



# Progetto

# OldMusa



# Introduzione

- Obiettivo
- Team

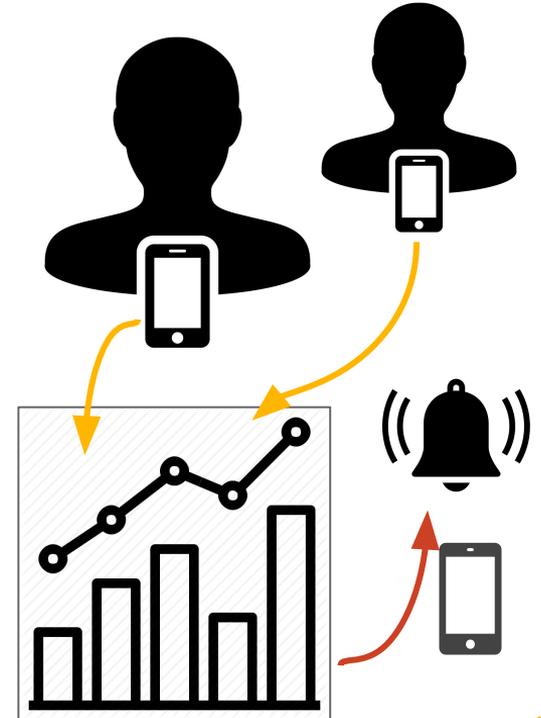
# Quale era la commessa



La commessa che ci ha affidato il **CNR**, con sede a Bologna, consiste in un'**applicazione** destinata all'uso in alcuni loro musei e siti.

Doveva permettere ai dipendenti di controllare, in maniera veloce, semplice e attraverso grafici, i valori rilevati dai sensori nelle varie sale.

Importante è l'avvertimento tramite notifica di eventuali errori nei valori ricevuti, perché i dipendenti possano accorgersene immediatamente e risolvere il problema il prima possibile.



# Il team



Lorenzo Rossi



Michele Silvestri

**Sviluppo Server**



Alessandro Pilleri



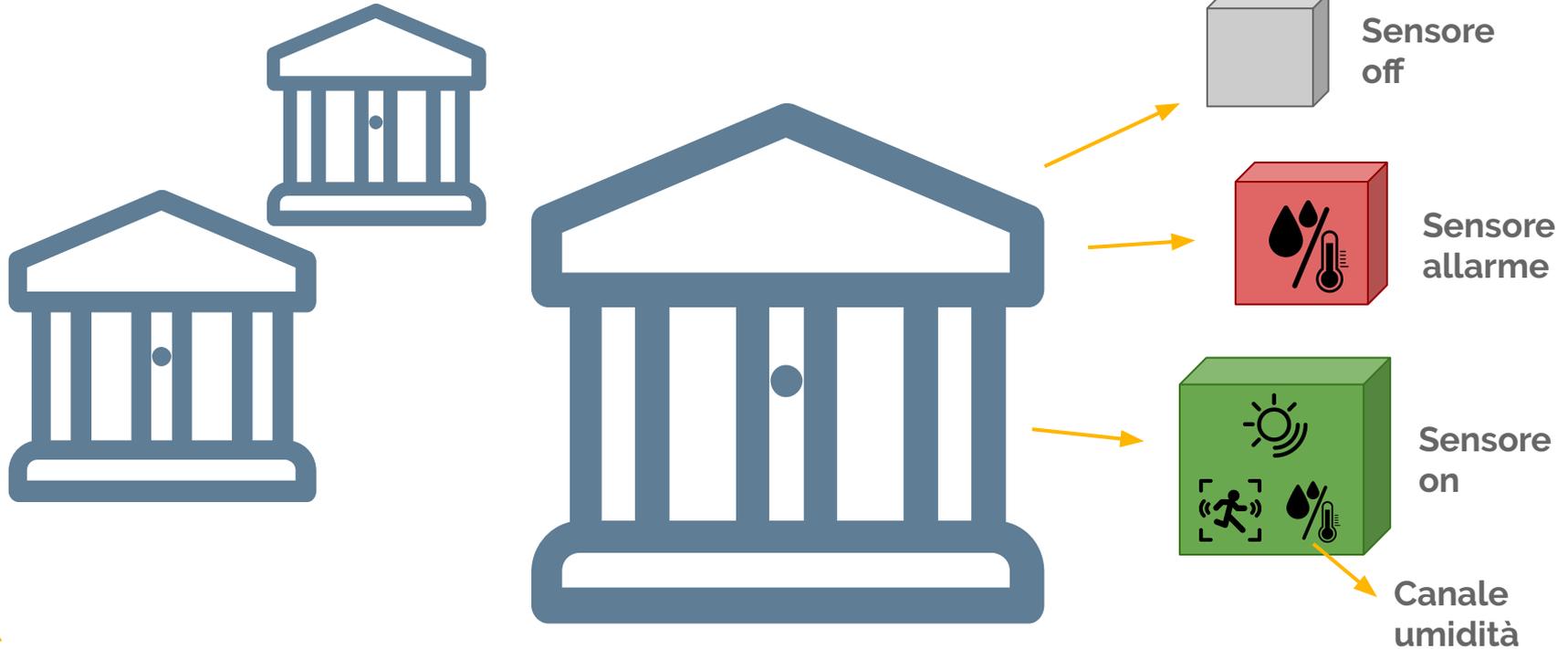
Giorgia Bertacchini

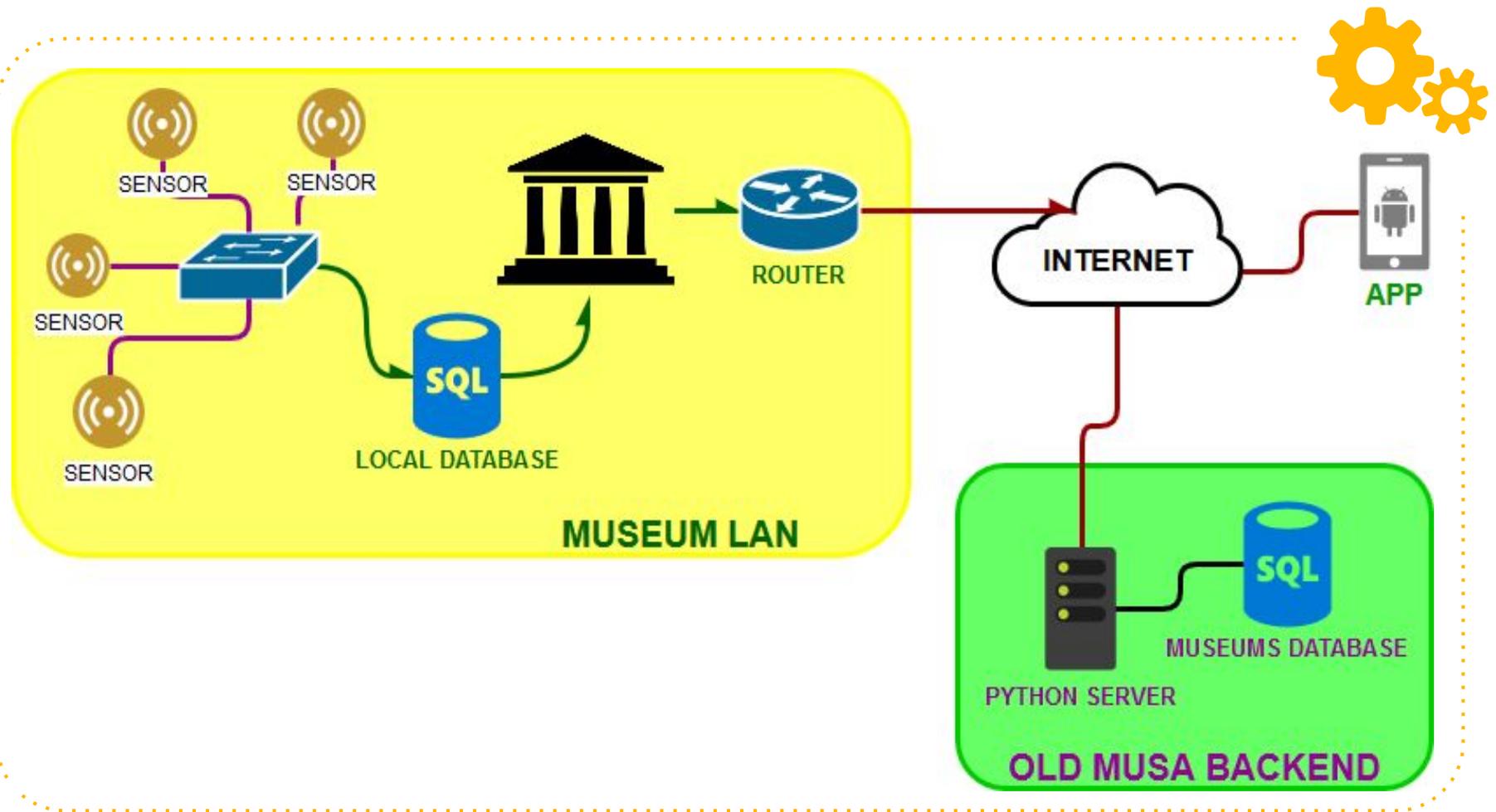
**Sviluppo Applicazione**

# Schema di progetto

- Funzionamento
- Struttura
- Sequenza schermate App
- Relazione tra database CNR e database OldMusa

# La struttura







# Il suo funzionamento

L'**utente** che sfrutta la nostra applicazione Android vede i valori rilevati dai sensori. Tali valori sono salvati nel nostro **Database**, che il **Server** carica facendo richieste al Database del CNR.

Dall'applicazione, oltre alla visualizzazione, è possibile anche la **modifica** e l'**aggiunta** dei dati nel database OldMusa, facilitando la modifica delle nostre tabelle. Ciò è possibile solo per gli utenti **amministratori**.





# La sequenza di layout



# Applicazione



**1**

**Login**

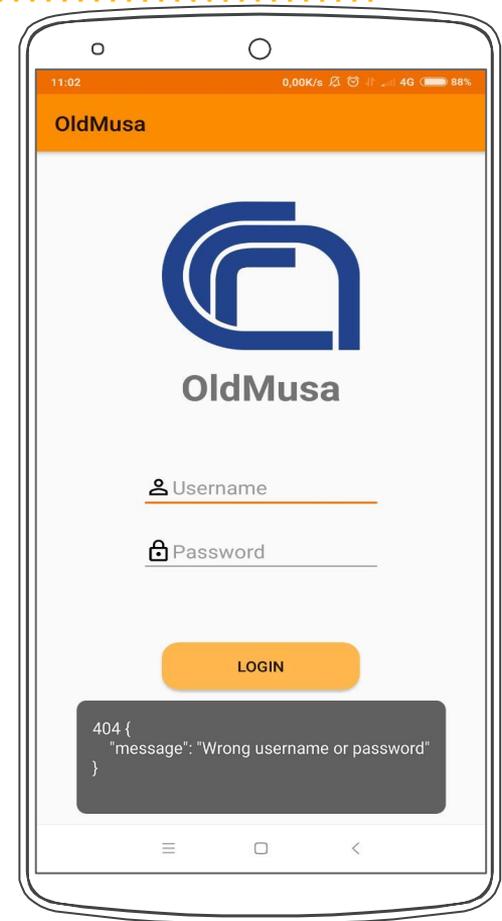
# Schermata App

La prima schermata è per **identificazione** dell'utente nell'App.

Da questa, i successivi layout verranno caricati in maniera diversa se si è un **Admin** o semplice **user**.

Infatti l'Admin ha tutti i permessi e ha il potere di cambiare i contenuti delle tabelle del DB. In aggiunta può anche dare i permessi di visualizzazione a singoli users riferiti a precisi musei.

→ PopUp di errore nel caso nome o password sono errati.



**2**

**Home**

# Schermata App

## Vista dall' User

- **Elenco dei Musei** che l'user ha il permesso di vedere.

## Vista dall' Admin

- **Elenco completo dei musei**
- **Bottone Aggiunta museo** al DataBase, inserendo **nome** e **idCnr** del museo nuovo.



3

**NavigationBar**

# Schermata App

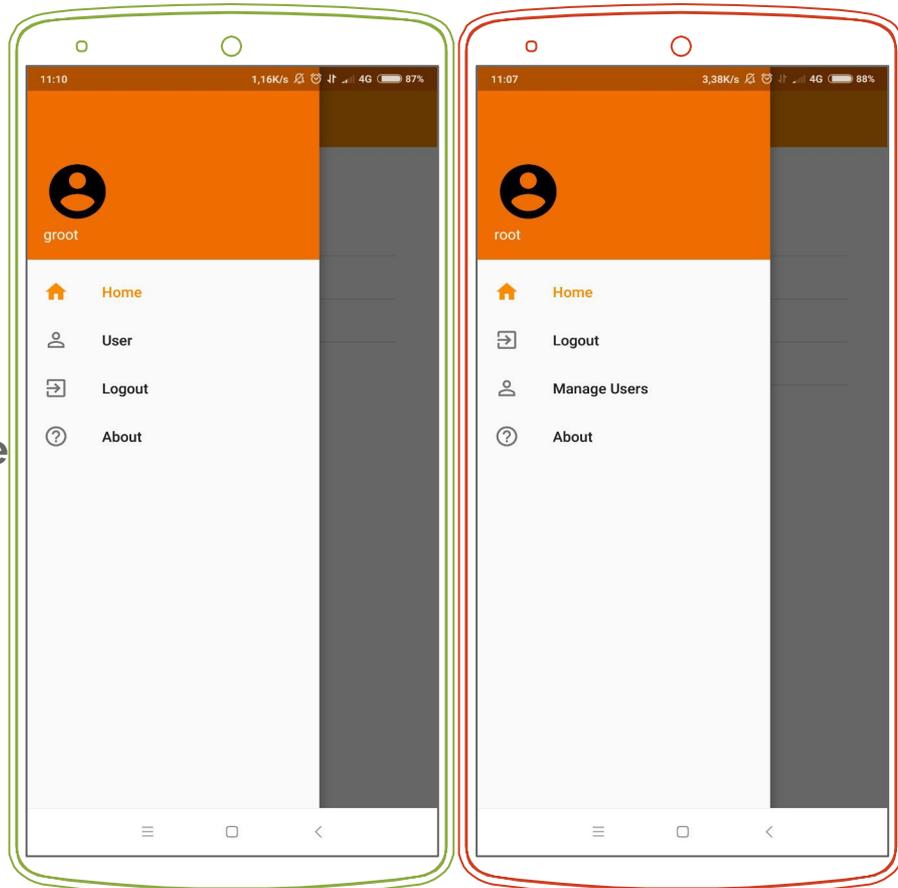
È un menu accessibile in questo e nei successivi layout.

## Vista dall' User

- **Home**: torna alla Home.
- **Logout**: per uscire dal profilo.
- **User**: per cambiare il proprio **nome user** o **password**.

## Vista dall' Admin

- **Home**: torna alla Home.
- **Logout**: per uscire dal profilo.
- **Manage users**: porta alla schermata per la gestione utenti.



# Schermata App

Premendo su User/Manage Users:

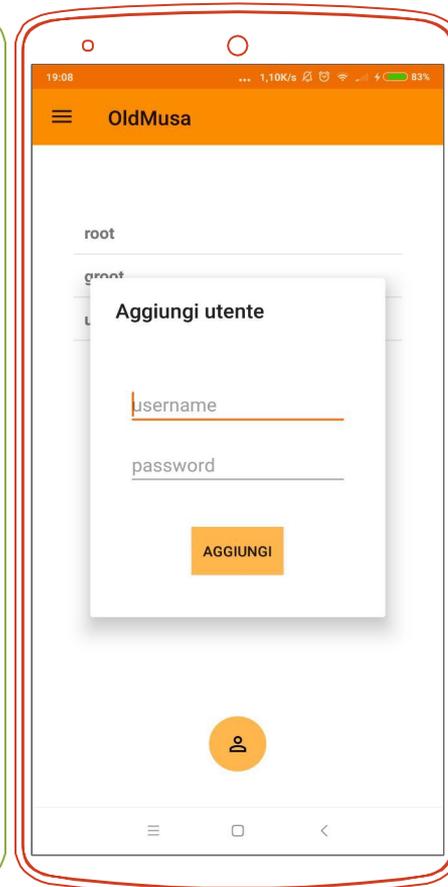
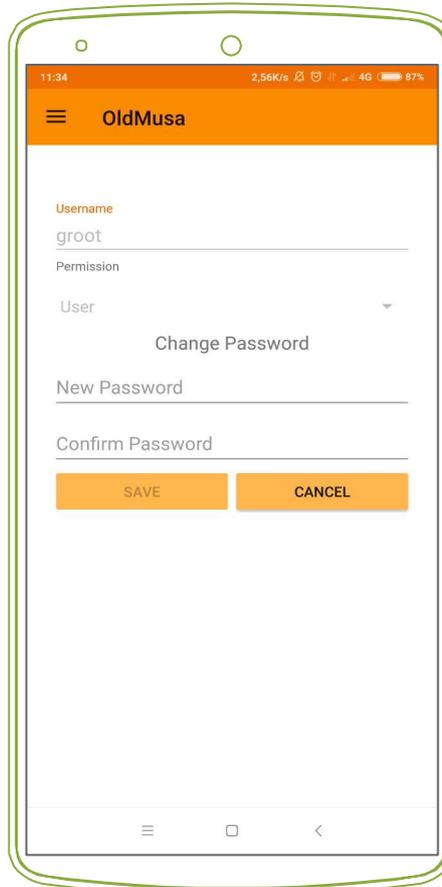
## Vista dall' User

- **Cambia username**
- **Cambia password**
- **Cancella utente**

## Vista dall' Admin

Premendo su **Manage users** si vede:

- **Elenco di tutti gli utenti,**
- **Aggiunta utente:**
  - con **username** e **password**

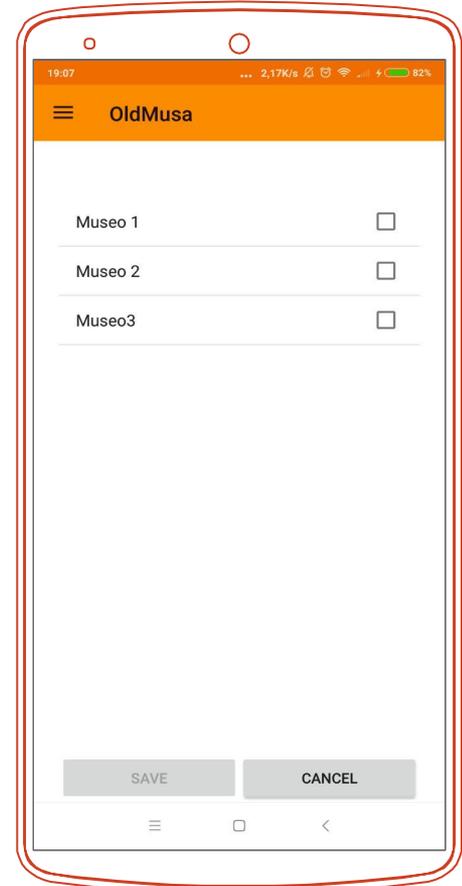


# Schermata App

## Vista dall' Admin

Premendo un utente, appare la schermata per modificare:

- **Username**
- **Stato User oppure Admin**
- **Password**
- **Dare accessi** al singolo utente per vedere certi siti



4

**Sito**

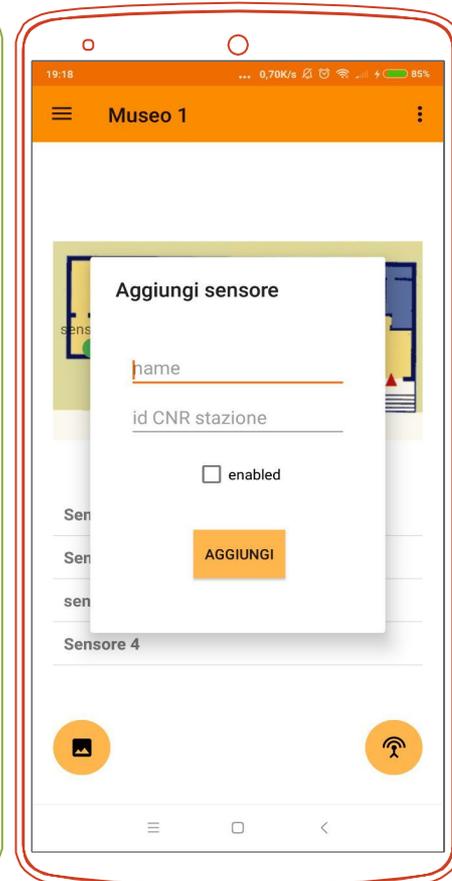
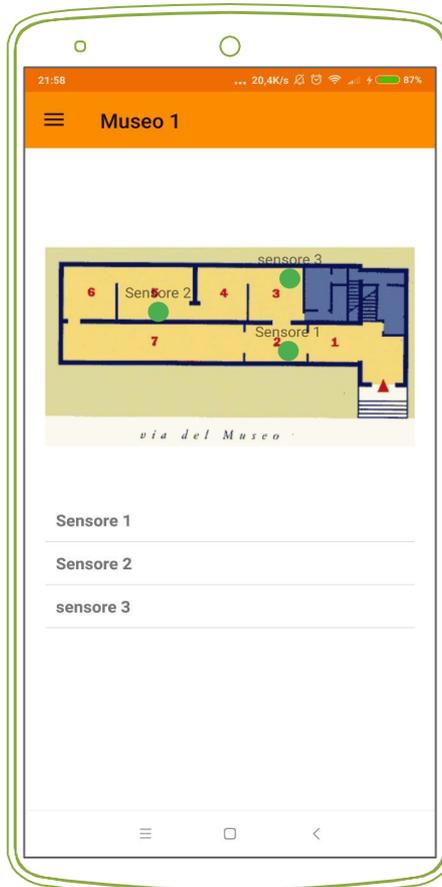
# Schermata App

## Vista dall' User

- **Mappa del museo**
- **Lista dei sensori:** nel museo.

## Vista dall' Admin

- **Mappa museo, Lista dei sensori**
- **Aggiunta mappa:** immagine viene scelta dalla galleria del cellulare.
- **Aggiunta sensore:**
  - **Nome sensore, idCnr**
  - **Enabled:** se acceso o spento
- **Menu:**
  - **Eliminazione museo**
  - **Modifica museo:** cambiare nome, idCnr.



# Led sulla Mappa Museo

Rappresentano i sensori:

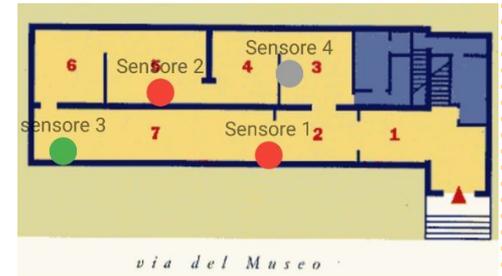
- **Led Verde:** Sensore acceso.
- **Led Rosso:** Sensore ha rilevato errori negli ultimi valori.
- **Led Grigio:** Sensore spento.
- Posizione: Indica la posizione del museo del sensore.
- Se si preme sopra al Led:
  - si passa alla visualizzazione del suo Grafico dei valori.
- Se si tiene premuto sul Led:

## Vista dall' User

- Si passa alla schermata con il suo Grafico dei valori.

**Vista dall' Admin:** premendo sopra al Led, può:

- **Move:** trascinare il Led per la mappa per cambiare la sua posizione.
- **Open:** passa alla schermata con il suo Grafico





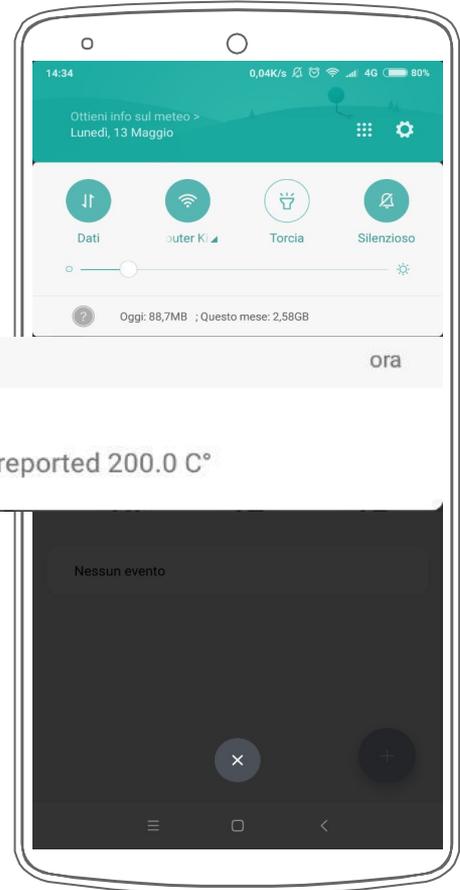
# Allarmi

# Notifica su Smartphone

In caso di allarme di un sensore, arriva una notifica:

Vista dall' User e Vista dall' Admin

- **Notifica di allarme:**
  - Nome museo
  - Nome sensore
  - Valore riportato fuori range



**5**

**Sensore**

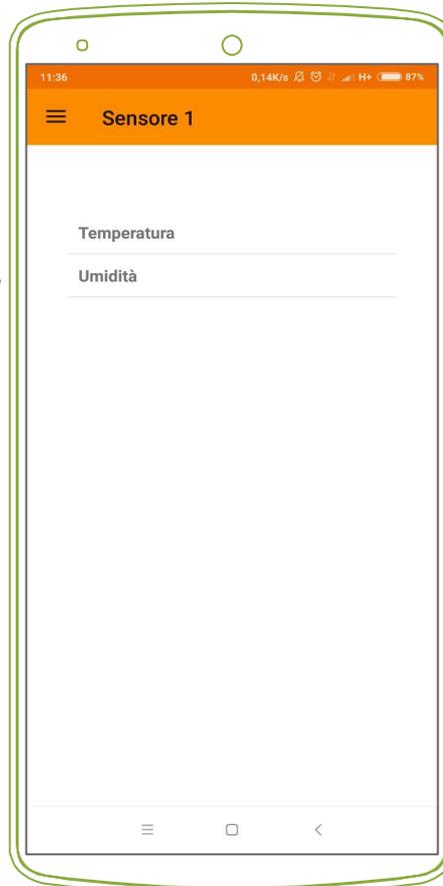
# Schermata App

## Vista dall' User

- **Elenco dei canali:** selezionando un canale si passa al suo Grafico valori.

## Vista dall' Admin

- **Elenco dei canali**
- **Aggiunta canale:** inserendo,
  - **Nome**
  - **Unità di misura**
  - **idCnr**
  - **Range** di valori accettabili
- **Menu:**
  - **Eliminazione sensore**
  - **Modifica sensore:** cambiare **nome** o **idCnr** del sensore.



6

**Canale e grafico**

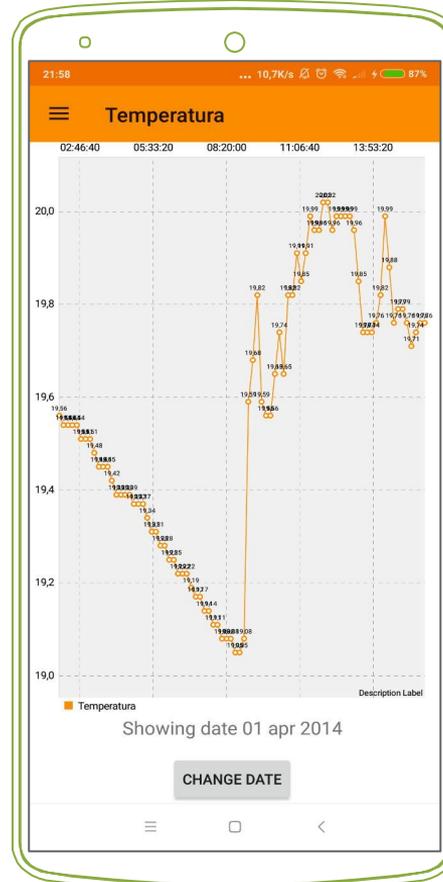
# Schermata App

## Vista dall' User

- **Grafico dei valori** del canale scelto
- **Scelta del giorno** da voler vedere; calendario a disposizione

## Vista dall' Admin

- **Grafico dei valori** del canale
- **Scelta del giorno** da voler vedere
- **Menu:**
  - **Elimina canale**
  - **Modifica canale:** cambiare nome, unità di misura, idCnr o range.

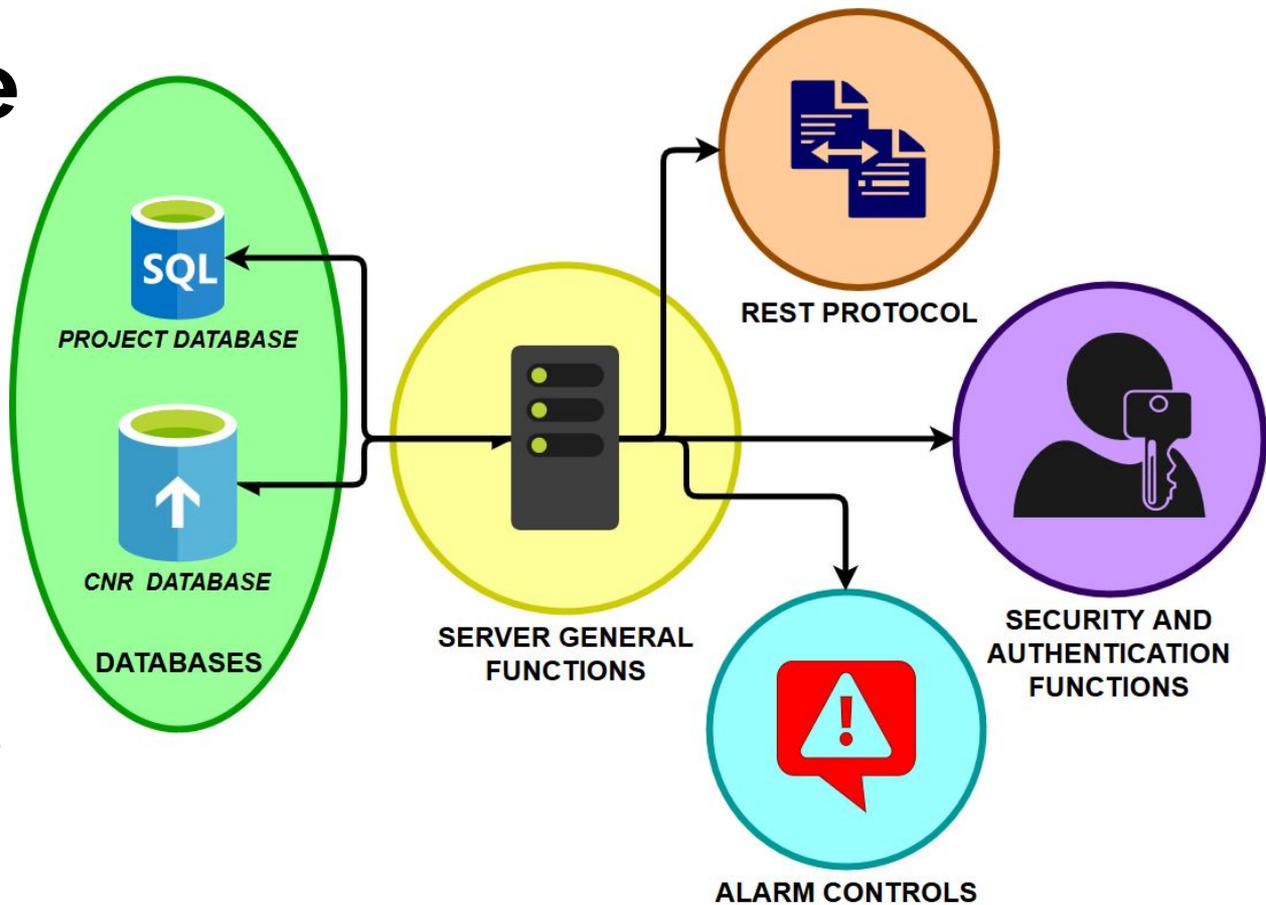




# Server

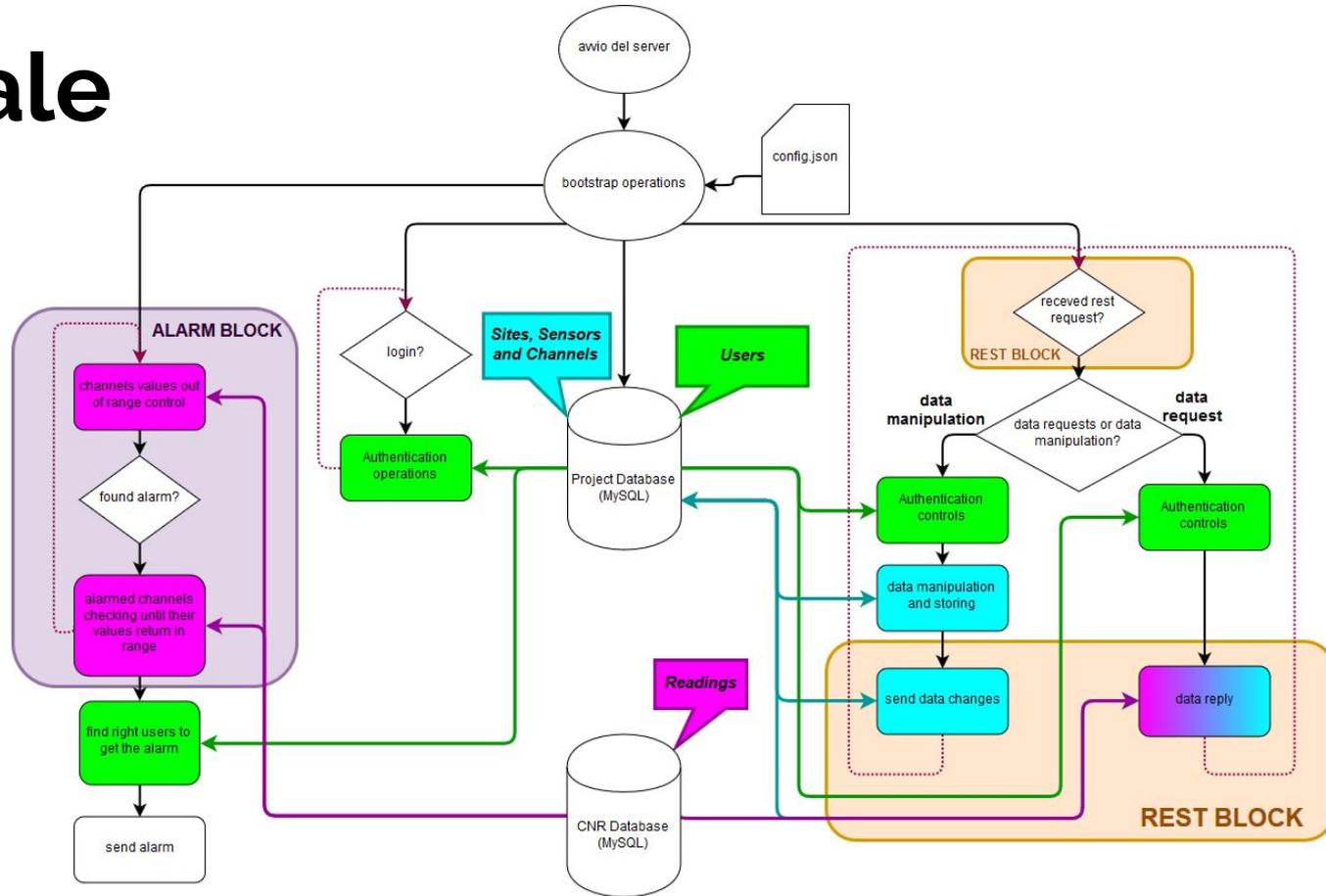
# In Generale

- **Database:** letture prese dal DB del cnr
- **REST:** protocollo di comunicazione con l'app
- **Autenticazione:** gestione degli utenti, login e assegnazione dei permessi
- **Allarmi:** controllo periodico delle letture e invio dell'allarme all'app



# In Generale

Il **database** interno al progetto permette di conoscere **siti, sensori, canali** e gli **utenti**. Il **server**, oltre a controllare costantemente la presenza di **allarmi**, permette di effettuare il **login**, la **modifica** dei dati nel database e l'**invio** degli stessi (se si hanno i permessi)



**1**

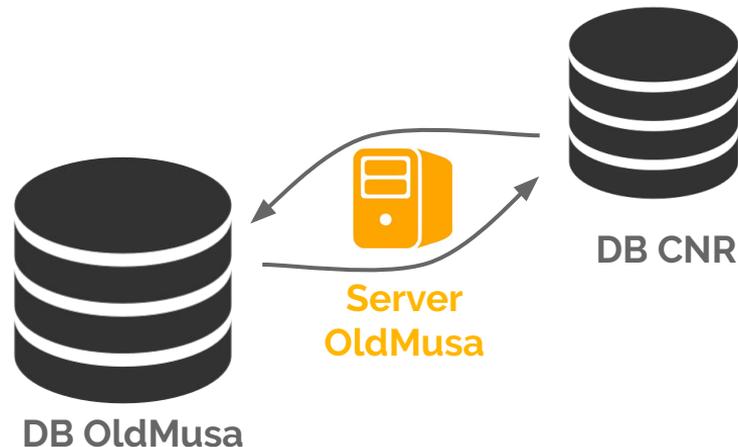
**Database**



# Relazione **DB CNR** e **DB OldMusa**

Il Database che abbiamo creato è del tutto indipendente da quello del CNR, che serve solo come supporto per le letture dei canali.

Il riempimento del database OldMusa avviene attraverso l'app.

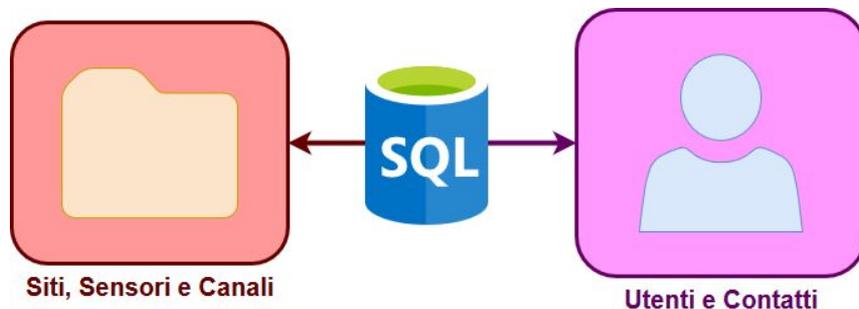


# Dal Database del CNR

Dal database del CNR si accede in **sola lettura** solo alla relazione t\_rilevamento\_dati, in cui sono presenti tutte le **letture** di tutti i canali

#	idsito	idstanza	idstazione	idsensore	canale	valore_min	valore_med	valore_max	scarto	data	errore	misura	step
1	RA_ECP	Corridoio 1P	13061840	06	11	19.56	19.56	19.56	0	2014-04-01 00:00:00	N	T	10
2	RA_ECP	Corridoio 1P	13061840	06	12	46.2	46.2	46.2	0	2014-04-01 00:00:00	N	UR	10
3	RA_ECP	Esterno	13061840	01	1	10.32	10.34	10.35	0.02	2014-04-01 00:00:00	N	T	10
4	RA_ECP	Esterno	13061840	01	2	47.1	47.25	47.4	0.15	2014-04-01 00:00:00	N	UR	10
5	RA_ECP	Sala1	13061840	05	10	48.3	48.3	48.3	0	2014-04-01 00:00:00	N	UR	10
6	RA_ECP	Sala1	13061840	05	9	19.17	19.17	19.17	0	2014-04-01 00:00:00	N	T	10
7	RA_ECP	Sala2	13061840	02	3	20.58	20.6	20.61	0.01	2014-04-01 00:00:00	N	T	10
8	RA_ECP	Sala2	13061840	02	4	44.8	44.85	44.9	0.05	2014-04-01 00:00:00	N	UR	10
9	RA_ECP	Sala2	13061840	03	5	20.19	20.19	20.19	0	2014-04-01 00:00:00	N	T	10
10	RA_ECP	Sala2	13061840	03	6	49.4	49.4	49.4	0	2014-04-01 00:00:00	N	UR	10
11	RA_ECP	Sala3	13061840	04	7	19.92	19.92	19.92	0	2014-04-01 00:00:00	N	T	10
12	RA_ECP	Sala3	13061840	04	8	45	45	45	0	2014-04-01 00:00:00	N	UR	10
13	RA_ECP	Corridoio 1P	13061840	06	11	19.54	19.55	19.56	0.01	2014-04-01 00:10:00	N	T	10

# Il database del Server



Il database interno al server è diviso in **due blocchi** principali:

- La **sensoristica** (Siti, Sensori e Canali), per identificare mezzi e luoghi fisici
- L' **autenticazione** (users e contatti), strettamente legata all'app e ai permessi di ogni utente.

# Sensoristica: i Siti

#	id	name	id_cnr
1	12	Museo 1	PI_CM
2	13	Museo 2	1002
3	14	Museo3	CnrMuse3

La relazione contiene le **informazioni di base** per i **siti**:

- **Id**: numero univoco che identifica il sito
- **Name**: nome del sito
- **Id\_cnr**: id che identifica il sito all'interno del DB del CNR (per compatibilità)

# Sensoristica: i Sensori

La relazione contiene le **informazioni utili** di tutti i **sensori**:

- **Id**: identifica il sensore
- **Site\_id**: riferimento al sito
- **Id\_cnr**: id del sensore nel DB del CNR (per compatibilità)
- **Name**: nome del sensore

- **loc\_x/y**: coordinate del sensore nel sito
- **enabled**: flag per dell'attivazione
- **status**: diverso da "ok" in caso di allarmi

#	id	site_id	id_cnr	name	loc_x	loc_y	enabled	status
1	3	12	7824	Sensore 1	430	240	1	[4, 4] fired
2	5	12	7824	Sensore 2	239	322	1	[12, 12] fired
3	10	14	Jfzirz	S1	218	431	0	ok
4	11	14	Kgxotx	S2	295	466	0	ok
5	14	12	Se_1123567	sensor 3	68	245	1	ok
6	22	12	SA_000	Sensore 4	466	347	0	ok

# Sensoristica: i Canali

- **Name**: nome o descrizione
- **Measure\_unit**: unità di misura del canale
- **Range\_min/max**: range dei valori accettati

Come per i sensori, sono contenuti tutti i dati dei **canali**:

- **Id**: identifica il canale
- **sensor\_id**: riferimento al sensore in cui è contenuto
- **Id\_cnr**: id del canale nel DB del CNR (per compatibilità)

id	sensor_id	id_cnr	name	measure_unit	range_min	range_max
1	3	07	Temperatura	C°	10	100
4	3	08	Umidità	%	20	40
12	5	10	Umidità	%	20	80
13	5	09	Temperatura	C°	14	20

# Autenticazione: gli Utenti

La relazione contiene i **dati di base** per gli utenti:

- **id**: identifica l'utente
- **password\_hash**: password criptata
- **Last\_password\_change**: data dell'ultima modifica della password
- **Permission**: flag per admin o user normale

#	id	username	password_hash	last_password_change	permission
1	1	root	\$6\$rounds=656000\$6MuLaIKkZD5...	1558882298	A
2	2	groot	\$6\$rounds=656000\$2.epFd2Rrlyitj...	1556725153	U
3	3	user1	\$6\$rounds=656000\$.tuJXmu3Zt20...	1557409562	U

# Autenticazione: i Contatti

#	registration_id	user_id
1	c12UTpN67FQ:APA91bGze5EnQp...	1
2	c7rQuNy-nJ0:APA91bHzVktzN4jD...	1
3	c8GDCzdLPqs:APA91bHh7crQdlp...	1
4	cGmpP_x_KC4:APA91bG2xQyVB...	1
5	cIMem6l8D0l:APA91bEhrLdFhWO...	1
6	cWykzI_Rd48:APA91bFMSh1BHiV...	1

Agli utenti vengono affiancati i suoi **recapiti telefonici**:

- **registration\_id**: stringa privata che contiene un recapito telefonico di un user
- **user\_id**: riferimento ad uno user

**2**

**REST**



# Protocollo **REST**

Usato per la comunicazione tra **server** (backend) e **applicazione** (frontend).

La documentazione ufficiale è disponibile sul sito github, sotto riportato.

É possibile fare un test di prova senza l'uso dell'applicazione per comprendere meglio cosa succede dietro alle quinte.

```
→ curl -X GET \  
  -H "Token: $TOKEN" \  
  ${API_URL}/site/23  
{  
  "id": 23,  
  "name": "Doc Site",  
  "id_cnr": "DS-XL"  
}
```

<https://github.com/OldMusa-5H/OldMusaDoc>



# Protocollo REST

```
→ curl -X GET \  
    "${API_URL}/token?username=root&password=password"  
{  
  "token": "eyJhbGciOiJIUzUxMiJ9.eyJpZCI6MSwiZGF0ZSI6
```

Ok, quindi iniziamo a fare qualche query!

Per prima cosa dobbiamo prendere il **token** specificando username e password.

<https://github.com/OldMusa-5H/OldMusaDoc>



# Protocollo **REST**

```
→ curl -X GET \  
  -H "Token: $TOKEN" \  
  ${API_URL}/site  
  
[  
  12,  
  13,  
  14,  
  23  
]
```

Una volta ottenuto il token possiamo listare **tutti i siti** visibili, come fa l'applicazione appena entriamo.

```
→ curl -X GET \  
  -H "Token: $TOKEN" \  
  ${API_URL}/site/12  
  
{  
  "id": 12,  
  "name": "Museo 1",  
  "id_cnr": "RA_ECP"  
}
```

Dopo possiamo ottenere i **dettagli** di ogni sito.



# Protocollo REST

```
→ curl -X GET \  
  -H "Token: $TOKEN" \  
  ${API_URL}/site/12/sensor  
  
[  
  3,  
  5,  
  14  
]
```

Possiamo ora "entrare" nel sito e otteniamo l'elenco completo dei **sensori**.

```
→ curl -X GET \  
  -H "Token: $TOKEN" \  
  ${API_URL}/sensor/3  
  
{  
  "id": 3,  
  "site_id": 12,  
  "id_cnr": "13061840",  
  "name": "Sensore 1",  
  "loc_x": 430,  
  "loc_y": 240,  
  "enabled": true,  
  "status": "ok"  
}
```

Specificando il sensore otteniamo tutti i suoi **dettagli**.



# Protocollo REST

Possiamo fare in ugual modo con i canali.

```
→ curl -X GET \  
  -H "Token: $TOKEN" \  
  ${API_URL}/sensor/3/channel  
  
[  
  1,  
  4  
]
```

“Entriamo” nel sensore e otteniamo l'elenco completo dei **canali**.

```
→ curl -X GET \  
  -H "Token: $TOKEN" \  
  ${API_URL}/channel/1  
  
{  
  "id": 1,  
  "sensor_id": 3,  
  "id_cnr": "11",  
  "name": "Temperatura ",  
  "measure_unit": "C\u00b0",  
  "range_min": "10",  
  "range_max": "100"  
}
```

Specificando il canale otteniamo tutti i suoi **dettagli**.



# Protocollo REST

Finalmente possiamo ottenere i **valori** rilevati dal canale nel museo, specificando data di inizio e di fine.

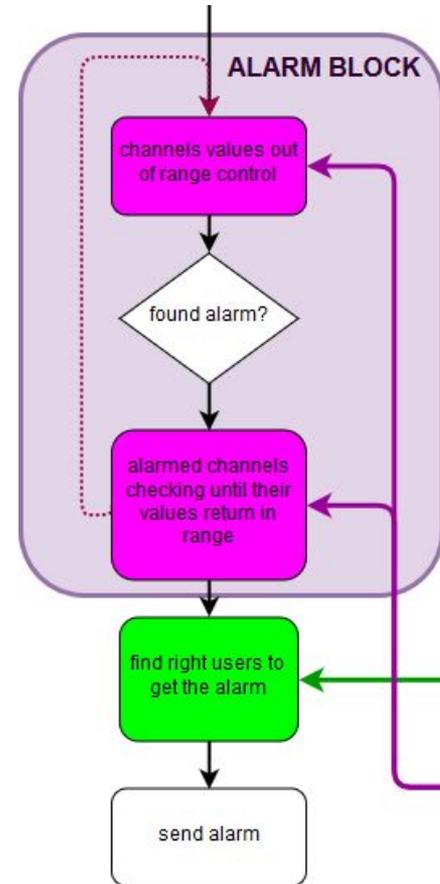
```
→ curl -X GET \  
  -H "Token: $TOKEN" \  
  "${API_URL}/channel/1/readings?start=2014-03-31T22:00:00.000Z&  
end=2014-04-01T00:20:00.000Z&precision=atomic"  
[  
  {  
    "date": "2014-04-01T00:00:00.000000Z",  
    "value_min": "19.56",  
    "value_avg": "19.56",  
    "value_max": "19.56",  
    "deviation": "0.0",  
    "error": "N"  
  },  
  {  
    "date": "2014-04-01T00:10:00.000000Z",  
    "value_min": "19.54",  
    "value_avg": "19.55",  
    "value_max": "19.56",  
    "deviation": "0.01",  
    "error": "N"  
  },  
  {  
    "date": "2014-04-01T00:20:00.000000Z",  
    "value_min": "19.54",  
    "value_avg": "19.55",  
    "value_max": "19.56",  
    "deviation": "0.01",  
    "error": "N"  
  }  
]
```

**3**

**Allarmi**

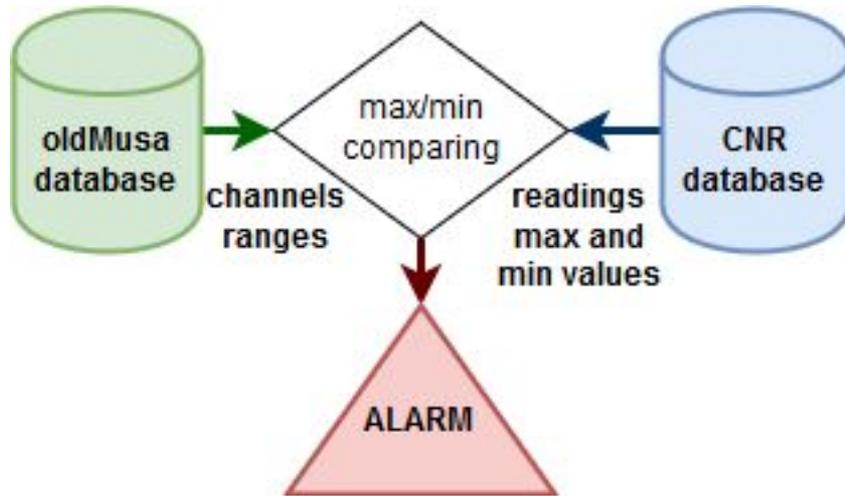
# Gli Allarmi

Il sistema di **allarmi** permette un **controllo periodico** dei valori letti da ogni canale e, in caso si trovi un'anomalia, è capace di inviare una **notifica** ai diretti interessati





# Gli Allarmi: il controllo

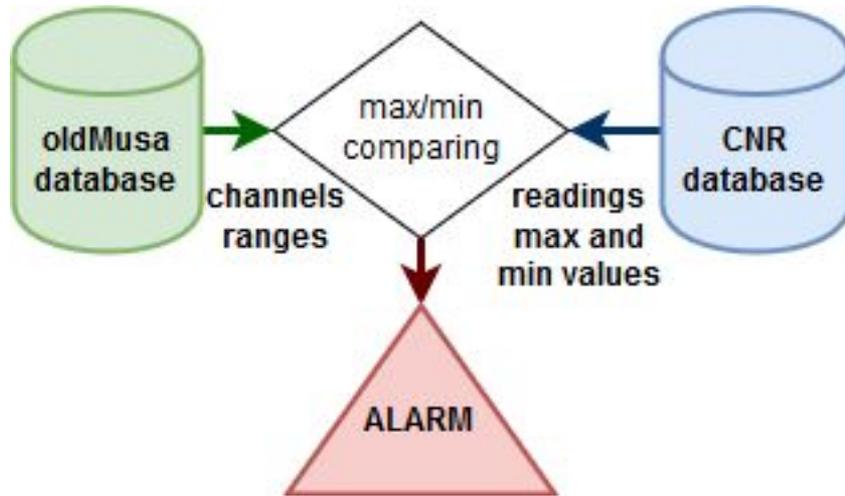


Il sistema di **controllo** legge tutti i dati non ancora controllati e ritorna i valori **massimi e minimi** letti da ogni canale.

Per sapere quali record non sono ancora stati controllati, viene tenuto l'orario dell'ultimo controllo.



# Gli Allarmi: il controllo



Successivamente vengono **confrontati** i massimi e minimi letti con il **range** dello stesso canale.

Se un valore esce dal range viene lanciato un **allarme**.

# Gli Allarmi: l'allarme



Quando viene lanciato un allarme, si invia una **notifica** agli utenti che possono accedere al sito in cui avviene si è verificata l'anomalia.

Contemporaneamente si **controllano** separatamente **i canali** in cui si è verificata l'allarme. Non appena i valori rientrano nel range l'allarme viene ritirata

4

**Sicurezza**



# Sicurezza

Ovviamente i **controlli di sicurezza** non sono fatti solamente dall'applicazione.

Ogni controllo è effettuato anche dal server per assicurarsi che non siano possibili attacchi.

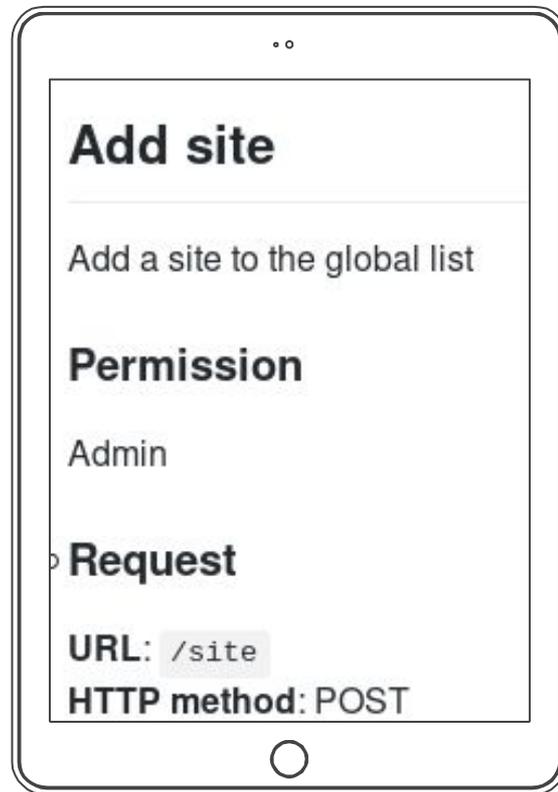
```
~ via ● v11.15.0
→ curl -X GET \
    -H "Token: $ADMIN_TOKEN" \
    ${API_URL}/site/23
{
  "id": 23,
  "name": "Doc Site",
  "id_cnr": "DS-XL"
}

~ via ● v11.15.0
→ curl -X GET \
    -H "Token: $USER_TOKEN" \
    ${API_URL}/site/23
{
  "message": "Cannot find site 23"
}
```

# Sicurezza

Nella documentazione è riportato il **permesso** richiesto per ogni query.

Per assicurare la sicurezza, il server ha vari test in cui prova ad accedere a url protetti senza i permessi richiesti, assicurandosi che dia l'errore corretto.





# Thanks!

## Altre domande?

Ringraziamo il **CNR** di averci dato l'opportunità di poter svolgere un progetto mentre stiamo ancora frequentando le scuole superiori. Abbiamo così potuto simulare un vero progetto di lavoro. Ringraziamo anche i referenti con i quali abbiamo relazionato e i quali si sono dimostrati disponibili alle nostre richieste di materiale utile e alle nostre domande riguardanti il prodotto finale.



# Elenco slide

1. Intro generale
  - a. Obiettivo
  - b. Team di sviluppo
2. Schema progetto
  - a. Funzionamento
  - b. Database CNR VS Database OldMusa
3. Login
  - a. App
  - b. Server
  - c. Sicurezza & database
4. Home (NavigationBar)
5. Museum
6. Sensori con canali
7. Grafico
8. Ringraziamenti al CNR