

Dustbin IoT



2020

“Tecnico Superiore per Smart City & Clean Energy Management”

Ariel Montes Nogueira • info@montesariel.com

INDICE

Dustbin IoT	1
Descrizione	1
Utilizzo	1
Caratteristiche tecniche	1
L'applicazione	2
L'interoperabilità	6
Altre informazioni	7
Analisi del progetto	8
Bibliografia	9

DUSTBIN IOT

Descrizione

Sistema intelligente per la gestione della raccolta dei rifiuti di una Smart City.

Costituito da una rete di "nodi sensore" (ovvero dispositivi che incorporano un sensore di rilevamento) posizionati sotto ogni cestino rifiuti della città che comunicano in rete attraverso appositi gateway, fornendo dati in tempo reale a un server. I dati sono inoltre memorizzati in un database relazionale con diversi livelli di autorizzazione e possono essere consultati H24 tramite un'interfaccia web. É inoltre previsto il controllo remoto dei suddetti nodi.

Utilizzo

L'applicazione è accessibile online [1] 24/7 e gli aggiornamenti possono essere eseguiti senza interrompere il servizio.

L'amministratore può aggiungere, modificare o eliminare dall'applicazione le istanze che rappresentano: ordini, contenitori, operatori e accessi. Ogni utente vede i propri ordini e il livello di riempimento dei cestini da raccogliere. Gli ordini scaduti non verranno visualizzati dagli operatori.

Caratteristiche tecniche

Applicazione web scalabile (caratteristica che consente di aggiungere nuove funzionalità al programma) scritta in Python utilizzando Django e distribuita sulla piattaforma Heroku, come database utilizza PostgreSQL appoggiandosi all'ORM di Django. Per il Frontend invece utilizza JavaScript, HTML e CSS.

L'applicazione

Di seguito, viene dettagliata la funzionalità dell'applicazione web utilizzando le immagini tratte da essa.

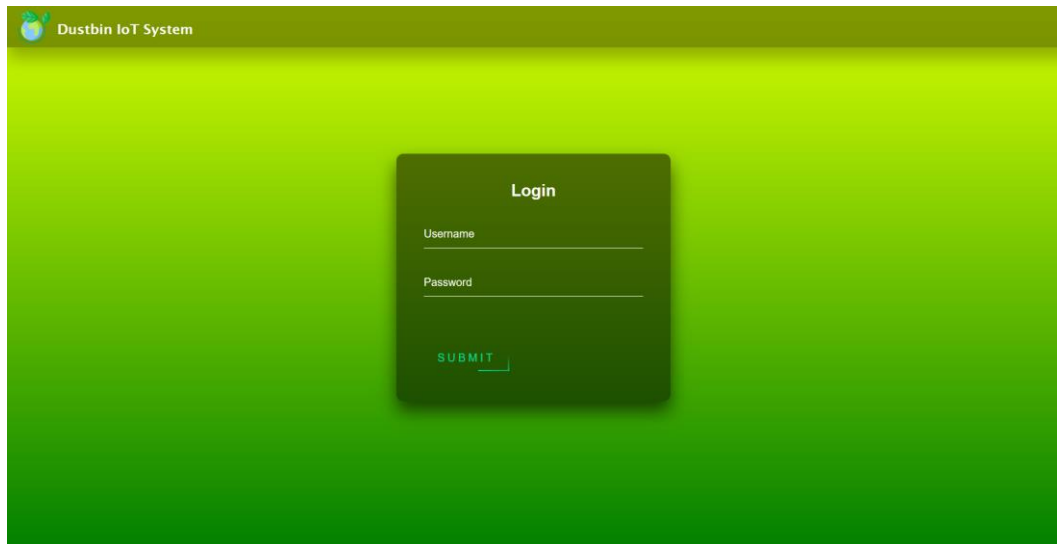


Figura 1 (Pagina di accesso al sistema)

Dalla figura 1 è possibile notare come in questa pagina sia gestito l'accesso come operatore o amministratore. Ciò che verrà visualizzato nella schermata successiva dipende dai privilegi di accesso.

Le password di accesso vengono automaticamente crittografate prima di essere salvate nel database.

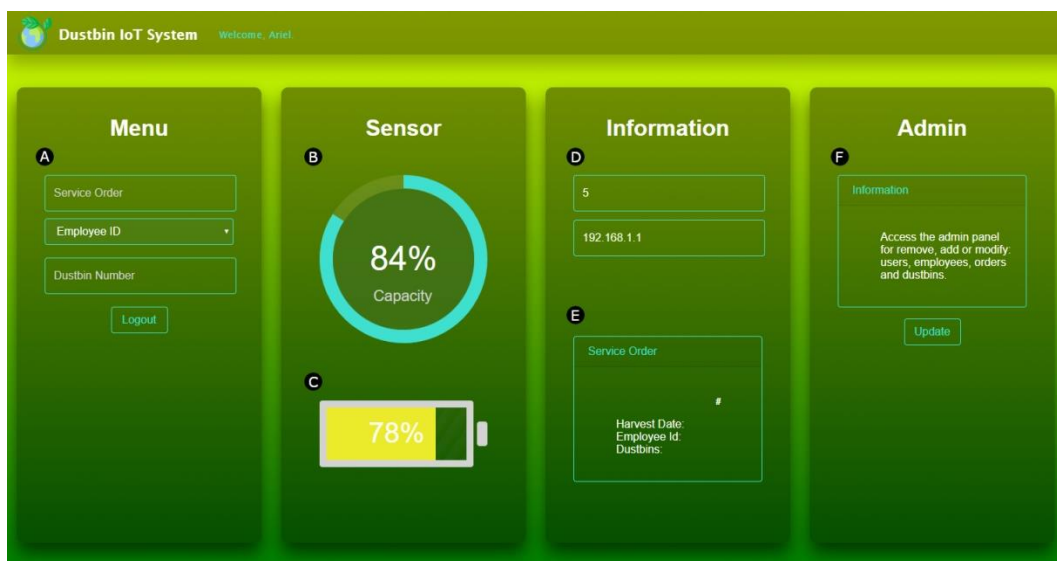


Figura 2 (Interfaccia di sistema dell'amministratore)

Una volta effettuato l'accesso come amministratore, verrà visualizzata l'interfaccia mostrata nella figura 2, contenente i punti elencati di seguito.

- Nel punto A, è possibile visualizzare un modulo per ottenere le informazioni relative ai sensori, filtrandole in base all'ID, accedere a tutti gli ordini di servizio in base al loro numero di identificazione, controllare gli ordini di servizio più vicini alla scadenza per ciascun operatore e uscire dalla sessione con il tasto di logout.
- Nel punto B e C, troviamo una rappresentazione grafica dei dati del sensore, ovvero la capacità del cestino e lo stato della batteria che alimenta il sensore.
- I campi nel punto D, servono a mostrare il numero d'identificazione (o ID) e l'indirizzo IP del sensore. Quest'ultimo può essere utile per interventi nella configurazione della rete.
- Il punto E, viene utilizzato per visualizzare le informazioni sull'ordine di servizio come il numero identificativo, la data di ritiro, il codice dell'operatore ad esso assegnato e la lista dei cestini da svuotare.
- Il punto F, consente l'accesso al pannello di controllo dove è possibile svolgere attività di impatto critico sul programma, come negare l'accesso a tutti gli altri utenti. Per questo motivo solo l'amministratore di sistema può accedervi.

Le funzioni del resto dei punti non cambiano.

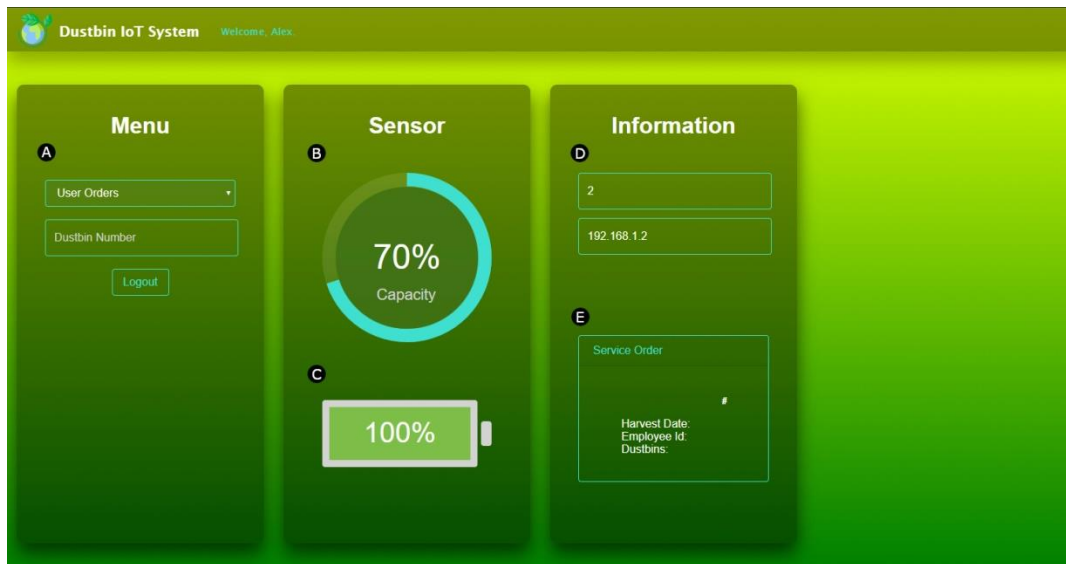


Figura 3 (Interfaccia di sistema dell'operatore)

Le differenze dell'interfaccia dell'amministratore (figura 2) rispetto a quelle dell'interfaccia operatore (figura 3) sono elencate di seguito.

- La prima cosa da notare nella figura 3 è l'assenza del menu amministratore (figura 2, punto F) per impedire l'accesso al pannello di controllo.
- Nel punto A dell'interfaccia operatore (figura 3) non è possibile accedere a tutti gli ordini di servizio, ma solo a quelli assegnati all'operatore che ha effettuato l'accesso, gli è anche consentito visualizzare le informazioni relative ai sensori.

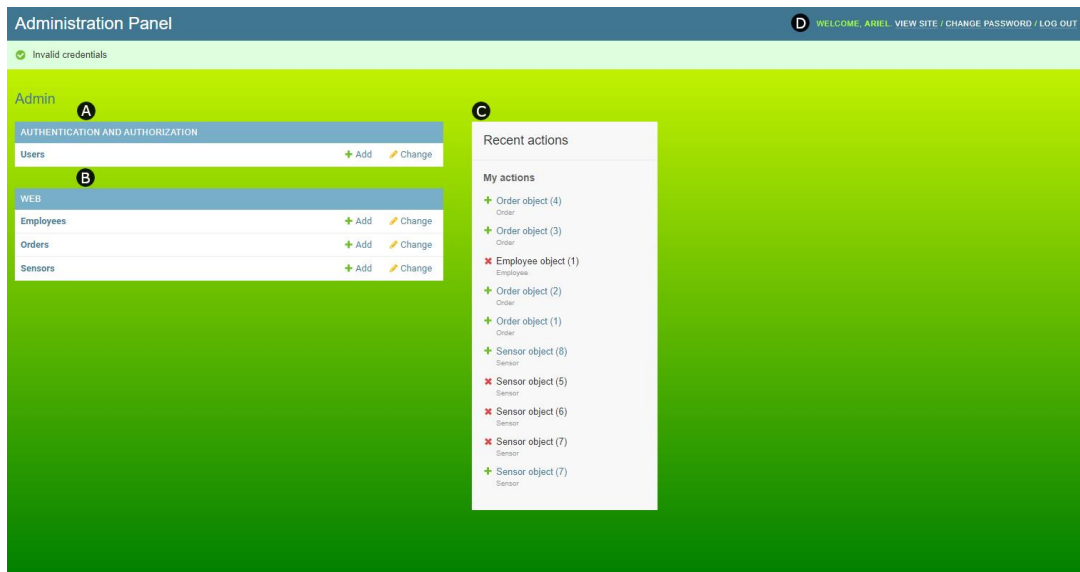


Figura 4 (Pannello di controllo dell'amministratore)

All'interno della figura 4:

- Il punto A, consente all'amministratore di modificare o creare nuovi accessi (o utenti) nell'applicazione.
- Nel punto B, è possibile creare o modificare nuovi ordini, sensori od operatori. Tuttavia, questi ultimi vengono generati automaticamente dopo aver creato un nuovo accesso all'applicazione.
- Nel punto C, vengono visualizzate semplicemente le ultime modifiche registrate all'interno del pannello di controllo.
- Il punto D, fornisce all'amministratore le opzioni di tornare alla home page, modificare la password di accesso e disconnettersi.

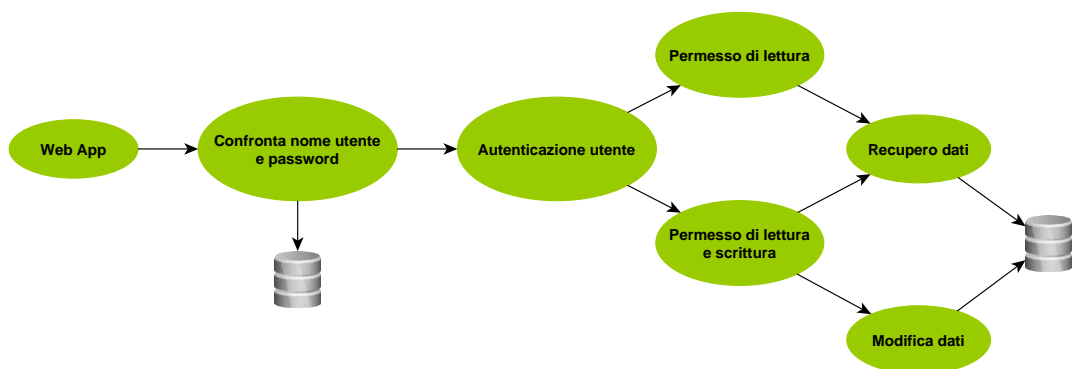


Figura 5 (Diagramma di flusso dell'applicazione web semplificato)

L'interoperabilità

I dati rilevati dai sensori dei cestini rifiuti devono essere consegnati al database dell'applicazione per consentire successivamente la loro visualizzazione.

L'invio dei suddetti dati avviene tramite un apposito calcolatore (pc, raspberry o arduino) con accesso a internet e dedicato alla ricompilazione, elaborazione e invio dell'informazione finale in formato JSON al database. Di tutto ciò si occupa un altro programma che ho sviluppato (di seguito denominato "db-filler") il quale consente di svolgere queste attività senza compromettere l'integrità dei dati o la sicurezza dell'database.

In ambito sicurezza, oltre alla gestione degli errori, consente l'accesso al database tramite una password che viene richiesta al primo accesso o in caso di riavvio del computer.

La seguente riga di codice all'interno del db-filler ottiene l'URL che consente l'accesso al database e che richiede la password nel caso in cui non si sia autenticati.

```
DATABASE_URL = popen('heroku config:get DATABASE_URL -a dustbin-  
iot').read()[:-1]
```

Il db-filler prima di scrivere i nuovi dati effettua dei controlli in database usando delle query, come si può apprezzare nel seguente codice.

```
# Create new sensors if id is None  
if i['id'] is None:  
    cur.execute("INSERT INTO web_sensor (capacity, battery,  
ip, location) VALUES(%(capacity)s, %(battery)s, %(ip)s,  
%(location)s)", i)  
  
else:  
    # Create new sensors with a specified id  
    if not exist(i):  
        cur.execute("INSERT INTO web_sensor (capacity, battery,  
ip, location, id) VALUES(%(capacity)s, %(battery)s, %(ip)s,  
%(location)s, %(id)s)", i)  
  
# Update sensors (find sensor by id)  
cur.executemany("UPDATE web_sensor SET capacity=%(capacity)s,  
battery=%(battery)s, ip=%(ip)s, location=%(location)s WHERE  
id=%(id)s", sensors)
```

Questi controlli consentono di identificare se un sensore è nuovo e va creata una nuova istanza oppure se esiste già e va aggiornato,

permettendo inoltre di sincronizzare l'esecuzione di questo programma con il momento in cui verranno ricevuti i dati provenienti dai sensori per una maggiore ottimizzazione delle prestazioni.

Altre informazioni

Prima dello sviluppo dell'applicazione web ho realizzato un prototipo concettuale utilizzando il framework Node-Red.

Ulteriori informazioni sull'applicazione e il suo prototipo, come il codice o il video informativo, sono disponibili online [2].

ANALISI DEL PROGETTO

Come spiegato in precedenza, il progetto Dustbin IoT è scalabile e consente l'integrazione di nuove funzionalità. Ciò consente a questa applicazione web di migliorare progressivamente. Di seguito è riportato un elenco di alcune delle funzionalità che potrebbero essere implementate per migliorare questo programma.

- Invio automatico di e-mail o notifiche agli operatori per comunicare l'assegnazione di una raccolta.
- Generazione automatica degli ordini di servizio in base ai seguenti parametri: operatori con meno carico di lavoro, stato di capienza dei cestini e giorni adatti alla raccolta (dal lunedì al venerdì, festivi esclusi).
- Calcolo del percorso più efficiente per eseguire la raccolta e possibilità di visualizzarlo su una mappa sfruttando l'accesso al GPS dall'applicazione.
- Supporto in più lingue per un maggiore comprensione da parte degli operatori.
- Tutor che spiega come utilizzare l'applicazione al primo accesso, ripetibile se richiesto.
- Possibilità di identificare chi è il responsabile del giorno corrente ed eventualmente contattarlo tramite l'app.

BIBLIOGRAFIA

- [1] A. Montes, «Dustbin IoT» [Online]. Disponibile in: dustbin-iot.herokuapp.com
- [2] A. Montes, «Project 1» [Online]. Disponibile in: montesariel.com/portfolio/project-1