b>	Password del database.	<code>tua_password</code>
<b>MAIL</b>	Indirizzo email mittente.	<code>noreply@tuodominio.it</code>
<b>PASSWORD_MAIL</b>	Password o Token applicativo.	<code>password_smtp</code>

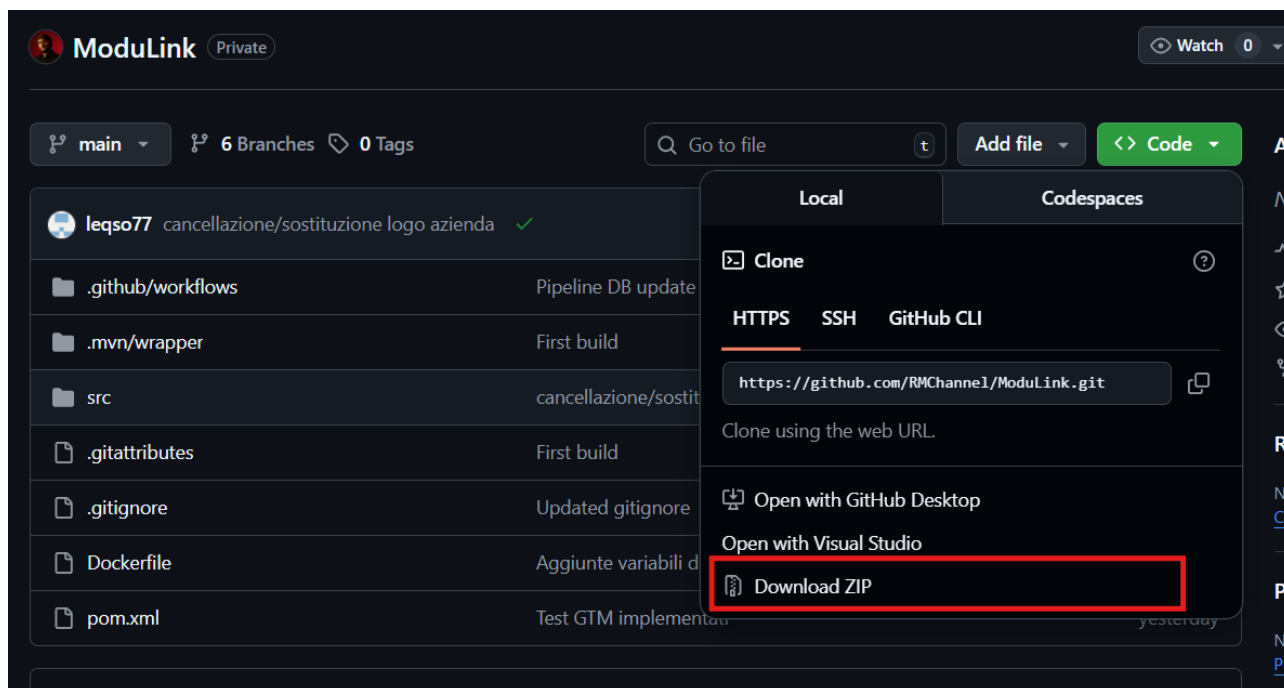
## 2.3 Scaricare il progetto

È possibile ottenere il codice sorgente in due modalità:

- **Via Git (Scelta consigliata):** Aprire il terminale e clonare il repository:

```
C:\>git clone https://github.com/RMChannel/ModuLink
```

- **Download Manuale:** Recarsi sulla pagina GitHub del progetto e scaricare il pacchetto `.zip`, estraendolo successivamente in una cartella locale.



Dopo aver scaricato il zip, estrarlo nella cartella desiderata.



## 2.4 Compilazione ed Esecuzione

Seguire questi passaggi per trasformare il codice sorgente in un file eseguibile.

Posizionarsi tramite terminale nella cartella radice del progetto (dove è presente il file pom.xml) ed eseguire il comando:

```
mvn clean package
```

**Cosa succede durante questo comando?**

- **clean:** Rimuove eventuali file di vecchie compilazioni.
- **package:** Scarica le dipendenze, esegue i test e genera un file eseguibile nella cartella target/.

Una volta terminata la compilazione, verrà generato un file con estensione .jar (es. modulink-0.0.1-SNAPSHOT.jar). Per avviare il server, eseguire:

```
java -jar target/nome-del-file.jar
```

L'applicazione Spring Boot si avvierà e sarà raggiungibile di default all'indirizzo <http://localhost:80>.

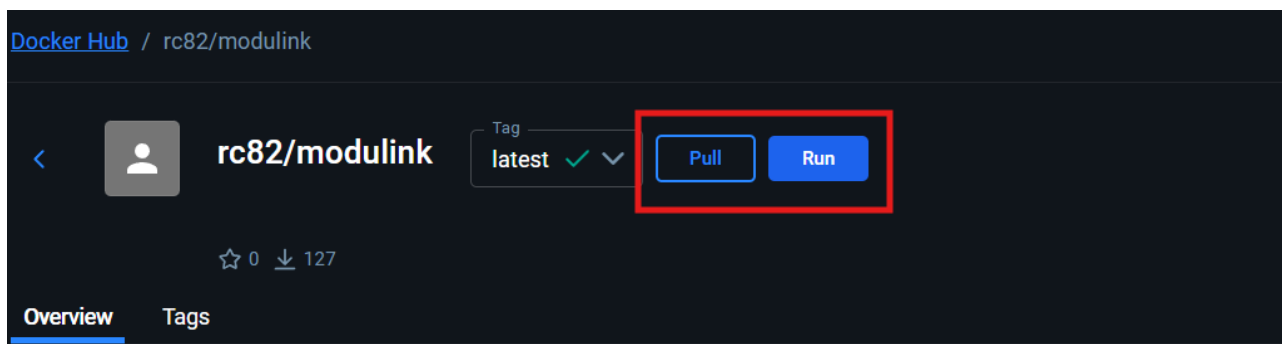


## 3. DEPLOYMENT CON DOCKER

Il progetto dispone di una pipeline che converte automaticamente l'ultima versione presente sulla branch main in un'immagine Docker pubblica su Docker Hub. In questo modo è possibile scaricare direttamente l'immagine, semplificando l'installazione e l'esecuzione del servizio. Il database, invece, non è incluso nell'immagine e va configurato separatamente.

### 3.1 Docker Desktop

Accedere a Docker Hub e scaricare l'immagine tramite il pull dal Docker Hub.






Laurea Triennale in informatica - Università di Salerno  
Corso di *Ingegneria del Software* - Prof. Gravino

Successivamente, nella sezione delle immagini, eseguire il run impostando tutte le variabili d'ambiente necessarie al corretto funzionamento, come descritto nella sezione 2.2

<input type="checkbox"/>	Name	Tag	Image ID	Created	Size	Actions
<input type="checkbox"/>	mysql	latest	57c892ddeb17	6 days ago	1.28 GB	
<input type="checkbox"/>	postgres	15-alpine	b3968e348b48	1 month ago	392.26 MB	
<input type="checkbox"/>	postgres	latest	5ec39c188013	2 months ago	643.03 MB	
<input type="checkbox"/>	rc82/modulink	latest	942cbf57769d	29 minutes ag	406.07 MB	

**Run a new container**  
rc82/modulink:latest

**Optional settings**

A random name is generated if you do not provide one.

**Ports**

Enter "0" to assign randomly generated host ports.

**Volumes**

**Environment variables**

<input type="text" value="Variable"/>	<input type="text" value="Value"/>	
<input type="text" value="Variable"/>	<input type="text" value="Value"/>	
<input type="text" value="Variable"/>	<input type="text" value="Value"/>	
<input type="text" value="Variable"/>	<input type="text" value="Value"/>	



## 3.2 Docker da terminale

Se non si vuole usare Docker Desktop, puoi usare anche i comandi da terminale nella stessa metodologia illustrata prima.

```
docker pull rc82/modulink
```

Inserire le variabili dopo il run:

```
docker run -it --rm \  
-e DB_URL="jdbc:mysql://localhost:3306/modulink" \  
-e DB_USER="root" \  
-e DB_PASS="tua_password" \  
-e MAIL="noreply@tuodominio.it" \  
-e PASSWORD_MAIL="password_smtp" \  
-p 80:8080 \  
rc82/modulink
```

## 4. UTILIZZO

Allora il server genera delle aziende di base per testare le sue funzionalità e genera l'account admin, che è un'altra azienda che ha accesso a modulo speciali, per accedere a questa sezione.

Utilizzare l'account di base:

EMAIL	PASSWORD
<a href="mailto:admin@modulink.run.place">admin@modulink.run.place</a>	Admin1234@



Laurea Triennale in informatica - Università di Salerno  
Corso di *Ingegneria del Software* - Prof. Gravino



Accedi al tuo account

Email

admin@modulink.run.place

Password

••••••••

Accedi

Non hai un account? [Registrati ora](#)

Password dimenticata? [Clicca qui](#)

Cambiare la password una volta, loggati per sicurezza.