

# LA *BUSINESS INTELLIGENCE* APPLICATA ALLA LOCALIZZAZIONE 4.0 E ALLA REALTA' AUMENTATA

*Commissione Industria 4.0*

*Ordine dei Dottori Commercialisti ed Esperti Contabili di Milano*

*e con il contributo di:*

*Riccardo Barbatti*

*Marco Sirini*

## Sommario

LA BUSINESS INTELLIGENCE APPLICATA ALLA LOCALIZZAZIONE 4.0 E ALLA REALTA' AUMENTATA.....	1
Prefazione.....	3
Il risultato della localizzazione 4.0 e della realtà aumentata .....	4
Paragrafo 1.1: la logica della rilevazione degli oggetti (a cura di Marco Sirini).....	5
Paragrafo 1.2: l'utilizzo del <i>raspberry</i> per la rilevazione dei dati degli oggetti (a cura di Marco Sirini)....	9
Paragrafo 1.3: l'utilizzo del <i>server SQL</i> per immagazzinare i dati del <i>raspberry</i> (a cura di Marco Sirini)	10
Paragrafo 1.4: il recupero dei dati dal <i>server SQL</i> .....	11
Paragrafo 1.5: formule e misure.....	13
Paragrafo 1.6: la visualizzazione dei dati.....	15
Paragrafo 1.7: la realtà aumentata (a cura di Riccardo Barbatti).....	16
Conclusioni .....	17

## Prefazione

Lo scopo del presente elaborato risulta essere quello di introdurre i Colleghi a come possano venire sfruttate le più avanzate tecnologie di rilevazione degli oggetti nello spazio, e conseguentemente, come queste possano creare un grande valore legato all'analisi dei dati che vengono in tal modo restituiti.

Inoltre, i dati che vengono ottenuti dalla rilevazione degli oggetti, risultano il punto di partenza che consente di abbinarvi altre tecnologie abilitanti dell'industria 4.0, ed in particolare, nel presente elaborato trattasi di *big data* e *analytics* e di realtà aumentata.

Il concetto di fondo che si vorrebbe mettere in risalto è che non risulta decisivo come il dato viene recuperato, bensì che risulta fondamentale ottenerlo e successivamente relazionarlo alle altre tecnologie, col fine ultimo di consentire il ritorno di dati strutturati agli imprenditori, i quali possono essere utili per il loro processo di definizione della strategia aziendale.

La rilevazione degli oggetti nello spazio, e dei dati che essi possono ritornare, si pone nella sostanza quale alternativa al *data entry*.

In sintesi, potremmo dire che l'utilizzo di queste nuove tecnologie consente di avere a disposizione degli operatori una quantità di dati che possono, a loro volta, restituire informazioni fondamentali per l'assunzione di decisioni strategiche in tempo reale rispetto agli accadimenti aziendali.

Quanto proposto nel presente documento vorrebbe passare il messaggio che la tecnologia può essere applicata alla realtà aziendale ai fini della raccolta, analisi, e trattamento del dato; e questo, per rispondere ad esigenze aziendali quali ad esempio conoscere il ciclo produttivo al fine di aumentarne la redditività, garantendo così in ultima analisi la continuità aziendale.

Oltre all'*analytical* ed al *big data*, un'altra potente e nuovissima tecnologia che viene presa in considerazione nel presente elaborato risulta essere la realtà aumentata, e ciò dato che il suo utilizzo ci consente di avere disponibili dati e risultati della loro analisi tramite immagini virtuali collegate agli oggetti reali, e questo, senza dover sfiorare nemmeno un tasto della nostra tastiera.

## Il risultato della localizzazione 4.0 e della realtà aumentata

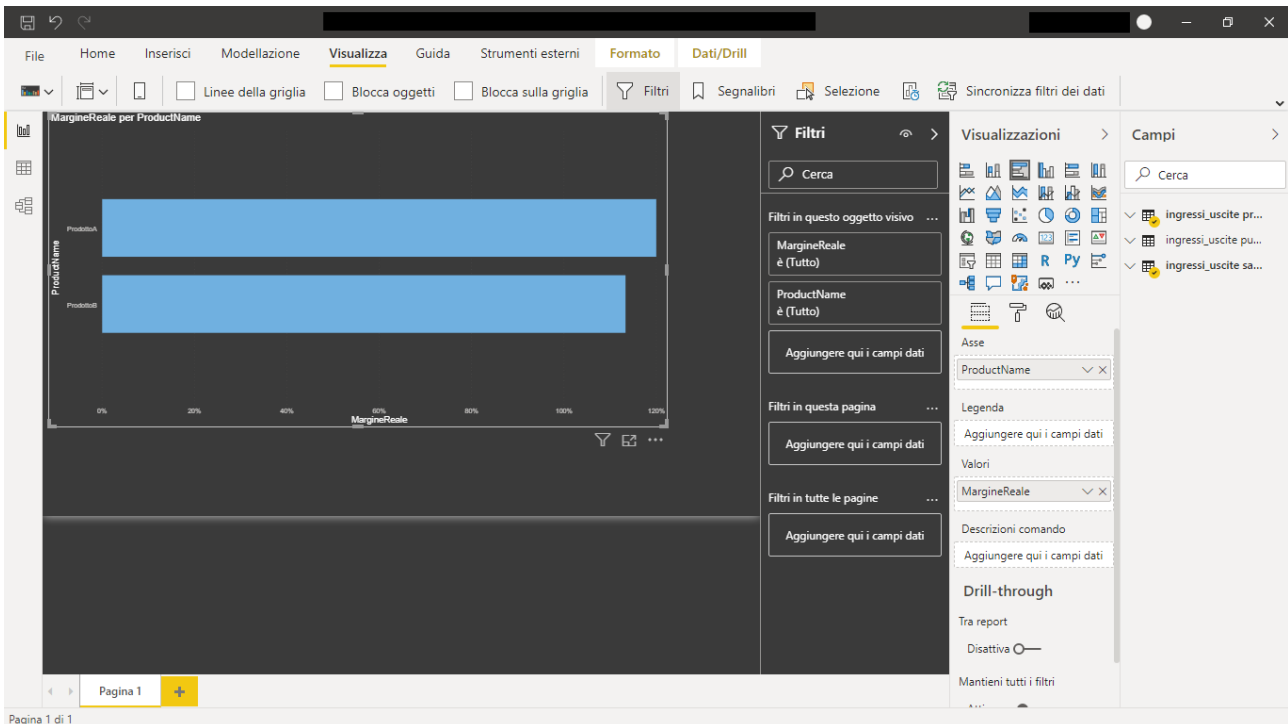


Fig. 1

I grafici consentono di visualizzare e di analizzare i dati recuperati dagli oggetti nello spazio.



Fig. 2

La realtà aumentata consente di “vedere” i dati degli oggetti reali e se desiderato collegarvi immagini e video a questi relativi.

Paragrafo 1.1: la logica della rilevazione degli oggetti (a cura di Marco Sirini)

Come già indicato poc' anzi, la linea di partenza del presente lavoro è incentrata sulla rilevazione degli oggetti nello spazio.

In particolare, tale rilevazione viene effettuata attraverso dei sensori di forma tondeggiante denominati *beacon ble* incollati al di sotto degli oggetti fisici utilizzato allo scopo.

Questi *beacon ble* sono in estrema sintesi una sorta di piccoli fari i quali, anziché emettere fotoni e quindi luce, emettono onde radio di tipo *bluetooth*.

A questo punto, l'altra metà dell'architettura elettro-informatica necessaria per la nostra rilevazione degli oggetti risulta essere il *computer* "tascabile" di tipo *raspberry*, il quale per il tramite delle antenne di cui risulta dotato ha la possibilità di captare gli oggetti ai quali viene applicato il *beacon ble*.

In sostanza, i *raspberry* grazie al fatto che sono a loro volta dotati di un modulo *bluetooth*, possono captare il segnale emanato dai *beacon ble* e conseguentemente elaborare i dati che questi ultimi trasmettono (*mac address* e *RSSI*).

In particolare, il *mac address* è l'identificativo del *beacon ble*, mentre il *RSSI* risulta essere la potenza del segnale.

Il procedimento di rilevazione consiste semplicemente nel fatto che se la potenza del segnale risulta essere al di sopra di una certa soglia, ecco che l'oggetto al quale è stato applicato il *beacon ble* viene scritto nel *database*, diversamente ciò non avviene.

Il *raspberry* (se la potenza del segnale è sufficiente) se rileva per la prima volta l'oggetto lo va ad iscrivere vuoi come *purchase*, vuoi come *sale* a seconda sia del tipo di oggetto rilevato (difatti può essere programmato come acquisto o vendita) sia di quale *raspberry* si tratta (ne viene programmato uno per la rilevazione degli acquisti ed uno per la rilevazione delle vendite).

I dati tecnici di quanto sopra sono i seguenti:

Impostata scheda *WiFi* del *raspberry PI* per connettersi alla rete tramite *access point*: il primo *raspberry* con IP 192.168.1.45, il secondo con IP 192.168.1.46.

Sul *server* abbiamo avviato *xampp* (*apache* e *mysql*) dove girano le *API* scritte in *php*.

Sui *raspberry* abbiamo scritto degli script in *js* che vengono mandati *run* con l'utilizzo del *framework* lato server chiamato *node.js* (sviluppato da *Google*).

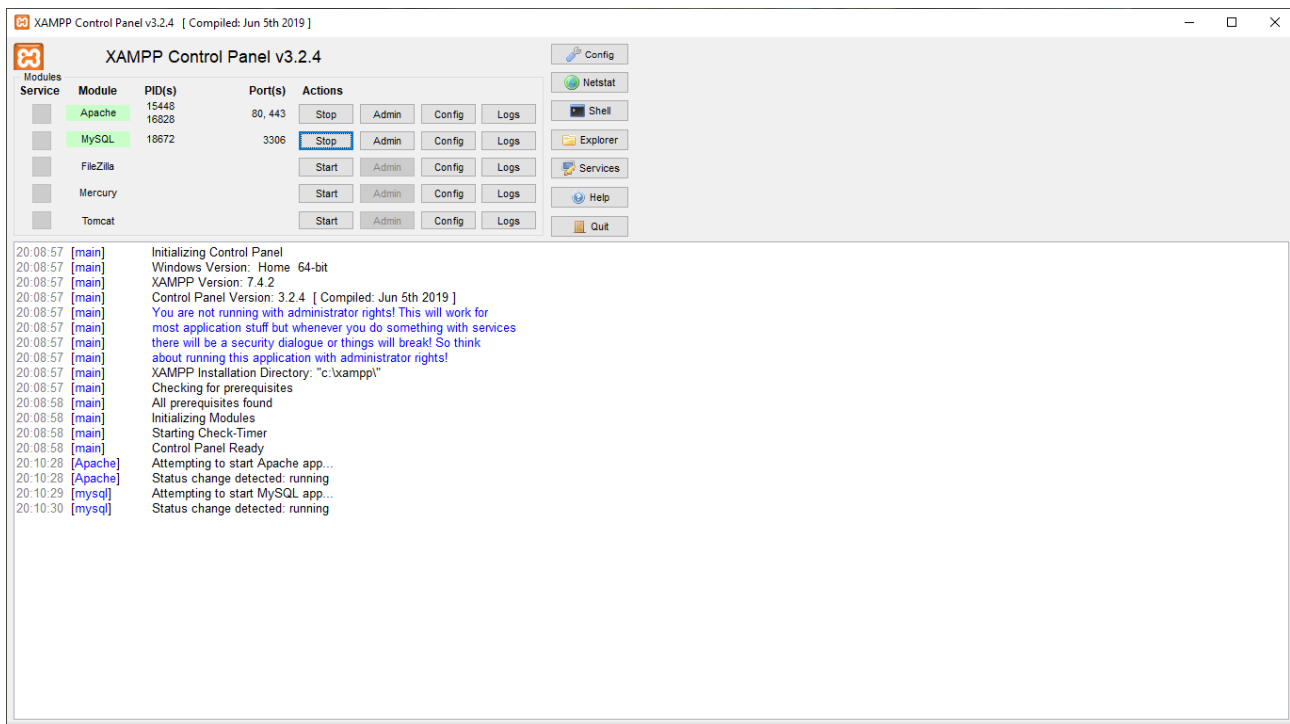


Fig. 3

Pannello di controllo di *xampp*.

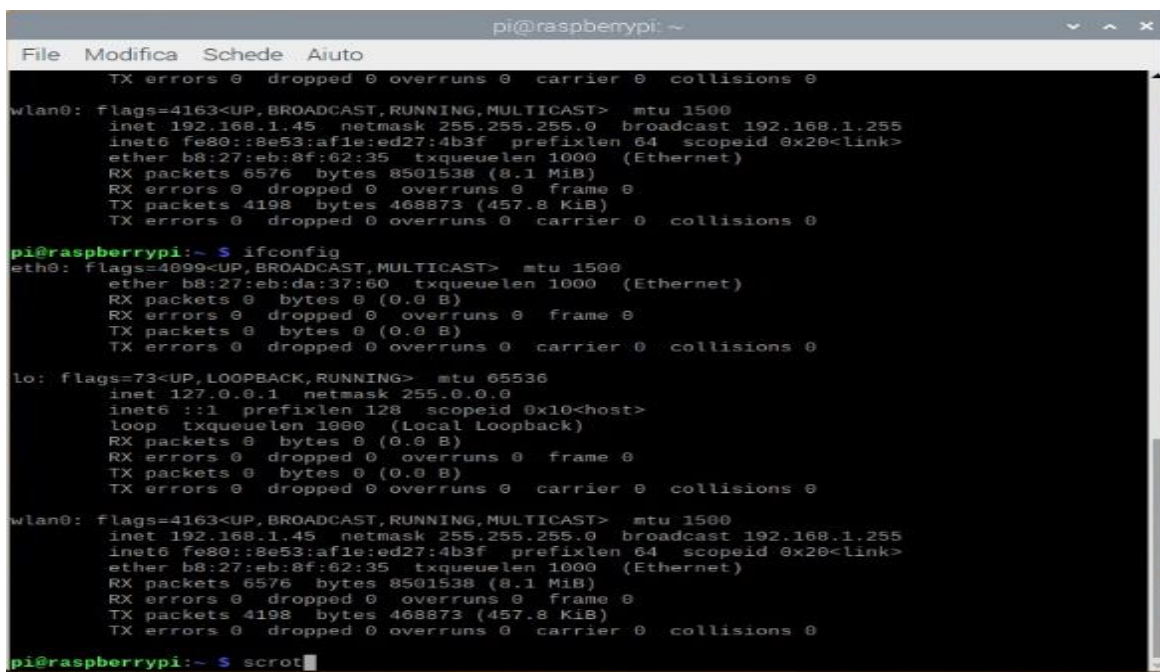


Fig. 4

Questo è uno *screenshot* dell'interfaccia grafica del *raspberrypi*, con aperta la *shell* in cui è possibile la configurazione di rete.

Per ottenere questo *screenshot* sul *raspberrypi* abbiamo usato il comando *scrot* e recuperato l'immagine nella cartella `/home/pi`.

Per collegarci al *raspberry* da *Windows* abbiamo utilizzato *Putty*.

Successivamente abbiamo verificato la connessione dei sistemi.

```
pi@raspberrypi ~$ ping 192.168.1.71
64 bytes from 192.168.1.71: icmp_seq=82 ttl=128 time=96.7 ms
64 bytes from 192.168.1.71: icmp_seq=83 ttl=128 time=4.94 ms
64 bytes from 192.168.1.71: icmp_seq=84 ttl=128 time=8.33 ms
64 bytes from 192.168.1.71: icmp_seq=85 ttl=128 time=4.85 ms
64 bytes from 192.168.1.71: icmp_seq=86 ttl=128 time=9.53 ms
64 bytes from 192.168.1.71: icmp_seq=87 ttl=128 time=56.2 ms
64 bytes from 192.168.1.71: icmp_seq=88 ttl=128 time=20.3 ms
64 bytes from 192.168.1.71: icmp_seq=89 ttl=128 time=15.8 ms
64 bytes from 192.168.1.71: icmp_seq=90 ttl=128 time=4.59 ms
64 bytes from 192.168.1.71: icmp_seq=91 ttl=128 time=102 ms
64 bytes from 192.168.1.71: icmp_seq=92 ttl=128 time=15.6 ms
64 bytes from 192.168.1.71: icmp_seq=93 ttl=128 time=12.8 ms
64 bytes from 192.168.1.71: icmp_seq=94 ttl=128 time=4.78 ms
64 bytes from 192.168.1.71: icmp_seq=95 ttl=128 time=4.90 ms
64 bytes from 192.168.1.71: icmp_seq=96 ttl=128 time=2492 ms
64 bytes from 192.168.1.71: icmp_seq=97 ttl=128 time=419 ms
64 bytes from 192.168.1.71: icmp_seq=100 ttl=128 time=5.03 ms
64 bytes from 192.168.1.71: icmp_seq=101 ttl=128 time=62.7 ms
64 bytes from 192.168.1.71: icmp_seq=102 ttl=128 time=10.4 ms
64 bytes from 192.168.1.71: icmp_seq=103 ttl=128 time=4.44 ms
64 bytes from 192.168.1.71: icmp_seq=104 ttl=128 time=10.9 ms
64 bytes from 192.168.1.71: icmp_seq=105 ttl=128 time=11.4 ms
64 bytes from 192.168.1.71: icmp_seq=106 ttl=128 time=14.8 ms
64 bytes from 192.168.1.71: icmp_seq=107 ttl=128 time=4.89 ms
64 bytes from 192.168.1.71: icmp_seq=108 ttl=128 time=5.26 ms
64 bytes from 192.168.1.71: icmp_seq=109 ttl=128 time=4.77 ms
64 bytes from 192.168.1.71: icmp_seq=110 ttl=128 time=5.14 ms
64 bytes from 192.168.1.71: icmp_seq=111 ttl=128 time=5.11 ms
64 bytes from 192.168.1.71: icmp_seq=112 ttl=128 time=5.07 ms
64 bytes from 192.168.1.71: icmp_seq=114 ttl=128 time=19.4 ms
64 bytes from 192.168.1.71: icmp_seq=115 ttl=128 time=9.35 ms
64 bytes from 192.168.1.71: icmp_seq=116 ttl=128 time=6.98 ms
64 bytes from 192.168.1.71: icmp_seq=117 ttl=128 time=8.02 ms
64 bytes from 192.168.1.71: icmp_seq=118 ttl=128 time=13.9 ms
64 bytes from 192.168.1.71: icmp_seq=119 ttl=128 time=7.75 ms
64 bytes from 192.168.1.71: icmp_seq=120 ttl=128 time=57.8 ms
64 bytes from 192.168.1.71: icmp_seq=121 ttl=128 time=11.1 ms
64 bytes from 192.168.1.71: icmp_seq=122 ttl=128 time=5.55 ms
64 bytes from 192.168.1.71: icmp_seq=123 ttl=128 time=10.6 ms
64 bytes from 192.168.1.71: icmp_seq=124 ttl=128 time=8.13 ms
64 bytes from 192.168.1.71: icmp_seq=125 ttl=128 time=7.75 ms
64 bytes from 192.168.1.71: icmp_seq=126 ttl=128 time=53.5 ms
64 bytes from 192.168.1.71: icmp_seq=127 ttl=128 time=68.10 ms
64 bytes from 192.168.1.71: icmp_seq=127 ttl=128 time=72.5 ms (DUP!)
64 bytes from 192.168.1.71: icmp_seq=128 ttl=128 time=7.97 ms
```

Fig. 5

Abbiamo verificato la comunicazione tra *server* e *raspberry* con comando *ping*.

Alla fine di queste operazioni, abbiamo potuto alimentare il nostro *server* SQL con le movimentazioni degli oggetti, rappresentanti idealmente due prodotti. A loro volta, i prodotti hanno un esemplare dedicato alla rilevazione dell'atto di acquisto, ed uno (ovviamente identico) dedicato all'atto della vendita.



Fig. 6

Il raspberry e i due oggetti atti ad essere rilevati nello spazio.

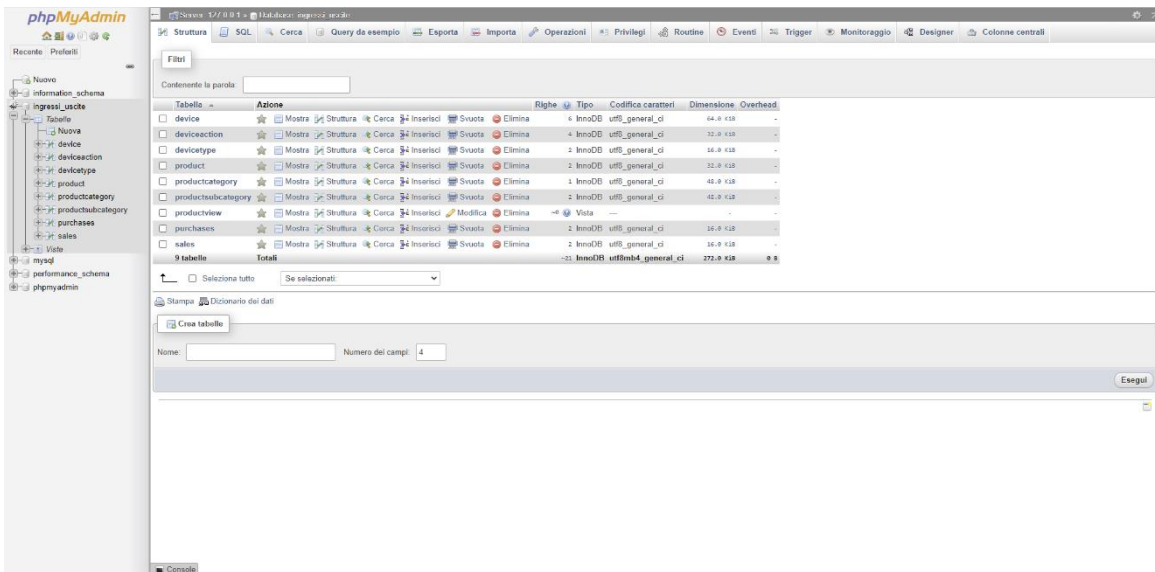


Fig. 7

Fig. 7

Database MySQL visualizzato con PHPMyAdmin.



Paragrafo 1.2: l'utilizzo del *raspberry* per la rilevazione dei dati degli oggetti (a cura di Marco Sirini)  
I *raspberry* sono dei *minicomputer* che hanno come sistema operativo *linux* e che grazie al modulo *bluetooth* presente al loro interno, possono rilevare i *beacon ble*.

Quando ciò avviene, inviano i dati al *server* attraverso una chiamata *http* a una *API*.

Come anticipato in precedenza, i *raspberry* utilizzano il sistema operativo *linux*, il quale prende il nome di *raspbian*.

Su di questi viene installato un *framework* che si chiama *node.js*, il quale altro non è che un *framework* sviluppato da *Google* che permette di eseguire codice *javascript* lato *server*.

Senza questo *framework javascript*, che come noto risulta essere un linguaggio pensato solo per lo sviluppo lato *client*, non potrebbe eseguire operazioni lato *server*.

Risulta esserci uno *script* relativo a ciascun *raspberry* nella cartella */home/pi* e che prende il nome di *accessControl.js*.

Ognuno di questi *script* monitora se sono presenti dei *device ble (bluetooth low energy)* nelle vicinanze utilizzando la scheda *bluetooth* del *raspberry*.

In caso di rilevazione, invia i dati propri di ogni oggetto rilevato al *server*.

In questo caso il *server* risulta essere un *computer Windows* sul quale è installato un *web server*, un *php server* e un *mysql server* (tutti all'interno di un pacchetto che si chiama *xampp*), e che per funzionare correttamente debbono essere tutti preliminarmente avviati.

Il *software* scritto lato *server* è in linguaggio *php*, e ha questi obiettivi:

- alcune parti sono funzionali a iscrivere nel *database* i dati;
- altre servono a ricevere i dati dai *raspberry* (tramite *web API*); in particolare le *API* discriminano i dati *in primis* verificando se il *mac address* è riconosciuto (cioè appartiene ad un oggetto tracciabile o meno) e secondariamente, se questo è il caso, verifica se l'oggetto tracciato rappresenta un atto di acquisto proveniente da un *raspberry* atto a rilevare gli acquisti, o invece se si tratta di un oggetto tracciabile che rappresenta un atto di vendita proveniente da un *raspberry* atto a rilevare le vendite.

Se i dati rilevati dai *raspberry* non sono già presenti nel *database* vengono inseriti nelle opportune tabelle (*sales* e *purchases*).

Infine, appare opportuno evidenziare che i *raspberry* filtrano attraverso lo *script* in *js* i *mac address* a seconda della potenza di segnale rilevata (denominata *RSSI*); si tratta di un segnale che normalmente risulta essere tra i -99 e gli zero *decibel*.

Più la potenza del segnale è vicina a zero, più il *ble* è vicino al *raspberry*: nel nostro lavoro è stato inserito un filtro in cui tutti i *ble* che vengono rilevati sotto i -75 *decibel*, quindi da -75 a -99 vengono scartati, in quanto sono troppo lontani per essere considerati passati sotto le antenne.

Paragrafo 1.3: l'utilizzo del *server SQL* per immagazzinare i dati del *raspberry* (a cura di Marco Sirini)  
Il *server mysql* serve per immagazzinare i dati, passando attraverso le *API* scritte in *php*.

Il *software* è in grado di discriminare se è un *beacon ble* associato agli oggetti tracciabili o meno; questo in quanto l'ambiente moderno risulta essere saturo di dispositivi *bluetooth* che inviano segnali. Conseguentemente, risulta necessario un filtro sul *mac address* che possa consentire al *software* di discriminare se si tratta di oggetti tracciabili relativi all'atto di acquisto (o vendita) rilevabili dal *raspberry* dedicato.

Il *database mysql* è accessibile attraverso il linguaggio *php* che ha al suo interno delle librerie già pronte per dialogare con il *dbms*: vi sono delle classi *php ad hoc* per il dialogo.

All'interno del *database* vi sono le seguenti tabelle:

- Device;
- deviceAction;
- deviceType;
- product;
- productcategory;
- productsubcategory;
- purchases;
- sales;

Il *database* è visualizzabile attraverso l'interfaccia grafica di *phpmyadmin*, presente in *xampp*.

Tutti i dati utili sono qui memorizzati, ossia risultano visibili gli acquisti e le vendite ricavati dal passaggio degli oggetti tracciati nelle vicinanze dei *raspberry*.

## Paragrafo 1.4: il recupero dei dati dal server SQL

A questo punto, per importare i dati in *powerBI*, occorre connettersi al *server SQL* con un connettore già pronto e predisposto all'interno del programma medesimo.

Si accede da Home / Recupera dati / Altro / Database MySQL:

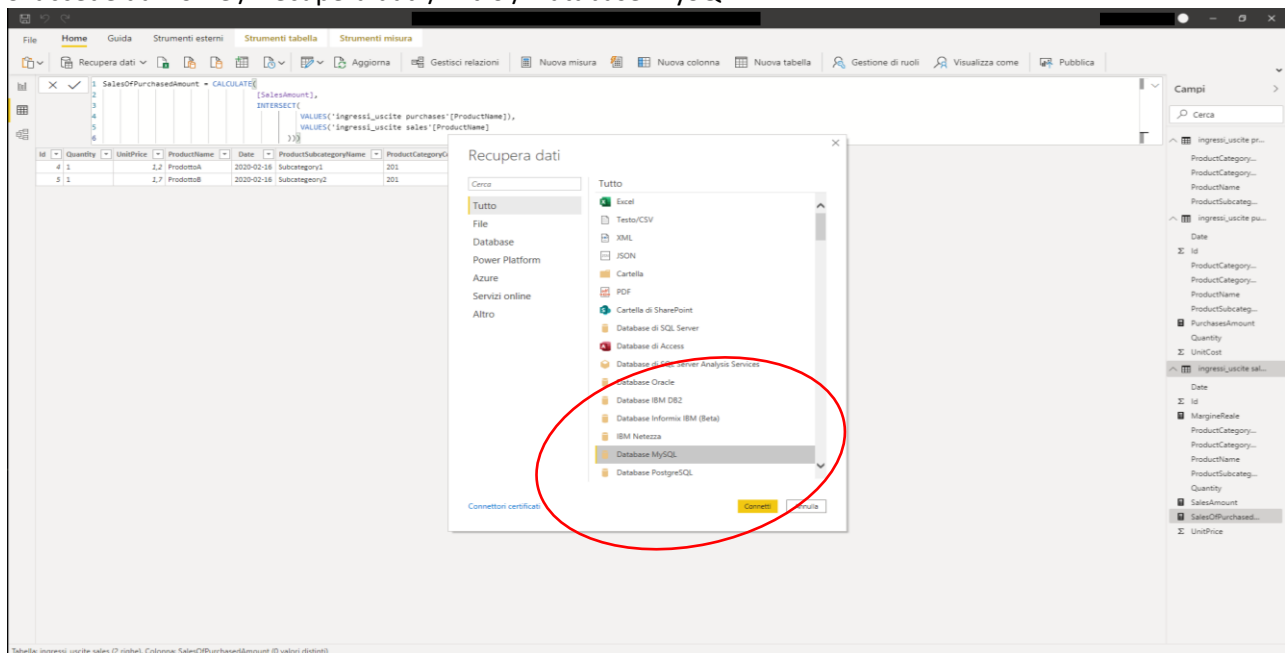


Fig. 8

Come si nota, il *software* ha una grandissima capacità di importazione di dati, praticamente da qualsiasi *database*, ivi compresi naturalmente quelli gestiti con il linguaggio SQL.

Ciò che recuperiamo dal *server SQL* sono le tre tabelle: *ingressi\_uscite purchases*, *ingressi\_uscite sales* e *ingressi\_uscite productview*.

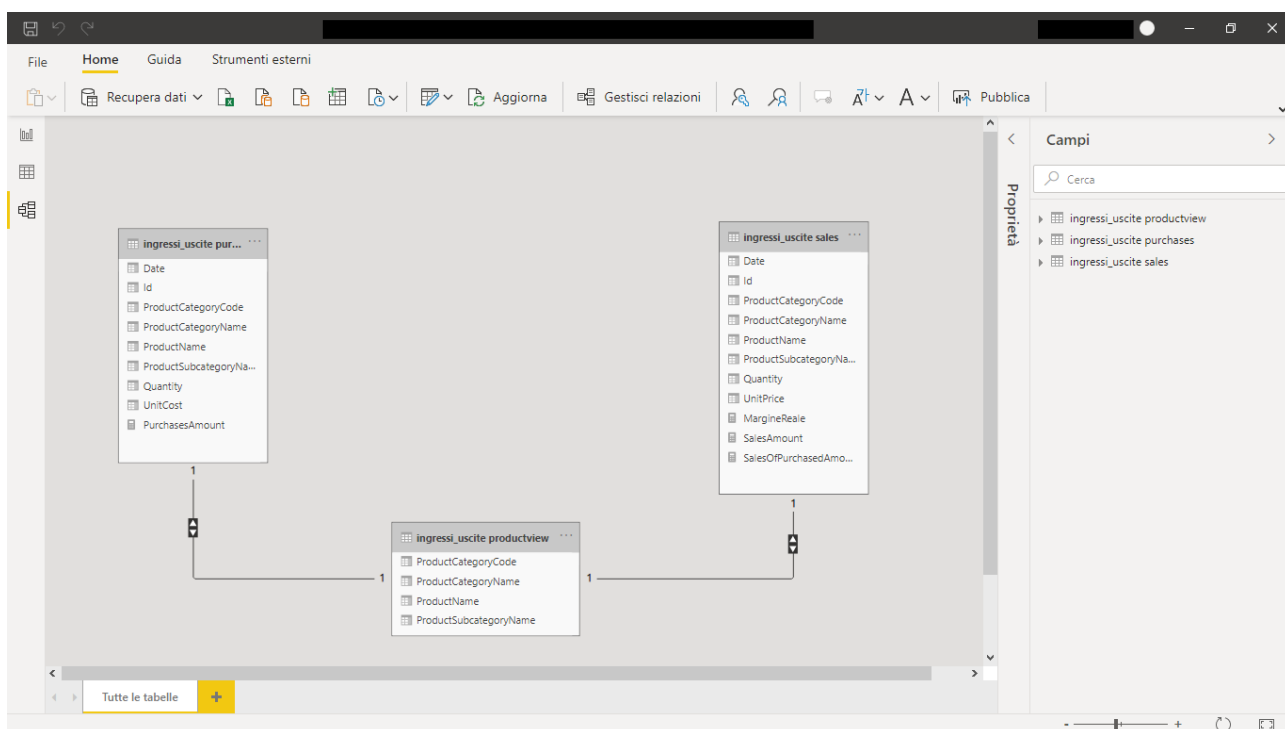


Fig. 9

In sostanza, abbiamo importato le due tabelle fatto, che sono la “ingressi\_uscite purchases” e “ingressi\_uscite sales”, oltre alla tabella dimensione “ingressi\_uscite productview”.

Tali tabelle le abbiamo ottenute con SQL, ma la “ingressi\_uscite productview”, la quale ha il compito di connettere le due tabelle fatto, l'avremmo potuta creare (in mancanza della sua creazione direttamente con SQL), con una formula in DAX levereggiante le funzioni DAX: DISTINCT, UNION e ALL.

Una modalità alternativa di recupero dei dati degli oggetti nello spazio potrebbe essere quella dei *barcode*, tecnologia oramai “tradizionale” e di uso comune ad esempio nella GDO.

Il funzionamento logico rimane identico, in quanto il punto di caduta dell'utilizzo dei *barcode* risulta essere la creazione di un *database* da cui estrarre a sua volta i dati per le analisi dei margini.



Fig. 10

Un esempio di codice a barre.

## Paragrafo 1.5: formule e misure

Come indicato in precedenza, il presente elaborato mira a “fondere” diverse tecnologie industria 4.0, tra le quali vi sono la rilevazione degli oggetti nello spazio ed i *big data* e *analytics*.

L’obiettivo più “operativo” risulta invece essere quello del calcolo del margine reale dei prodotti, i quali sono stati rappresentati da due coppie di oggetti tracciabili, le quali a loro volta vedono un esemplare rappresentante l’atto di acquisto e il suo gemello rappresentare l’atto di vendita.

Il rilevamento di questi oggetti tracciabili tramite *raspberry* ha simulato l’atto di acquisto (entrata nel punto vendita) e l’atto di vendita (uscita dal punto vendita).

La prima formula di cui abbiamo bisogno è la misura che ci consente di calcolare la somma degli acquisti:

```
PurchasesAmount =  
    SUMX('ingressi_uscite purchases',  
        'ingressi_uscite purchases'[Quantity] * 'ingressi_uscite purchases'[UnitCost]  
    )
```

Identica nella sostanza è la formula della misura che ci consente di calcolare la somma delle vendite:

```
SalesAmount =  
    SUMX('ingressi_uscite sales',  
        'ingressi_uscite sales'[Quantity]*'ingressi_uscite sales'[UnitPrice]  
    )
```

Successivamente, andiamo a calcolare (sempre con una misura) il totale delle vendite dei prodotti acquistati.

Per farlo utilizziamo la funzione INTERSECT che ci consente di tenere in considerazione solo i prodotti che esistono sia tra gli acquisti sia tra le vendite:

```
SalesOfPurchasedAmount = CALCULATE(  
    [SalesAmount],  
    INTERSECT(  
        VALUES('ingressi_uscite purchases'[ProductName]),  
        VALUES('ingressi_uscite sales'[ProductName])  
    )  
)
```

Infine, ecco pronta la misura per ottenere il margine reale:

```
MargineReale = CALCULATE(  
    DIVIDE(  
        [SalesOfPurchasedAmount],  
        [PurchasesAmount]  
    )  
)
```

Esperimento DB relazionali con localizzazione raspberry eseguito il 202-2-16 - Power BI Desktop

File Home Guida Strumenti esterni Strumenti tabella Strumenti misura

Recupera dati Aggiorna Gestisci relazioni Pubblica

```

1 MargineReale = CALCULATE(
2     DIVIDE(
3         [SalesOfPurchasedAmount],
4         [PurchasesAmount]
5     )
6 )

```

Id	Quantity	UnitPrice	ProductName	Date	ProductSubcategoryName	ProductCategoryCode	ProductCategoryName
4	1	1,2	ProdottoA	2020-02-16	Subcategory1	201	CategoryZ
5	1	1,7	ProdottoB	2020-02-16	Subcategory2	201	CategoryZ

Tabella: ingressi\_uscite sales (2 righe). Colonna: MargineReale (0 valori distinti)

Campi

- ingressi\_uscite purchases
  - Date
  - Σ Id
  - ProductCategoryCode
  - ProductCategoryName
  - ProductName
  - ProductSubcategoryName
  - PurchasesAmount
  - Quantity
  - Σ UnitCost
- ingressi\_uscite sales
  - Date
  - Σ Id
  - MargineReale
  - ProductCategoryCode
  - ProductCategoryName
  - ProductName
  - ProductSubcategoryName
  - Quantity
  - SalesAmount
  - SalesOfPurchasedAmount
  - Σ UnitPrice

Fig. 11

In evidenza la formula della misura del margine reale.

## Paragrafo 1.6: la visualizzazione dei dati

A questo punto, occorre consentire una corretta visualizzazione dei dati.

Per farlo andiamo ad utilizzare un grafico ad istogramma con confrontati i due prodotti oggetto del presente elaborato, ovvero il prodotto A ed il prodotto B.

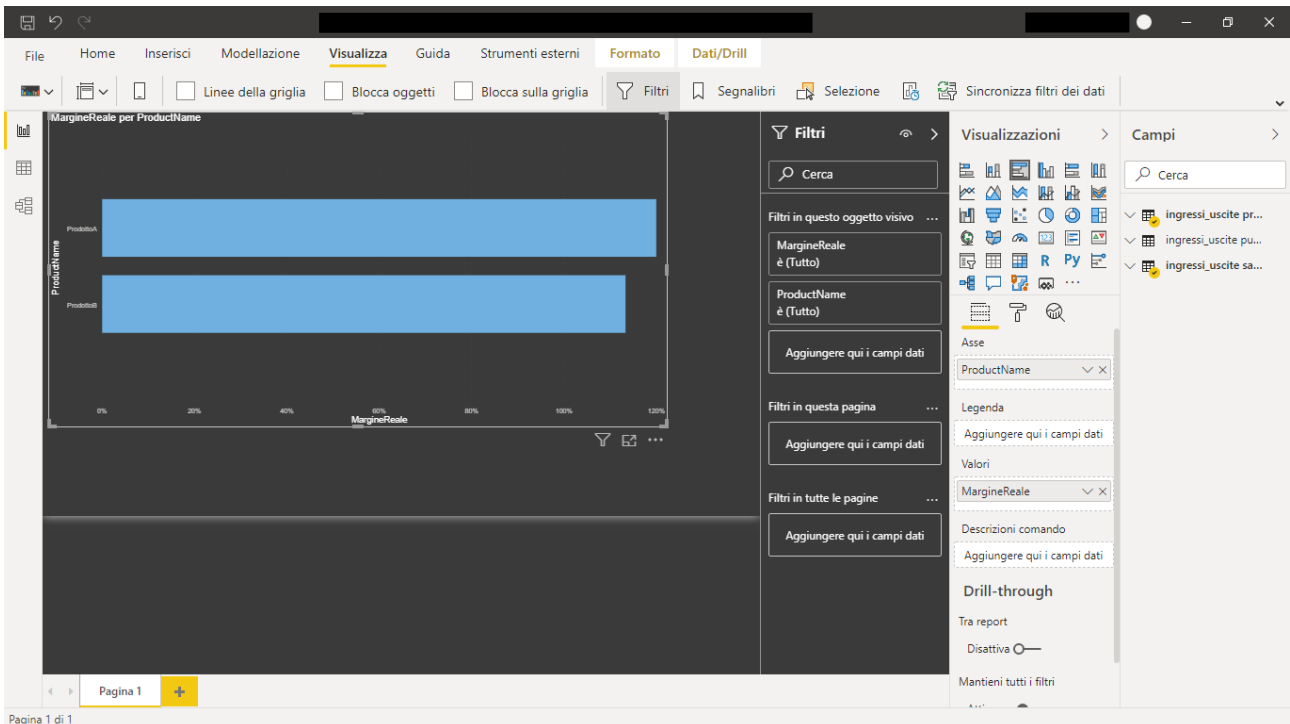


Fig. 12

Nei "valori" del grafico si è utilizzato la misura "MargineReale".

### Paragrafo 1.7: la realtà aumentata (a cura di Riccardo Barbatti)

Dopo aver rilevato le entrate e le uscite dei prodotti dal punto vendita tramite sistemi di localizzazione degli oggetti, abbiamo potuto immagazzinare i dati a queste correlate, ed infine calcolare il margine di vendita.

A questo punto, al lavoro sin qui fatto, possiamo “agganciare” un’ultima potente ed innovativa tecnologia abilitante dell’industria 4.0, ovvero la “realtà aumentata”.

Ma di cosa si tratta?

In estrema sintesi, puntando con un *device* opportunamente programmato un oggetto risulta possibile (tra le altre possibilità) riuscire a rendere leggibili i dati al suo interno contenuti senza dover utilizzare ulteriori interfacce informatiche.



Fig. 13

Nella figura 13 abbiamo come esempio illustrativo quello di una bottiglia di acqua minerale.

Quanto sopra si adatta perfettamente al nostro caso, in quanto l’oggetto di rilevazione nello spazio e di successiva analisi dei dati è il panierino dei prodotti venduti di un’impresa, tra i quali ben può esservi (soprattutto nel caso della GDO) una bottiglia d’acqua minerale.

L’utilità di quest’ultima tecnologia risulta evidente se immaginiamo di poter vedere immediatamente i margini dei prodotti che abbiamo di fronte a noi, potendo quindi, per esempio, elaborare delle politiche di posizionamento dei prodotti sugli scaffali.



## Conclusioni

Le tecnologie industria 4.0 che sono state utilizzate nel presente elaborato sono prese singolarmente molto utili; tuttavia, è dalla loro combinazione che ne deriva il modo migliore per utilizzarle.

L'esemplificazione di quanto poc'anzi indicato ne è stata data nelle pagine precedenti.

Dunque, non solo occorre possedere la conoscenza di queste nuove tecnologie per affrontare le sfide epocali che i nostri tempi ci pongono sia come Professionisti sia come semplici cittadini, ma anche la capacità di riuscire a progettare (almeno a livello concettuale) la loro combinazione per poter dare un contributo sia a livello consulenziale, sia a livello strategico ai nostri Clienti.

Inoltre, tutte le tecnologie industria 4.0 consentono di ottenere dati da attrezzature, impianti e più in generale sistemi complessi; ciò significa che *big data* e *analytics* sono assolutamente centrali, in quanto consentono di analizzare tutti questi dati.

Data la centralità dell'analisi dei dati, il ruolo del Commercialista sarà dunque fondamentale per gli operatori, se saprà fare propria la consapevolezza di dove le nuove tecnologie ci stanno velocemente portando.