

# **Manuale di installazione**

Informatica III - Modulo di Progettazione e Algoritmi

Michele Beretta - 1054365

Bianca Crippa - 1053356

Pape Alpha Toure - 1053327

A.A. 2020/2021



# Indice

<b>1</b>	<b>Installazione</b>	<b>1</b>
1.1	Installazione su macOS . . . . .	1
1.2	Installazione su Linux . . . . .	3
1.3	Installazione su Windows . . . . .	3
<b>2</b>	<b>Compilazione del codice sorgente</b>	<b>5</b>
2.1	Server back-end in Python . . . . .	5
2.1.1	Prerequisiti . . . . .	5
2.1.2	Installazione delle dipendenze . . . . .	5
2.1.3	Esecuzione . . . . .	6
2.1.4	Riassunto . . . . .	6
2.2	Front-end in JavaScript . . . . .	7
2.2.1	Prerequisiti . . . . .	7
2.2.2	Installazione delle dipendenze . . . . .	7
2.2.3	Variabili di ambiente . . . . .	7
2.2.4	Esecuzione e compilazione del sito . . . . .	7
2.2.5	Esecuzione e compilazione della app . . . . .	8

2.2.6	Riassunto	8
-------	-----------	---

# Capitolo 1

## Installazione

Il progetto è disponibile sia in forma web, all'indirizzo `aiticketing.xyz`, oppure sotto forma di app installabile sul proprio sistema operativo.

Di seguito sono elencati i procedimenti per l'installazione della app sui vari sistemi operativi.

### 1.1 Installazione su macOS

Per *macOS* è disponibile un file `.dmg`. I passi da seguire sono i seguenti:

1. Montare il `.dmg` facendo doppio click su di esso;
2. Comparirà un disco sul desktop, e dopo averlo aperto si vedrà una finestra come in figura 1.1;
3. Trascinare l'icona di **AI Ticketing** nella cartella delle applicazioni;
4. La app è ora installata e avviabile, come si vede in figura 1.2.

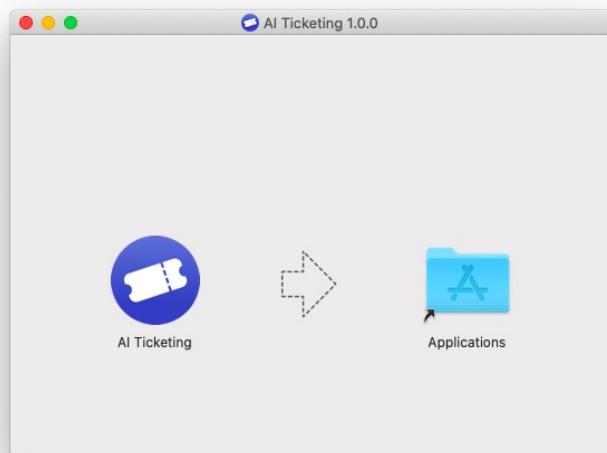


Figura 1.1: Installazione su macOS (1)

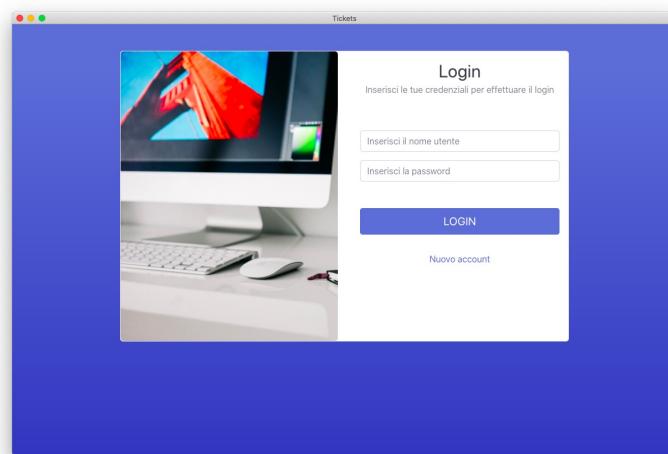
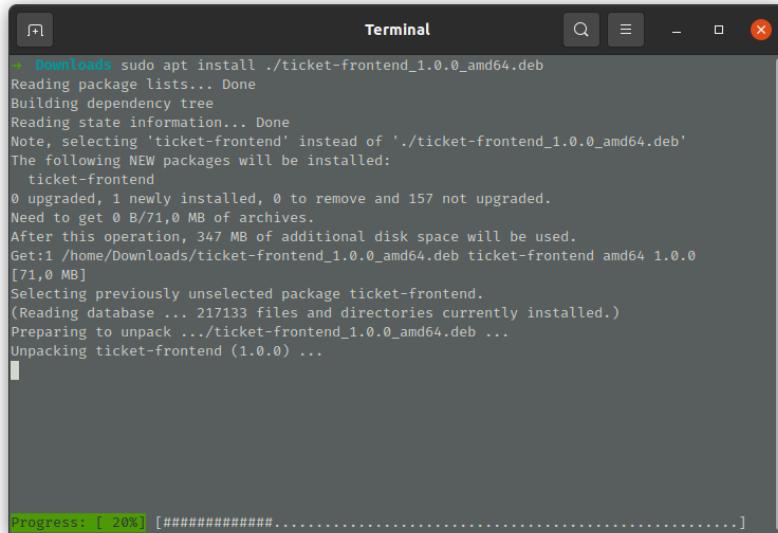


Figura 1.2: Installazione su macOS (2)

## 1.2 Installazione su Linux

È disponibile un pacchetto .deb per l'installazione sulle distro Debian-based.  
I passi da seguire sono i seguenti:

1. Aprire il terminale nella cartella dove si trova il .deb;
2. Eseguire il comando `sudo apt install ./ticket-frontend_1.0.0_amd64.deb`;
3. Inserire, se necessario, la password dell'utente;
4. Una volta terminato il processo visibile in figura 1.3, la applicazione sarà installata ed utilizzabile.



```
Terminal
Downloads: sudo apt install ./ticket-frontend_1.0.0_amd64.deb
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'ticket-frontend' instead of './ticket-frontend_1.0.0_amd64.deb'
The following NEW packages will be installed:
  ticket-frontend
0 upgraded, 1 newly installed, 0 to remove and 157 not upgraded.
Need to get 0 B/71,0 MB of archives.
After this operation, 347 MB of additional disk space will be used.
Get:1 /home/Downloads/ticket-frontend_1.0.0_amd64.deb ticket-frontend amd64 1.0.0
[71,0 MB]
Selecting previously unselected package ticket-frontend.
(Reading database ... 217133 files and directories currently installed.)
Preparing to unpack .../ticket-frontend_1.0.0_amd64.deb ...
Unpacking ticket-frontend (1.0.0) ...
Progress: [ 20%] [#####.....]
```

Figura 1.3: Installazione su Linux

## 1.3 Installazione su Windows

Per l'installazione su Windows è disponibile un *installer*, che si chiama **AI Ticketing 1.0.0.exe**. I passi da seguire sono i seguenti:

1. Aprire l'installer, si avvierà una finestra come in figura 1.4;
2. Una volta terminato l'installer si aprirà la app, come visibile in figura 1.5;
3. Sul desktop sarà presente il collegamento, e la app è disponibile cercando nel menù start.

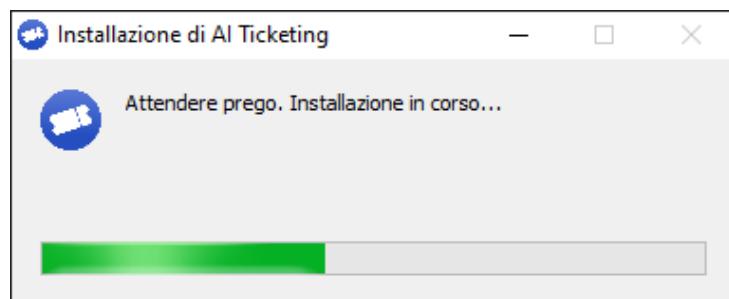


Figura 1.4: Installazione su Windows (1)

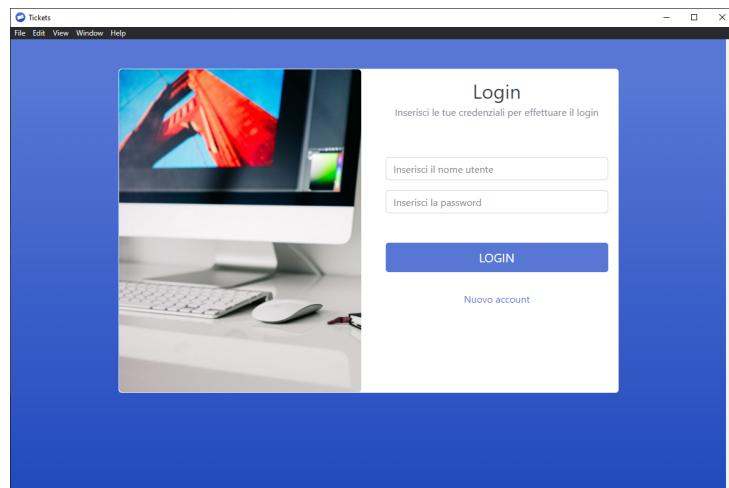


Figura 1.5: Installazione su Windows (2)

# Capitolo 2

## Compilazione del codice sorgente

### 2.1 Server back-end in Python

#### 2.1.1 Prerequisiti

È necessario avere *Python 3* e *pip* installati. Tutti i comandi di seguito saranno effettuati su una shell a propria scelta (o il *Command Prompt* su Windows).

#### 2.1.2 Installazione delle dipendenze

Il codice Python necessita di alcune dipendenze per il corretto funzionamento. Al fine di isolare il codice ed il suo environment da altri possibilmente esistenti sul sistema operativo, è necessario attivare un *virtual environment* con il comando `python -m venv venv1`.

A questo punto, si può attivare il virtual environment con il comando

- `source venv/bin/activate` per i sistemi Unix e Unix-like;
- `venv\bin\activate` per Windows.

---

<sup>1</sup>In alcuni casi potrebbe essere necessario usare `python3`

Ora è possibile installare le dipendenze, elencate nel file `requirements.txt`, tramite il comando `pip install -r requirements.txt`.

### 2.1.3 Esecuzione

Dopo aver installato le dipendenze ed essersi posti nella cartella `back-end`, è possibile eseguire il server. Una volta attivato il virtual environment, la prima volta bisogna eseguire due comandi per inizializzare il database ed il superuser:

- `python manage.py migrate`
- `python manage.py createsuperuser`

Per avviare il server si usa quindi il comando `python manage.py runserver`, che di default avvierà un processo in ascolto all'indirizzo `localhost:8080`.

### 2.1.4 Riassunto

I comandi da eseguire sono, supponendo di essere inizialmente nella root del progetto

1. `cd back-end`
2. `python -m venv venv`
3. `source venv/bin/activate`
4. `pip install -r requirements.txt`
5. Solo al primo avvio
  - `python manage.py migrate`
  - `python manage.py createsuperuser`
6. `python manage.py runserver`

## 2.2 Front-end in JavaScript

### 2.2.1 Prerequisiti

È necessario avere *node* e *npm* installati. Inoltre, seppur non essenziale, è consigliato l'uso di *yarn* per la gestione delle dipendenze. Tutti i comandi di seguito saranno effettuati su una shell a propria scelta (o il *Command Prompt* su Windows).

Se si vuole usare `npm` il comando per l'installazione delle dipendenze è `npm install`, mentre per gli altri comandi basta sostituire `yarn` con `npm run`.

### 2.2.2 Installazione delle dipendenze

Le dipendenze sono gestite tramite *node*. Per installarle bisogna porsi nella cartella `front-end` ed eseguire il comando `yarn install`.

### 2.2.3 Variabili di ambiente

Per il funzionamento del sito è necessario indicare qual è il server a cui inoltrare le richieste di rete. Questo viene fatto creando un file `secrets.js` nella cartella `src` e ponendoci il seguente contenuto.

```
global.SERVER_ADDRESS = 'https://address.totheserver.com';
```

### 2.2.4 Esecuzione e compilazione del sito

Una volta installate le dipendenze, è possibile eseguire i seguenti comandi relativi al sito:

- `yarn start` per avere un server di sviluppo all'indirizzo `localhost:3000` e che si aggiorna automaticamente ad ogni cambiamento nei file;

- `yarn build` per produrre una versione compilata ed ottimizzata per la produzione

### 2.2.5 Esecuzione e compilazione della app

Il progetto dà anche la possibilità di compilare una app con *Electron*. Notare che questo metodo è pensato unicamente per la produzione e non per lo sviluppo, quindi se si desidera modificare gli indirizzi del server essi vanno cambiati sia in `main.js` sia in `electron/menu.js`.

Per effettuare una compilazione con Electron è necessario eseguire i seguenti due comandi in serie: `yarn build` e `yarn electron-build`.

Prestare attenzione alla compilazione con Electron: nel caso il sistema operativo utilizzato non sia macOS, è possibile che non sia abilitata la compilazione della app per macOS. In tal caso, basta rimuovere il flag `-m` dallo script `electron` nel file `front-end/package.json`.

Questa compilazione produrrà i relativi eseguibili, installer e pacchetti nella cartella `dist`.

### 2.2.6 Riassunto

Di seguito sono riassunti i comandi principali per l'esecuzione e compilazione:

1. `cd front-end`
2. `yarn install`
3. `echo "global.SERVER_ADDRESS = 'http://server.address.com'" > src/secrets.js`
4. Per far partire il server
  - `yarn start`
5. Per compilare per la produzione

- `yarn build`
6. Per compilare la app in Electron
- `yarn build`
  - `yarn electron-build`