



# ZiFiorino - Guida Utente

Gianluca Gemini

GitHub Repo: <https://github.com/yolly98/ZiFiorino>

# Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
<b>2</b>	<b>Login</b>	<b>3</b>
<b>3</b>	<b>Registrazione</b>	<b>3</b>
<b>4</b>	<b>Home</b>	<b>4</b>
<b>5</b>	<b>Come accedere ai dati</b>	<b>5</b>
<b>6</b>	<b>Aggiungere un nuovo elemento</b>	<b>5</b>
<b>7</b>	<b>Backup</b>	<b>6</b>
<b>8</b>	<b>Cambiare password di accesso</b>	<b>7</b>
<b>9</b>	<b>Installare ZiFiorino sul proprio server casalingo</b>	<b>8</b>
9.1	Prerequisiti . . . . .	8
9.2	Installazione Rapida con Makefile e Dockerfile . . . . .	8
9.3	Installazione Manuale . . . . .	8
9.3.1	Installazione di Apache . . . . .	8
9.3.2	Installazione di MySql . . . . .	9
9.3.3	Installazione di ZiFiorino . . . . .	9
9.4	Problemi . . . . .	10
<b>10</b>	<b>Note tecniche sulla sicurezza</b>	<b>12</b>
10.1	Autenticazione . . . . .	12
10.2	Archiviazione Sicura . . . . .	12
10.3	Comunicazione Sicura . . . . .	12

# 1 Introduzione

ZiFiorino è un applicazione web che ha lo scopo di archiviare in maniera sicura i dati di accesso (come username e password) relativi ai siti e applicazioni in cui si ha un account. Ci sono molti strumenti per fare questo, dalle app ai browser stessi, tuttavia farne uso significa fidarsi di chi ci sta archiviando e gestendo i dati.

Con ZiFiorino è possibile avere un' app sicura da installare sui propri computer/server casalinghi così da non dover affidare a nessuno i propri dati sensibili.

Inoltre è possibile provare ZiFiorino all'indirizzo <https://zifiorino.altervista.org> con HTTPS (sicurezza nella comunicazione).

Il codice sorgente dell'app e la build sono scaricabili dal repository GitHub in fondo alla prima pagina. Alla fine del documento sarà presente una guida all'installazione e delle brevi note tecniche sulla sicurezza dell'app.

## 2 Login



**LOGIN**  
Questa è la schermata di login di ZiFiorino. Inserendo le proprie credenziali e premendo sul tasto ACCEDI si potrà accedere all'app. Premendo sul tasto REGISTRATI verrà aperta una finestra in cui potrai registrarti al servizio.

  
**ZiFiorino**

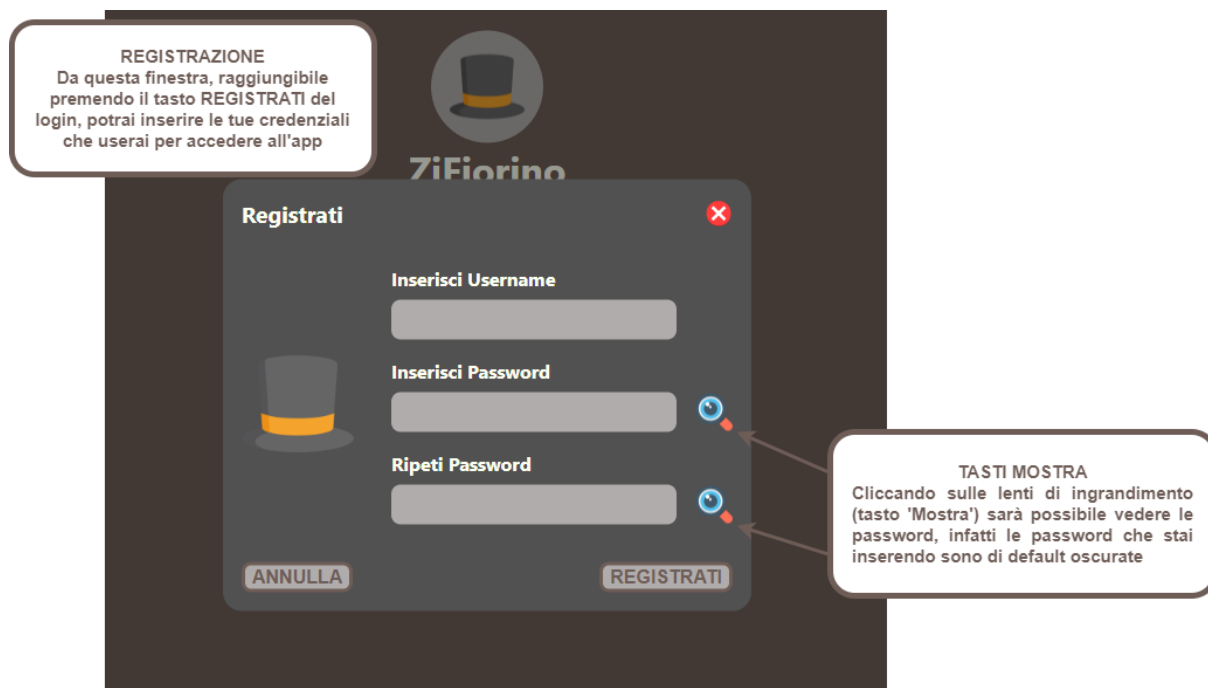
Nome Utente

Password


ACCEDI

REGISTRATI

## 3 Registrazione



**REGISTRAZIONE**  
Da questa finestra, raggiungibile premendo il tasto REGISTRATI del login, potrai inserire le tue credenziali che userai per accedere all'app

  
**ZiFiorino**

**Registrati** ✕

Inserisci Username

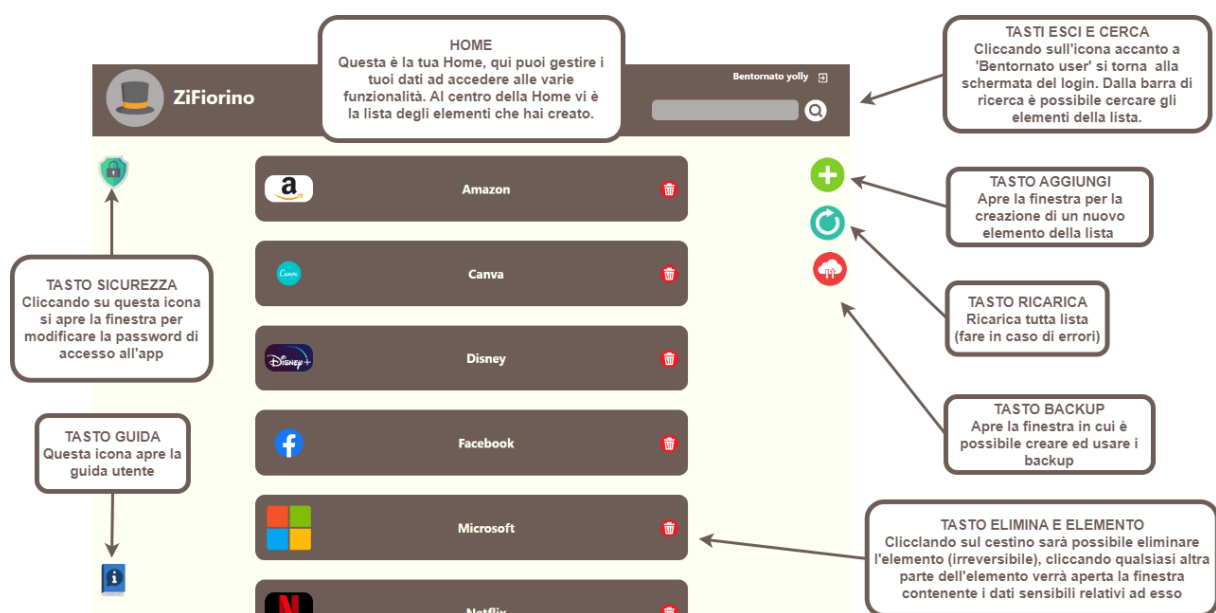
Inserisci Password

Ripeti Password

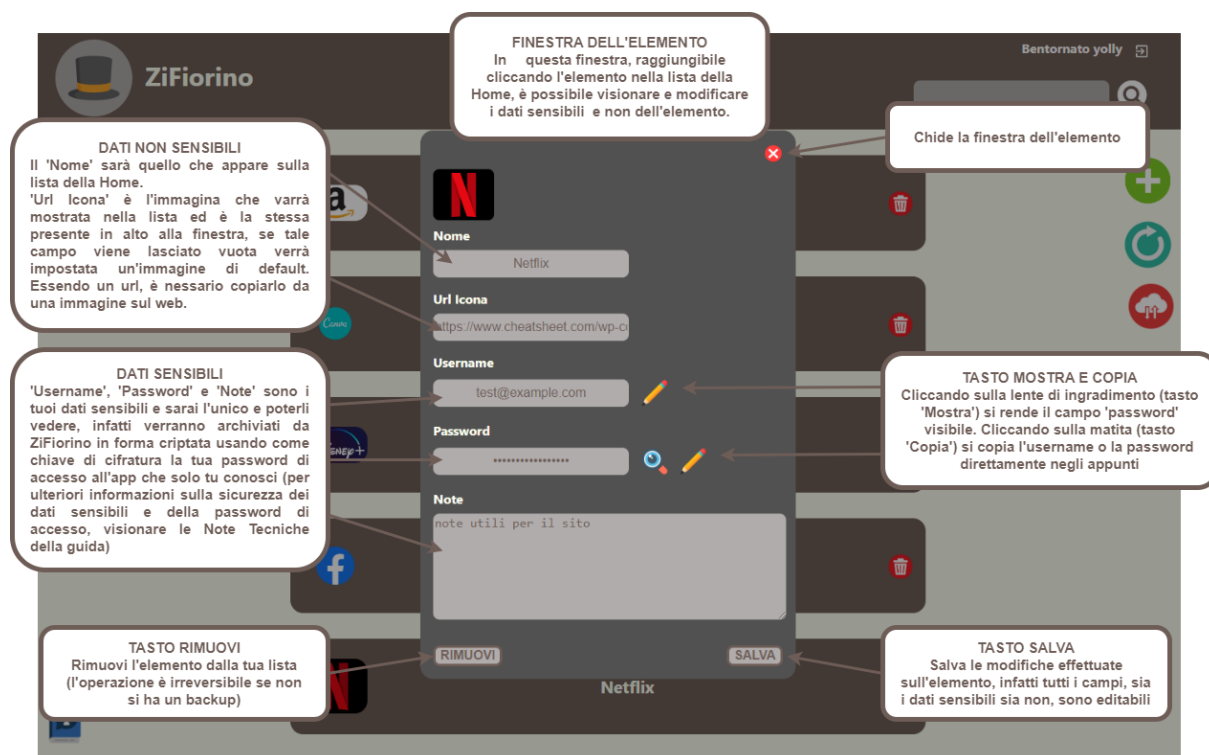
ANNULLA REGISTRATI

**TASTI MOSTRA**  
Cliccando sulle lenti di ingrandimento (tasto 'Mostra') sarà possibile vedere le password, infatti le password che stai inserendo sono di default oscurate

## 4 Home



## 5 Come accedere ai dati



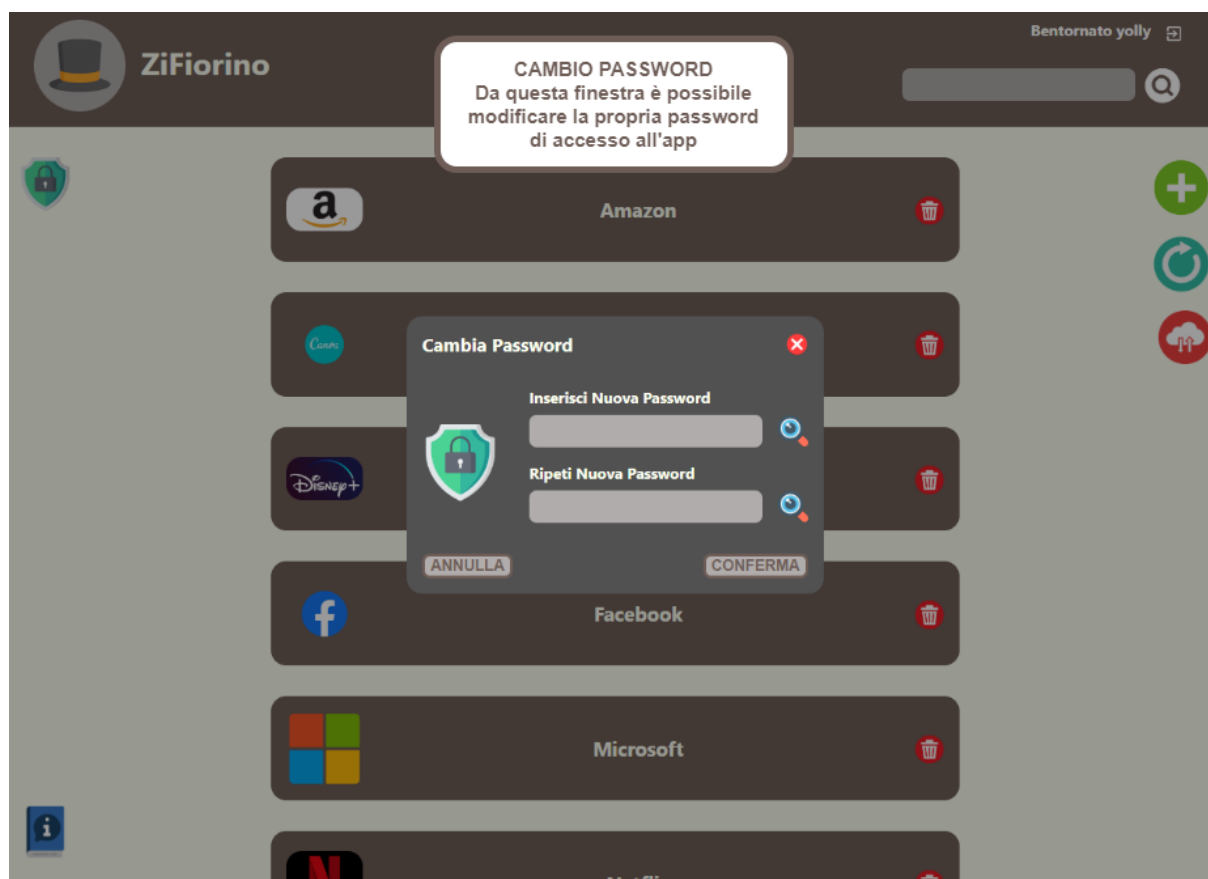
## 6 Aggiungere un nuovo elemento



## 7 Backup



## 8 Cambiare password di accesso





## 9 Installare ZiFiorino sul proprio server casalingo

In questa breve guida verrà mostrato come usare ZiFiorino su un proprio PC che funge da semplice server casalingo. In particolare verrà ricostruito l'ambiente per cui quest'applicazione è stata progettata, verrà installato su un sistema Ubuntu Linux sfruttando Apache Server e Mysql ciascuno su un container di docker.

### 9.1 Prerequisiti

- Sistema Linux o Windows WSL
- Docker

### 9.2 Installazione Rapida con Makefile e Dockerfile

Per installare ZiFiorino in maniera rapida basta seguire i seguenti passaggi:

- Scaricare il zip del repository di ZiFiorino
- Decomprimere il zip
- Aprire il terminale posizionandosi all'interno della cartella appena decompressa
- Eseguire i seguenti comandi:

```
make build  
make run
```

- Aprire un qualsiasi browser e andare all'indirizzo 'localhost'

Se si vuole eliminare ZiFiorino è sufficiente eseguire il comando 'make clean'.

### 9.3 Installazione Manuale

Per prima cosa è necessario creare una rete locale su docker così da permettere la comunicazione tra i container, di seguito vi è il comando per creare una rete con indirizzi ip 172.20.0.0/16 di nome 'zf-net'.

```
sudo docker network create --driver bridge --subnet 172.20.0.0/16 zf-net
```

#### 9.3.1 Installazione di Apache

Per installare Apache Server basta eseguire il seguente comando:

```
sudo docker run -d --name zf-apache-php --net zf-net --ip 172.20.0.10 -p 80:80  
-v /myLocalPath:/var/www/html php:7.4-apache
```

Porre attenzione ai seguenti argomenti del comando:

- `-name`: indica il nome con cui vogliamo chiamare il container (in questo caso 'apache-php')
- `-net`: indica il nome della rete di cui il container deve far parte (settare la rete creata in precedenza)
- `-ip` e `-p`: assegna un indirizzo ip statico della rete 'zf-net' e le porte di accesso
- `-v`: indica dove montare la cartella da cui il server fornirà l'accesso, il path da modificare è '/myLocalPath', questo permette di accedere facilmente alle risorse del server. Se si omette questo argomento, per accedere ai file del server, sarà necessario entrare nella shell del container e andare in '/var/www/html'

Fatto ciò bisogna installare l'estensione PHP per MySQL, quindi bisogna accedere alla shell del container:

```
sudo docker exec -it zf-apache-php /bin/bash
```

Poi bisogna inviare i seguenti comandi:

```
# docker-php-ext-install mysqli
# docker-php-ext-enable mysqli
```

### 9.3.2 Installazione di MySQL

Eeguire il seguente comando:

```
sudo docker run --name zf-mysql --net zf-net --ip 172.20.0.11 -p 3306:3306
-v mysql_volume:/var/lib/mysql/ -d -e "MYSQL_ROOT_PASSWORD=password" mysql
```

Per questo comando l'unica cosa di cui avere cura è di specificare la rete 'zf-net', indirizzo ip e porte.

### 9.3.3 Installazione di ZiFiorino

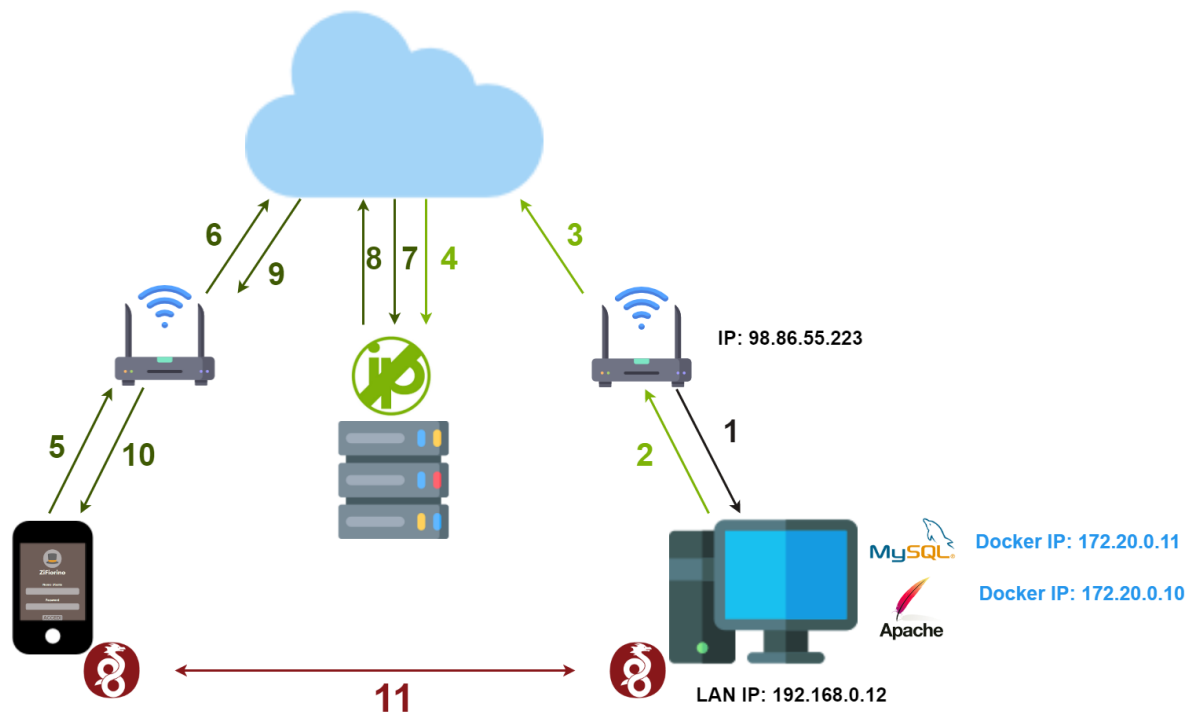
Per installare e utilizzare ZiFiorino è sufficiente seguire i seguenti passaggi:

- Avviare i container apache e mysql
- Scaricare la release dal repository
- Prendere il contenuto della cartella 'frontend-build' della release e copiarla nella cartella su cui è stato montato Apache Server
- Impostare il database eseguendo lo script 'db\_builder.sql' presente sul repository (per fare questo è possibile usare un qualsiasi software come MySQLWorkbench )
- Copiare la cartella 'backend' della release nella cartella principale di Apache Server

- Settare l'indirizzo ip di MySql nel campo \$IP\_ADDR (172.20.0.11 nell'esempio sopra) nel file 'backend/config.php' (dovrebbe essere già settato correttamente).
- Settare l'indirizzo ip di loopback nel file 'build/static/config.json' ('http://127.0.0.1:80/backend/requests/' dovrebbe già essere settato).
- Apri il browser e vai all'indirizzo 'http://localhost'

## 9.4 Problemi

- La comunicazione http non è sicura perciò NON usare l'app se non dalla stessa macchina su cui è hostato il backend di ZiFiorino.
- Se vuoi accedere a ZiFiorino dall'esterno senza configurare l'https potresti usare un tunneling VPN come Wireguard, infatti configurando Wireguard sul tuo sistema Linux come server, e configurando Wireguard come client sul tuo smartphone o altro computer, potrai accedere alla tua macchina da qualsiasi parte del mondo semplicemente attivando la VPN, inoltre essa stessa ti garantirà la sicurezza nella comunicazione che avresti avuto con l'https.
- Per configurare un server VPN come Wireguard dovresti aprire la sua porta tramite il firewall (sia del router sia della macchina Linux) e configurare un indirizzo ip statico sia del router sia del tuo sistema Linux.
- Per configurare l'indirizzo ip statico della tua macchina Linux dovrai farlo dalle impostazioni del tuo router (preferibile) o dalla macchina stessa. Per l'indirizzo ip statico del router invece, puoi aggirare il problema sfruttando il servizio gratuito di Nolp.



Non seguirà una guida su come risolvere i problemi sopracitati, ma solo dei link utili.

- <https://ubuntu.com/server/docs/security-firewall>
- <https://www.wireguard.com/>
- <https://www.noip.com/it-IT/>

## 10 Note tecniche sulla sicurezza

### 10.1 Autenticazione

L'autenticazione avviene in fase di login la prima volta tramite username e password. Al momento della registrazione la password viene salvata sul database MySQL facendone lo SHA 256 con salt, in questo modo sarà possibile verificare la correttezza della password al login senza mantenerla in chiaro, infatti conoscere l'hash non permette di ricavare la password da cui è stato calcolato. Alla fase di login, dopo la verifica della correttezza della password di accesso, viene generato un JWT con la seguente struttura:

```
JWT = {  
    "username" = "username",  
    "password" = sha256("password"),  
    "exp" : expiration_time  
}
```

Il JWT viene firmato dal server con la chiave segreta 'JWT\_KEY' presente nel file 'backend/config.php', poi il JWT viene inviato al client che lo dovrà inserire dentro ad ogni richiesta successiva al login fino al logout (sessione). Infatti quando il server riceve una richiesta dal cliente va a verificare il JWT e se la verifica va a buon fine può estrarre le informazioni al suo interno e riconoscere il client.

### 10.2 Archiviazione Sicura

Lo SHA 256 (senza salt) della password di accesso viene usato come chiave per criptare con l' AES 256 tutte le informazioni sensibili, inoltre viene generato un IV (initialization vector) per ogni elemento da criptare, tale IV viene salvato in chiaro nel database per permettere di decriptare. Tutte le informazioni sensibili archiviate nel database o nei backup sono quindi protette da crittografia simmetrica.

### 10.3 Comunicazione Sicura

Come già detto in precedenza, la comunicazione dovrebbe avvenire sotto HTTPS, fortunatamente se si intende sfruttare un servizio di hosting fornito da terzi, l' HTTPS viene garantito senza costi aggiuntivi, invece in caso si voglia usare un proprio server casalingo bisognerebbe munirsi di un certificato SSL valido o optare per una soluzione con VPN.