LS IoT Platform

Piattaforma per il monitoraggio di macchine utensili con integrazione a software ERP Microsoft Dynamics NAV

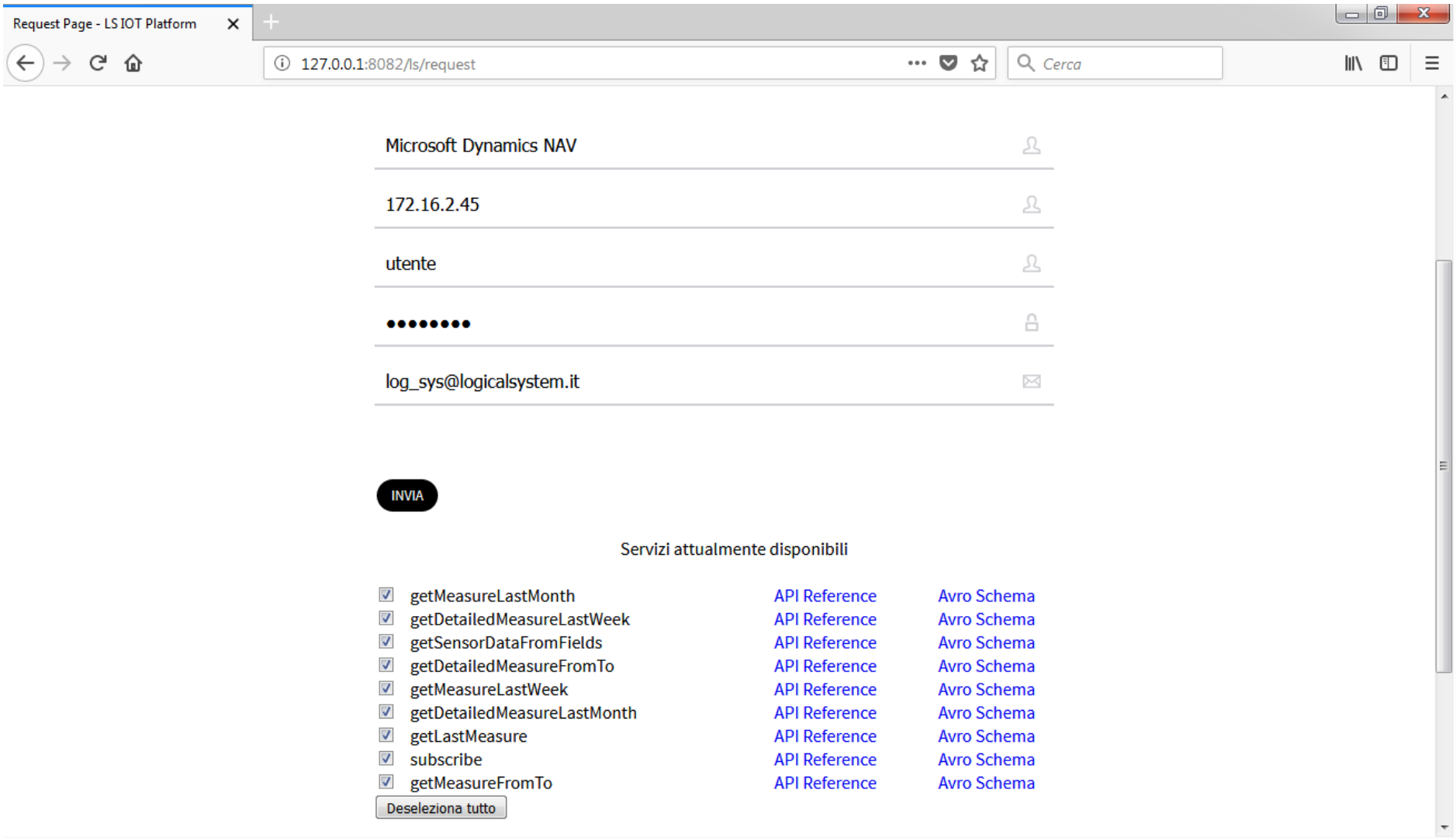
Vincenzo Nucci e Matteo Tiberi

Università di Camerino

LS IoT Platform

- Espone servizi REST per letture di sensori
  - Ultima lettura di un sensore
  - Letture di una certa settimana
  - Letture di un certo mese
  - Letture di determinati campi dei sensori
- Servizio di sottoscrizione con notifiche PUSH
  - Debolmente accoppiato grazie ad Apache ActiveMQ
- Integra il servizio di sottoscrizione con Dynamics NAV
  - Utilizzo dei web services SOAP offerti da NAV
- Indipendente da sorgenti dati e formato dei dati
  - Grazie alle interfacce e Apache Avro
- Interfaccia web
  - Pagina per la registrazione delle applicazioni
  - Pagina per la gestione delle richieste
  - Pagina per la gestione dei servizi attivi per gli utenti

Pagina web richiesta token



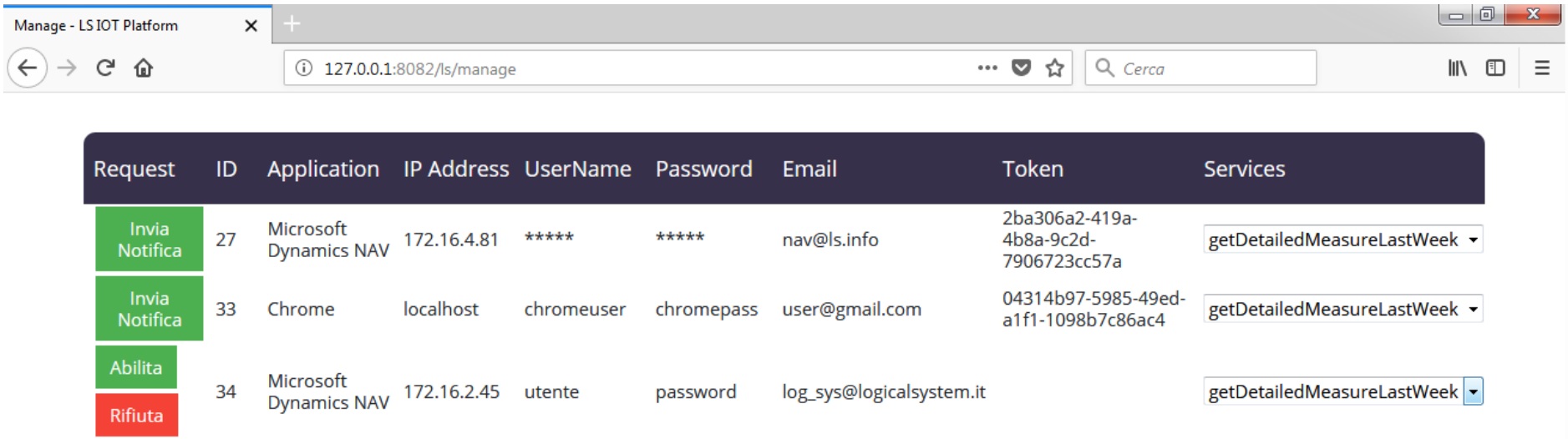
Gestione delle richieste nel database

Table: Requests

New Record | Delete Record

	ID	AppName	AppIp	Username	Password	Email	Token	Services
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	27	Microsoft Dyn...	172.16.4.81	*****	*****	nav@ls.info	2ba306a2-41...	1,2,3,4,5,6,7,8,9,10,11
2	33	Chrome	localhost	chromeuser	chromepass	user@gmail.c...	04314b97-59...	1,2,3,4,5,6,7,8,9,10,11
3	35	a	a	a	a	a	a7bb18cc-482...	1,2,3,4,5,6,7,8,9,10,11

Pagina web gestione delle richieste



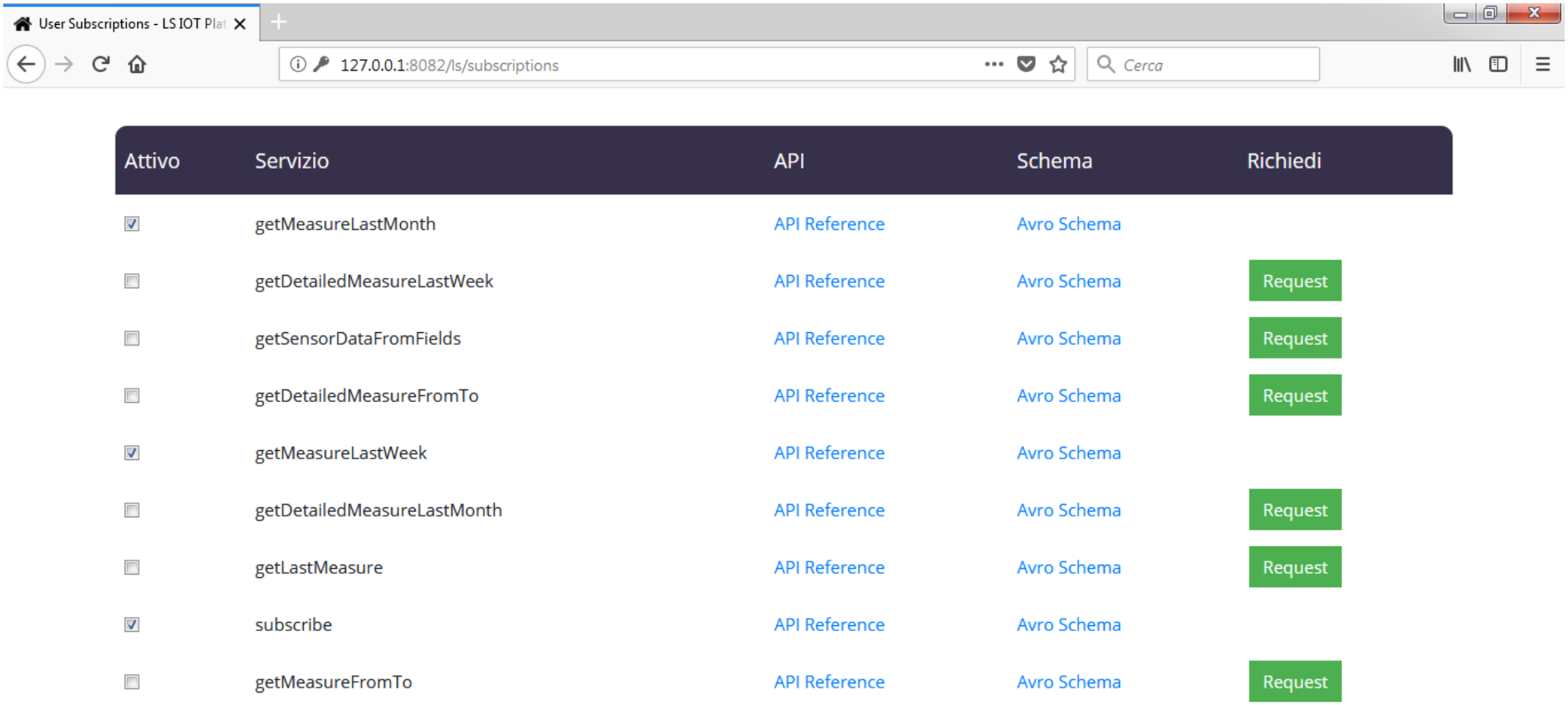
Sottoscrizioni database

Table: Subscriptions

New Record | Delete Record

	Token	ActiveServices	ServicesUsage	AppName	AppIp	Username	Password
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	a7bb18cc-482...	1,2,3,4,5,6,7,...	0,0,0,0,0,0,0,0,0,0,0	a	a	a	a
2	2ba306a2-41...	1,2,3,4,5,6,7,...	2,0,0,0,55,5,0,8,136,1,0	Microsoft Dyn...	172.16.4.51	*****	*****
3	04314b97-59...	1,2,3,4,5,6,7,...	0,0,0,0,0,0,0,0,0,0,0	Chrome	localhost	chromeuser	chromepass

Pagina web gestione dei servizi utenti



Esempio di un servizio

```
@Path("getlastmeasure/{sensorId}/{annotation}")
@GET
@Produces(MediaType.APPLICATION_JSON)
public String getLastMeasure(@HeaderParam("token") String token, @PathParam("sensorId") int sensorId,
        @PathParam("annotation") boolean annotation)
{
    try {
    if (!checkToken(token, "getLastMeasure"))
        throw new IllegalArgumentException("Token not valid/not authorized");

    Connection con;
    String query = "";
    System.out.println("GET last measure of sensor: " + sensorId);

    con = ((AbstractSQLConnection)conf.getProperty("mysql")).connect();
        if(annotation)
            query = "select tvmgenio.*,measann.idvalue, measann.prefname_meas, measann.prefprefix_name from "
                    + "tvplcset left join tvplcfam using(famplc) left join tvmgenio using (canplc) "
                    + "left join measann using (famplc) where (canplc="+sensorId+")";
        else
            query = "select * from tvmgenio where (canplc = "+sensorId+")";

        return (String)getDataFromDbToSensorList(con, query, new SensorData(), new SensorDataList().getListContainer()
                ,annotation,true);


    } catch (Exception e) {
        System.out.println("Error found: " + e.getMessage());
        return new ExceptionMessageHandlerBuilder(e)
                .build().toString();
    }
}
```

Valori di ritorno di getlastmeasure

GET 127.0.0.1:8082/ls/iot/getlastmeasure/204/true Params Send Save

Pretty Raw Preview JSON Save Response


```
[
    {
        "canplc": 204,
        "canplcannotation": null,
        "numpez": 19.0864,
        "numpezannotation": {
            "prefname_meas": "kilogram",
            "prefprefix_name": "kilo"
        },
        "oldpez": 19.1876,
        "oldpezannotation": {
            "prefname_meas": "kilogram",
            "prefprefix_name": "kilo"
        },
        "nocich": 27,
        "nocichannotation": {
            "prefname_meas": "minutes",
            "prefprefix_name": "min"
        },
        "ferma": 0,
        "fermaannotation": null,
        "manaut": 0,
        "manautannotation": null,
        "allarm": 0,
        "allarmannotation": null,
        "acceso": 0,
        "accesoannotation": null,
        "valor1": 0,
        "valor1annotation": null,
        "valor2": 0,
        "valor2annotation": null,
        "valor3": 0,
        "valor3annotation": null,
```

Pagina catalogo Smart Object

Smart Objects - LS IOT Platform

127.0.0.1:8082/ls/plc

Cerca

208	3		Silos per materiale presse a iniezione	kilo	kilogram	numpez	2	Qta materiale
208	3		Silos per materiale presse a iniezione	kilo	kilogram	oldpez	3	vecchia qta silos
221	4		Silos per materiale estrusori	kilo	kilogram	numpez	2	Qta materiale
222	4		Silos per materiale estrusori	kilo	kilogram	numpez	2	Qta materiale

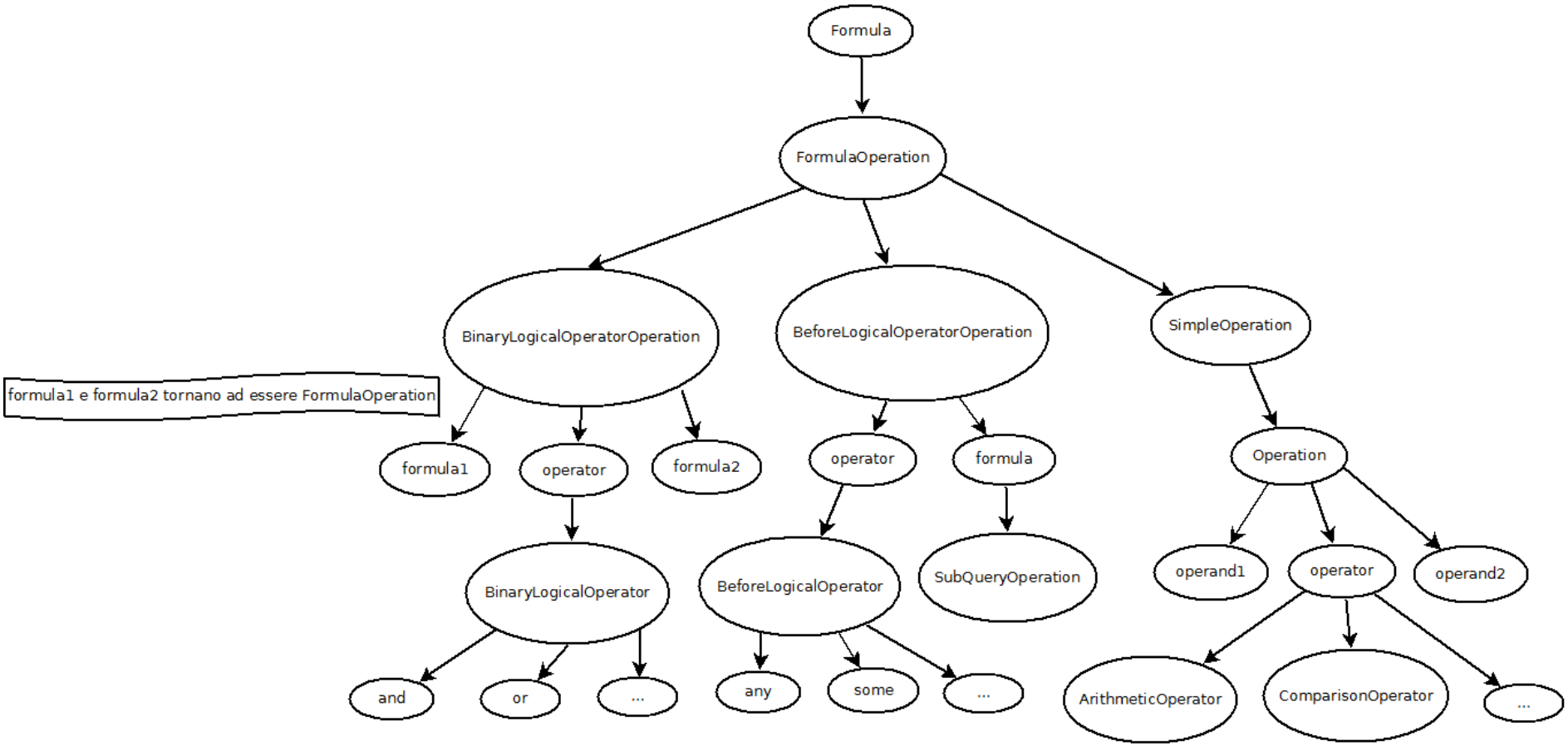
Subscribe Rule

- Messaggio JSON che il servizio di sottoscrizione si aspetta di ricevere
- L'utente specifica la condizione
  - Nome e IP applicazione
  - Coda o topic di ActiveMQ
  - Subscribe Rule
    - ★ Nome tabella da interrogare
    - ★ Lista dei campi dei quali monitorarne i cambiamenti
    - ★ Espressione CRON
    - ★ Clausola where

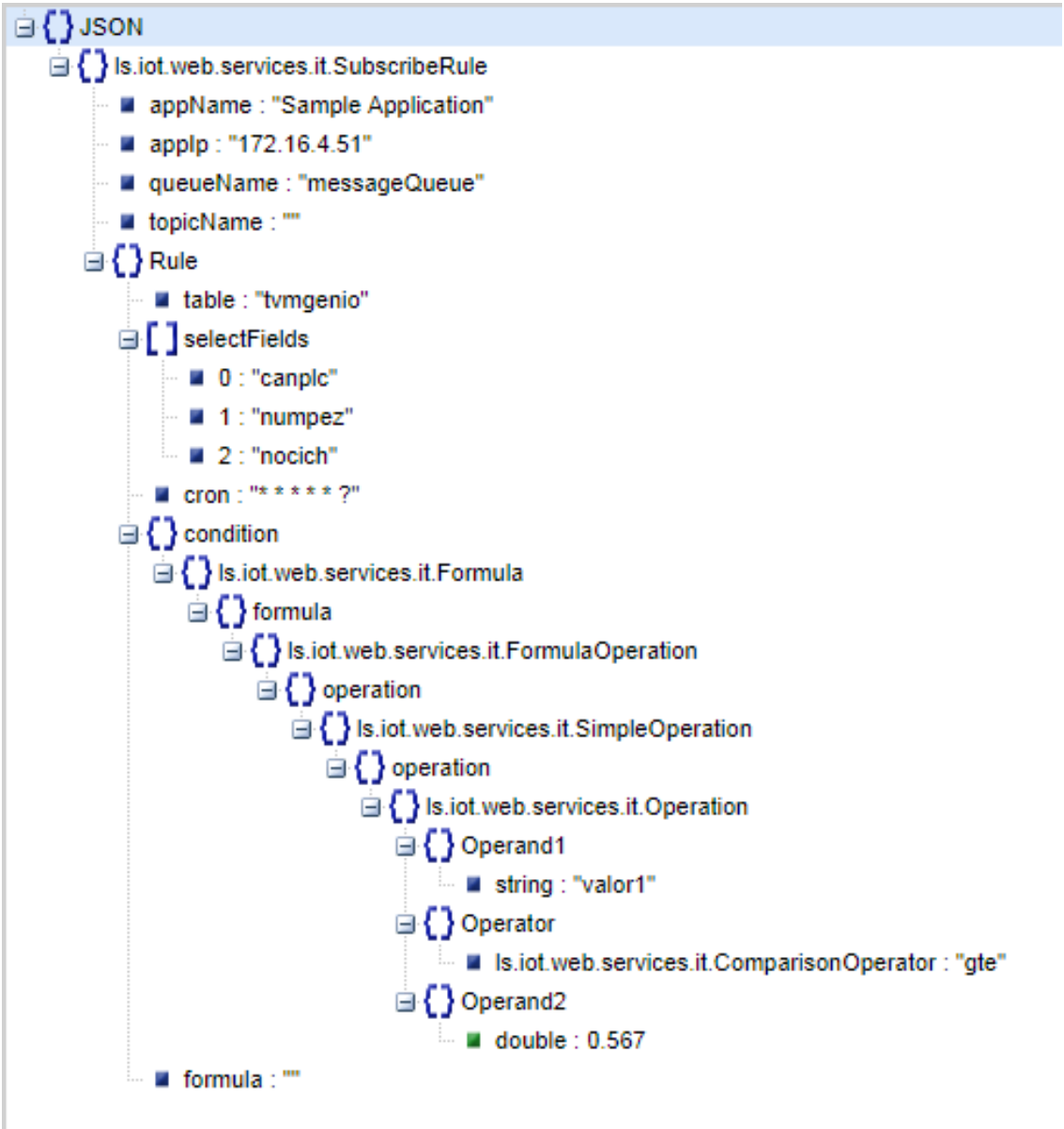
Clausola where

- Condition
  - insieme di oggetti Avro annidati
- Formula
  - Semplice stringa (per NAV)

Albero della condition



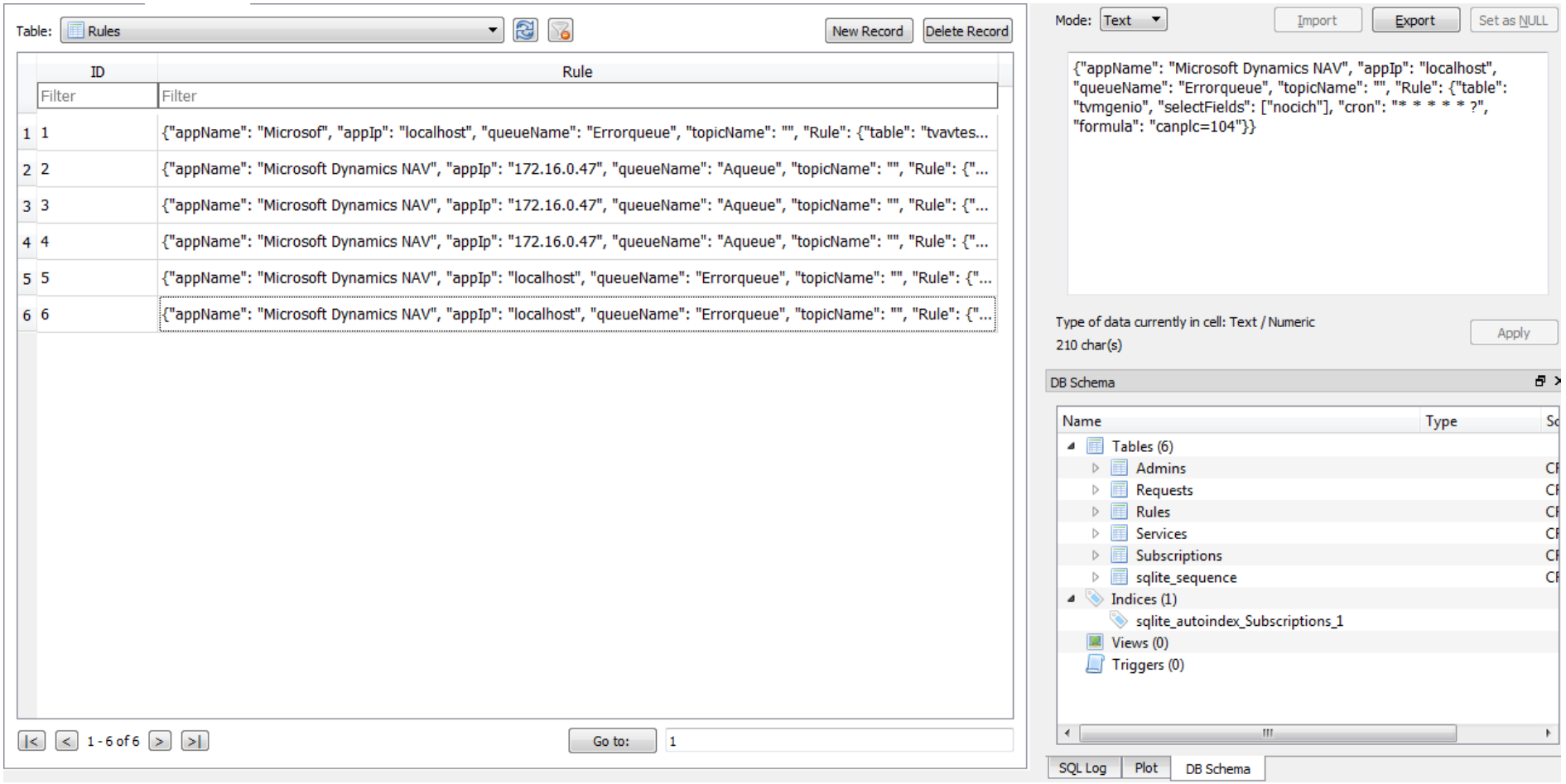
Subscribe Rule di una applicazione



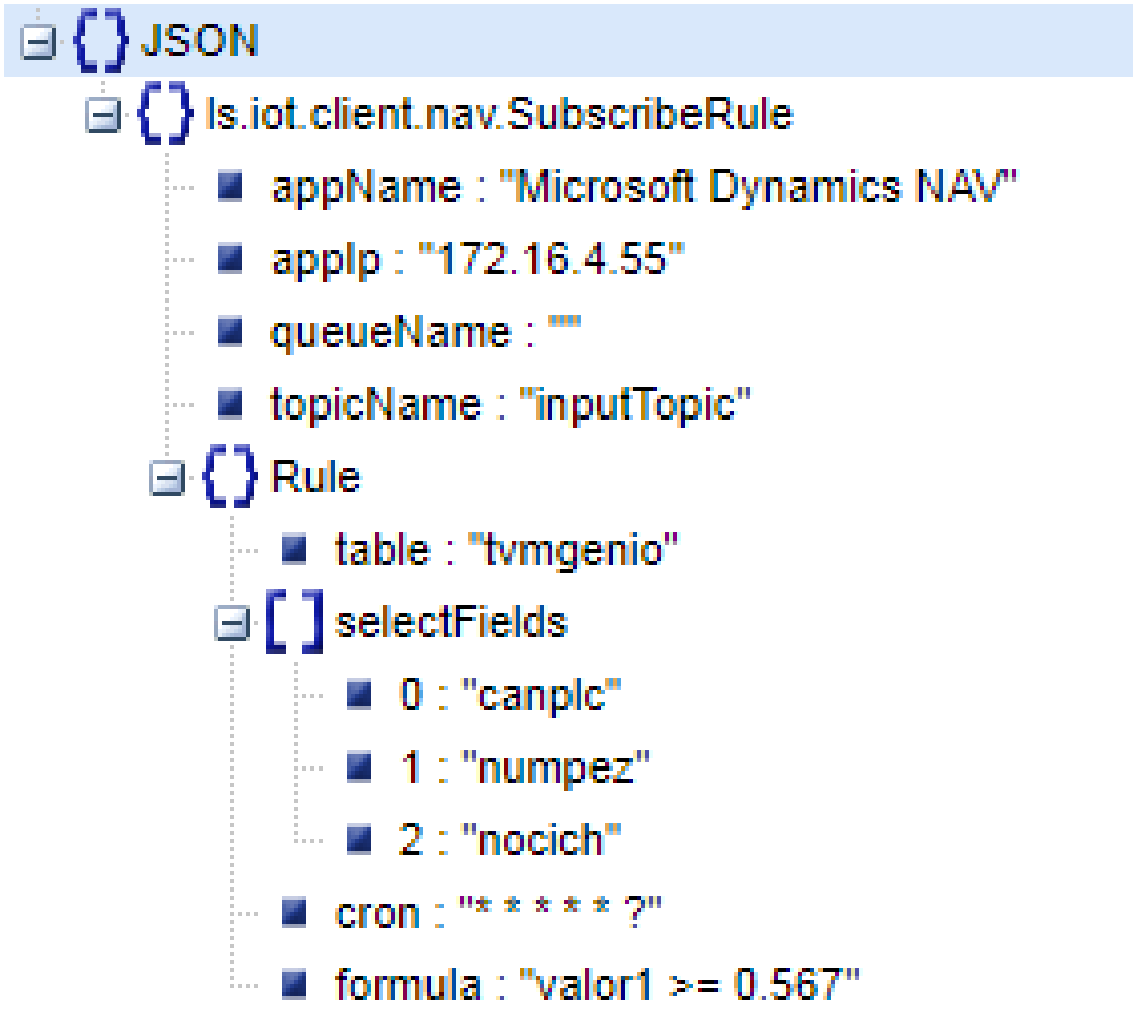
Schema Avro Subscribe Rule di una applicazione


```
{
  "namespace":"ls.iot.web.services.it",
  "type":"record",
  "name":"SubscribeRule",
  "fields":[
    {
      "name":"appName",
      "type":"string",
      "doc":"Nome dell'appliczione."
    },
    {
      "name":"appIp",
      "type":"string",
      "doc":"IP dell'applicazione."
    },
    {
      "name":"queueName",
      "type":"string",
      "doc":"Nome della coda in cui verranno inviati i nuovi dati."
    },
    {
      "name":"topicName",
      "type":"string",
      "doc":"Nome del topic in cui verranno inviati i nuovi dati."
    },
    {
      "name":"Rule",
      "type":{
        "name":"RuleRecord",
        "type":"record",
        "fields":[
          {
            "name":"table",
            "type":"string",
            "doc":"Tabella che verrà interrogata."
          },
          {
            "name":"selectFields",
            "type":{
              "type":"array",
              "items":"string"
            },
            "doc":"Campi della tabella che verranno restituiti popolati in caso di modifiche rilevate."
          },
          {
            "name":"cron",
            "type":"string",
            "doc":"Intervallo di tempo in CRON che indica ogni quanto verranno controllati nuovi dati."
          },
          {
            "name":"condition",
            "type":[
              "null",
              "Formula"
            ],
            "default":null,
            "doc":"Condizione di controllo (SQL WHERE) descritta con record annidati."
          },
          {
            "name":"formula",
            "type":"string",
            "doc":"Condizione di controllo (SQL WHERE) descritta come stringa"
          }
        ],
        "doc":"Regola di sottoscrizione al servizio."
      },
      "doc":"Regola di sottoscrizione al servizio."
```

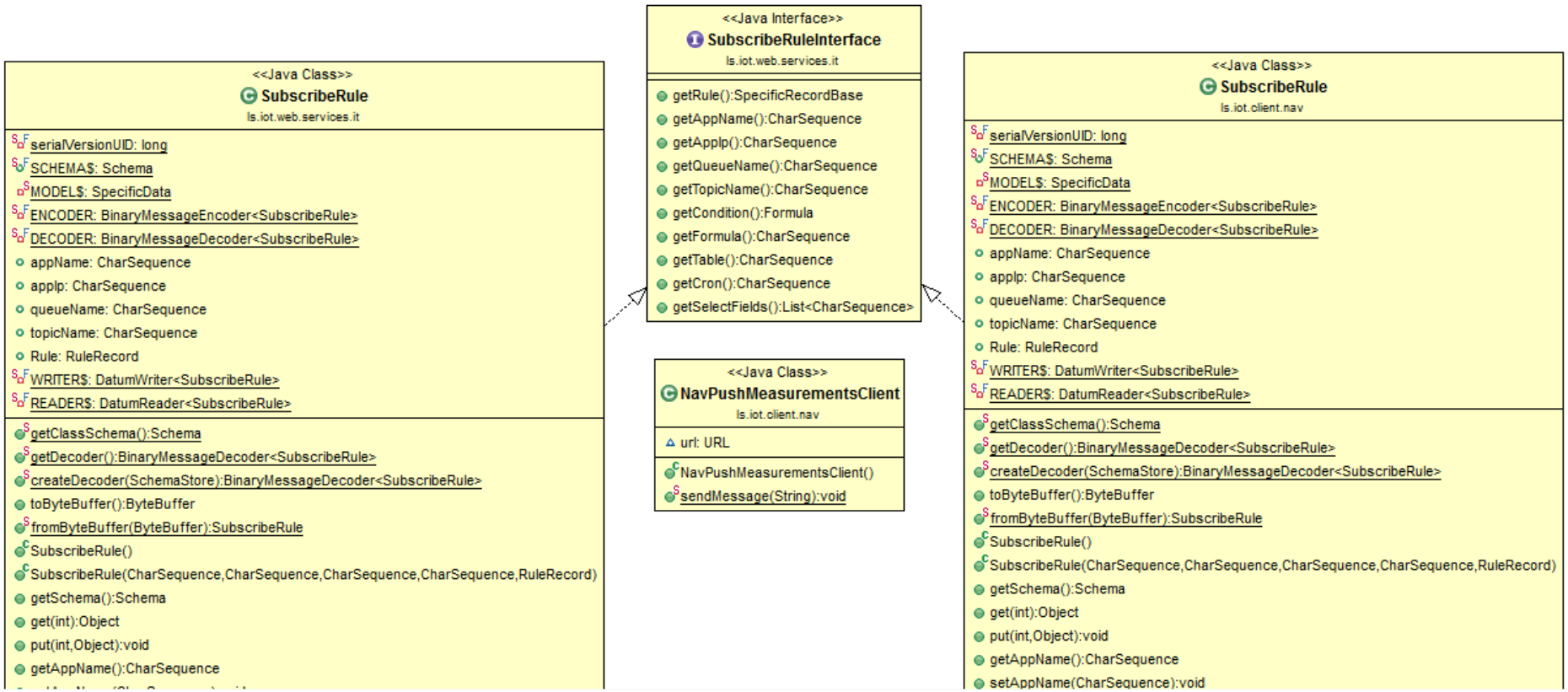
Gestione delle regole nel database



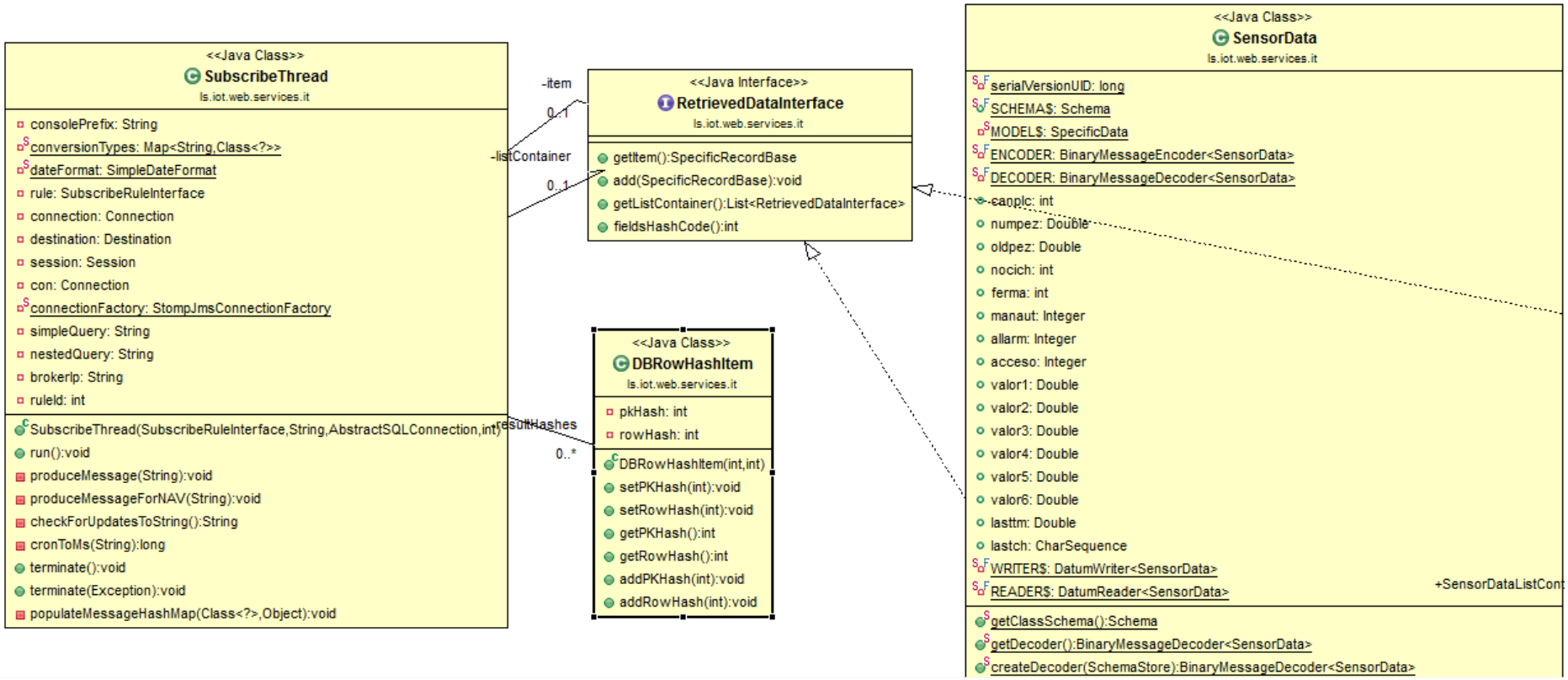
Schema JSON Subscribe Rule di NAV



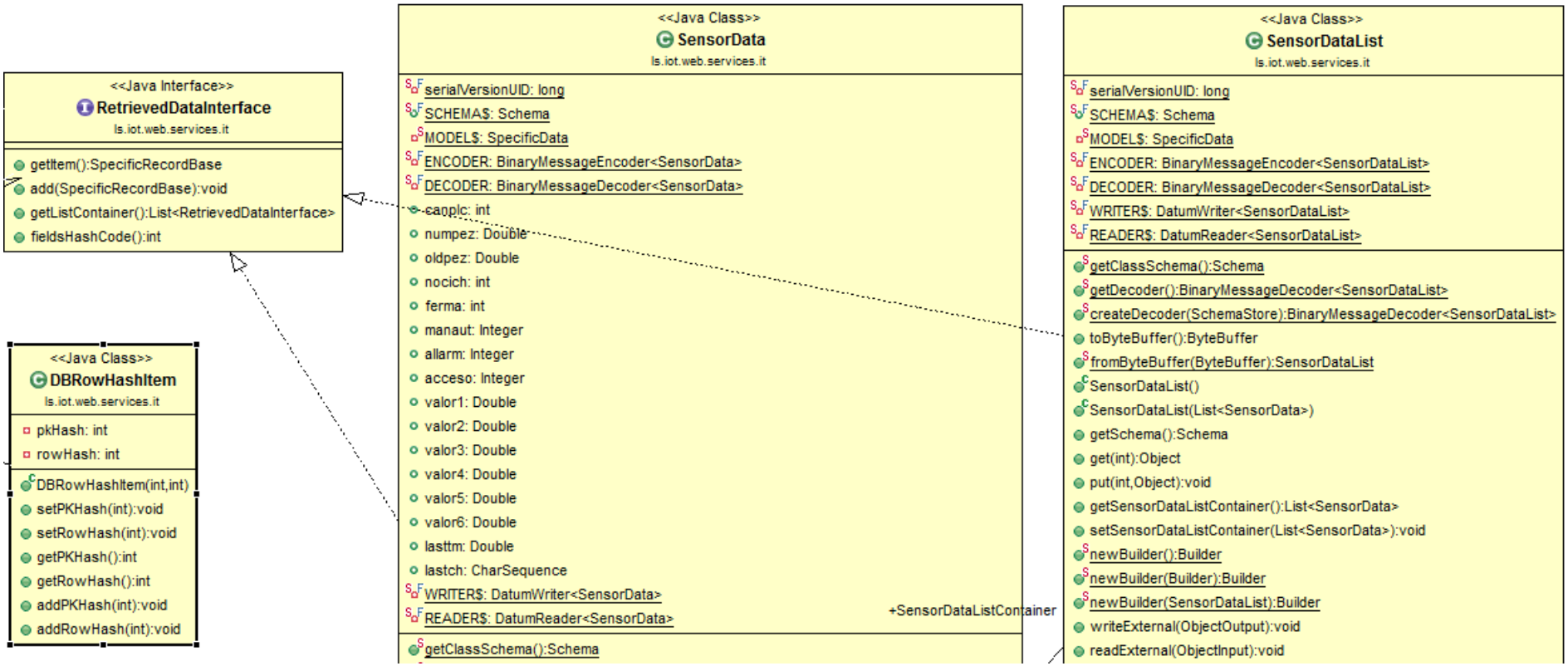
Class Diagram SubscribeRuleInterface



Class Diagram RetrievedDataInterface 1



Class Diagram RetrievedDataInterface 2



Schema Avro SensorData


```
{
  "namespace":"ls.iot.web.services.it",
  "name":"SensorDataList",
  "type":"record",
  "fields":[
    {
      "name":"SensorDataListContainer",
      "type":{
        "name":"SensorDataListContainer",
        "type":"array",
        "namespace":"ls.iot.web.services.it",
        "items":{
          "type":"record",
          "name":"SensorData",
          "namespace":"ls.iot.web.services.it",
          "fields":[
            {
              "name":"canplc",
              "type":"int",
              "doc":"ID del canale PLC."
            },
            {
              "name":"canplcannotation",
              "type":[
                "null",
                "SensorDataAnnotation"
              ],
              "doc":"Annotazione del campo."
            },
            {
              "name":"numpez",
              "type":[
                "null",
                "double"
              ],
              "doc":"Contatore principale del macchinario."
            },
            {
              "name":"numpezannotation",
              "type":[
                "null",
                "SensorDataAnnotation"
              ],
              "doc":"Annotazione del campo."
            },
            {
              "name":"oldpez",
              "type":[
                "null",
                "double"
              ],
              "doc":"Valore precedente del Contatore principale del macchinario."
            },
            {
              "name":"oldpezannotation",
              "type":[
                "null",
                "SensorDataAnnotation"
              ],
              "doc":"Annotazione del campo."
            },
            {
```

Schema Avro SensorDataAnnotation


```
{
   "type":"record",
   "name":"SensorDataAnnotation",
   "namespace":"ls.iot.web.services.it",
   "fields":[
      {
         "name":"prefname_meas",
         "type":[
            "null",
            "string"
         ],
         "doc":"Nome ontologia misurazione."
      },
      {
         "name":"prefprefix_name",
         "type":[
            "null",
            "string"
         ],
         "doc":"Prefisso ontologia misurazione."
      }
```

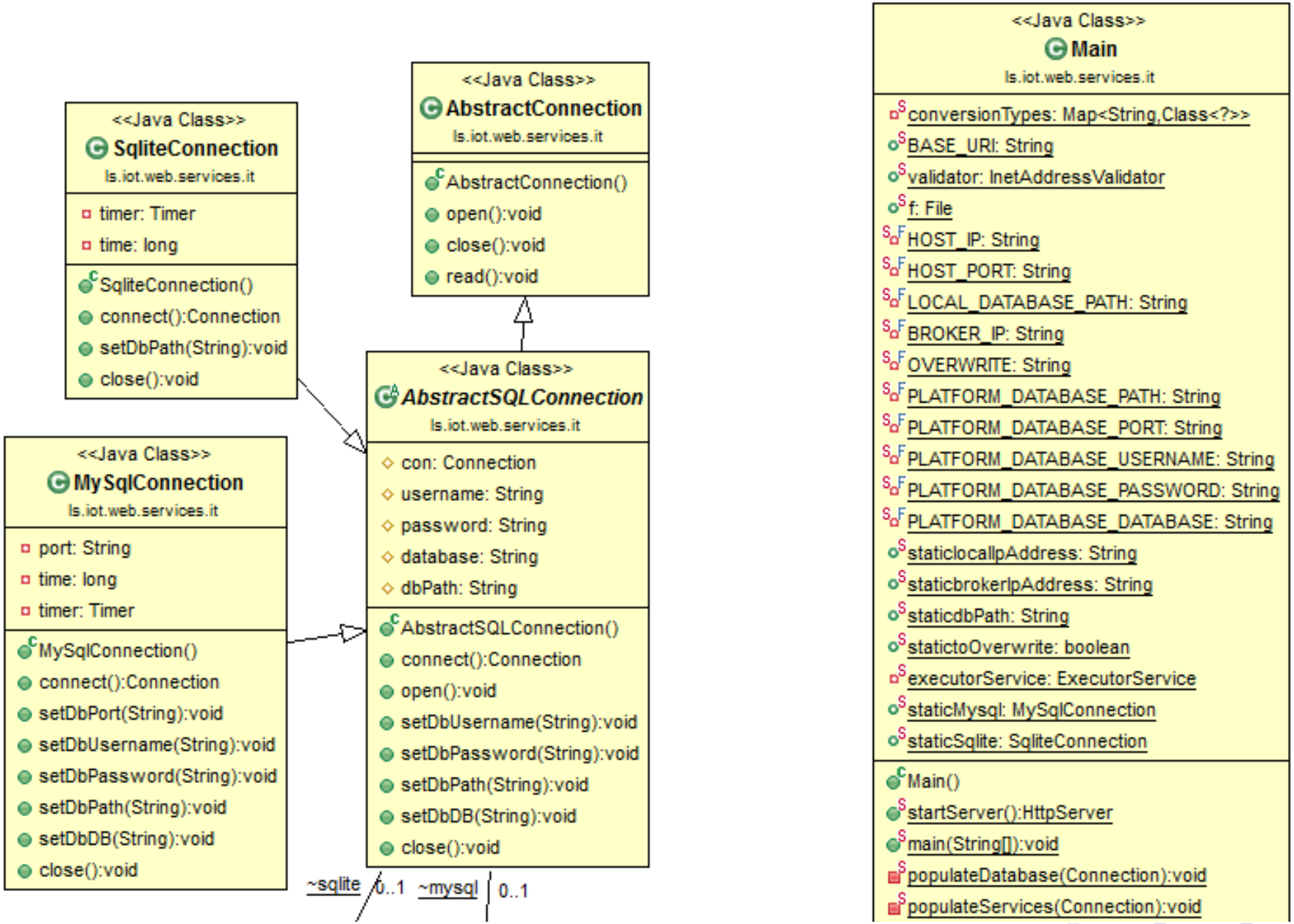
Estratto di getDataFromDbToSensorList

```
while(i <= rs.getMetaData().getColumnCount()) {

    if(rs.getMetaData().getColumnName(i).equals("idvalue")) {
        sensorFieldsEnd = true;
        currentIdValue = rs.getString(rs.getMetaData().getColumnName(i));
        if(currentIdValue==null) {
            currentIdValue="";
            break;
        }

        if(!currentIdValue.isEmpty()) {
            annot.put(rs.getMetaData().getColumnName(i+1), (rs.getObject(rs.getMetaData().getColumnName(i+1), conversionTypes) instanceof Date ?
                        dateFormat.format(((Date)rs.getObject(rs.getMetaData().getColumnName(i+1), conversionTypes)))
                    : rs.getObject(rs.getMetaData().getColumnName(i+1), conversionTypes)
                    ));
            annot.put(rs.getMetaData().getColumnName(i+2), (rs.getObject(rs.getMetaData().getColumnName(i+2), conversionTypes) instanceof Date ?
                    dateFormat.format(((Date)rs.getObject(rs.getMetaData().getColumnName(i+2), conversionTypes)))
                : rs.getObject(rs.getMetaData().getColumnName(i+2), conversionTypes)
                    ));
            if(prevItem != null && prevItem.fieldsHashCode() == item.fieldsHashCode() && sensorFieldsEnd) {
                item = prevItem;
            }
            item.getItem().put(currentIdValue+"annotation",annot);
        }
    } else if (!sensorFieldsEnd) {
        item.getItem().put(rs.getMetaData().getColumnName(i),
                (rs.getObject(rs.getMetaData().getColumnName(i), conversionTypes) instanceof Timestamp ?
                        ((Timestamp)rs.getObject(rs.getMetaData().getColumnName(i))).toString().split("\\.")[0]
                    : rs.getObject(rs.getMetaData().getColumnName(i), conversionTypes) instanceof Date ?
                            dateFormat.format((Date)rs.getObject(rs.getMetaData().getColumnName(i)))
                    : rs.getObject(rs.getMetaData().getColumnName(i), conversionTypes)
        ));
    }
```

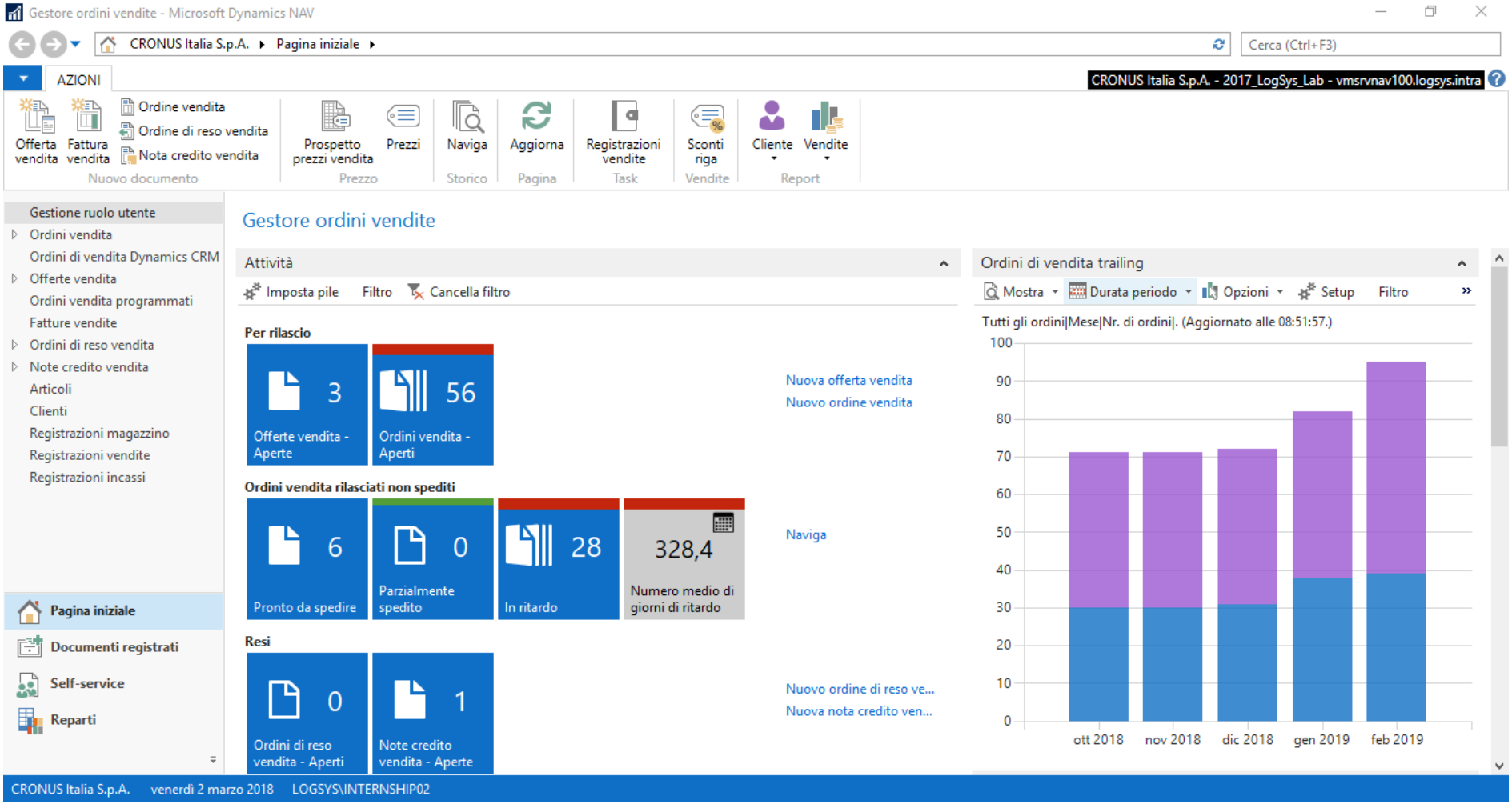
Class Diagram AbstractConnection



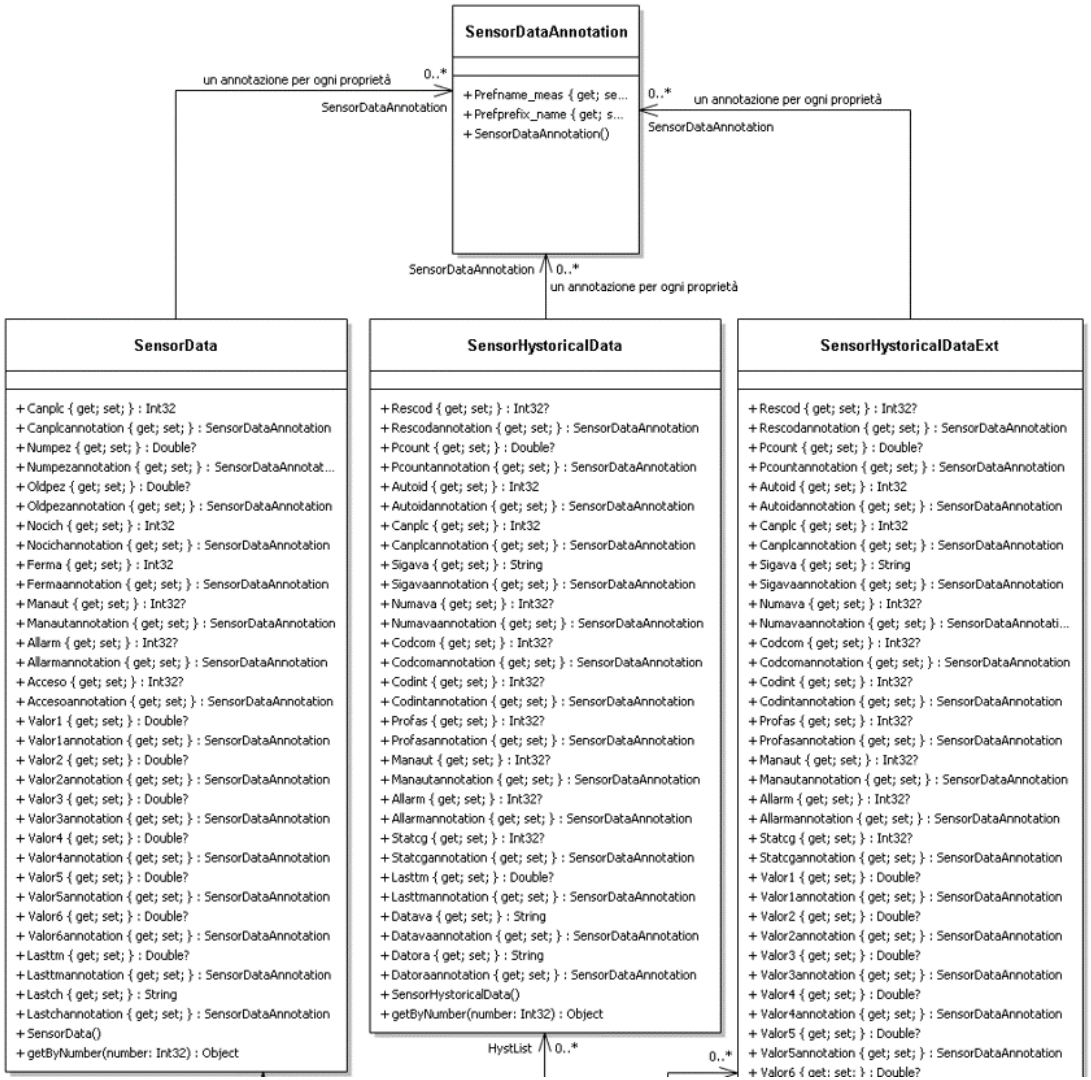
Client per la piattaforma

- Sviluppato su Microsoft Dynamics NAV nonostante diverse lacune dell'ambiente
  - Mancata possibilità di consumo diretto di servizi REST
  - Mancata possibilità di gestione del formato JSON
  - Difficoltà nell'interazione con software esterni non Microsoft
- Risoluzione tramite sviluppo di un client C#
  - con chiamata dei servizi REST, serializzazione e deserializzazione del JSON
  - in conformità con le classi della piattaforma tramite Apache Avro
  - integrato poi in NAV tramite dll
- Sviluppo di un "setup" per impostare le chiamate ai servizi su NAV
  - svolto mediante 2 approcci (PLC e Machine Center)
  - con trattamento dei dati per l'ambiente Navision
  - evitando di prendere valori già inseriti o errati
- Interazione con il servizio di sottoscrizione nell'ambiente NAV
  - tramite esternazione di una codeunit come web service SOAP

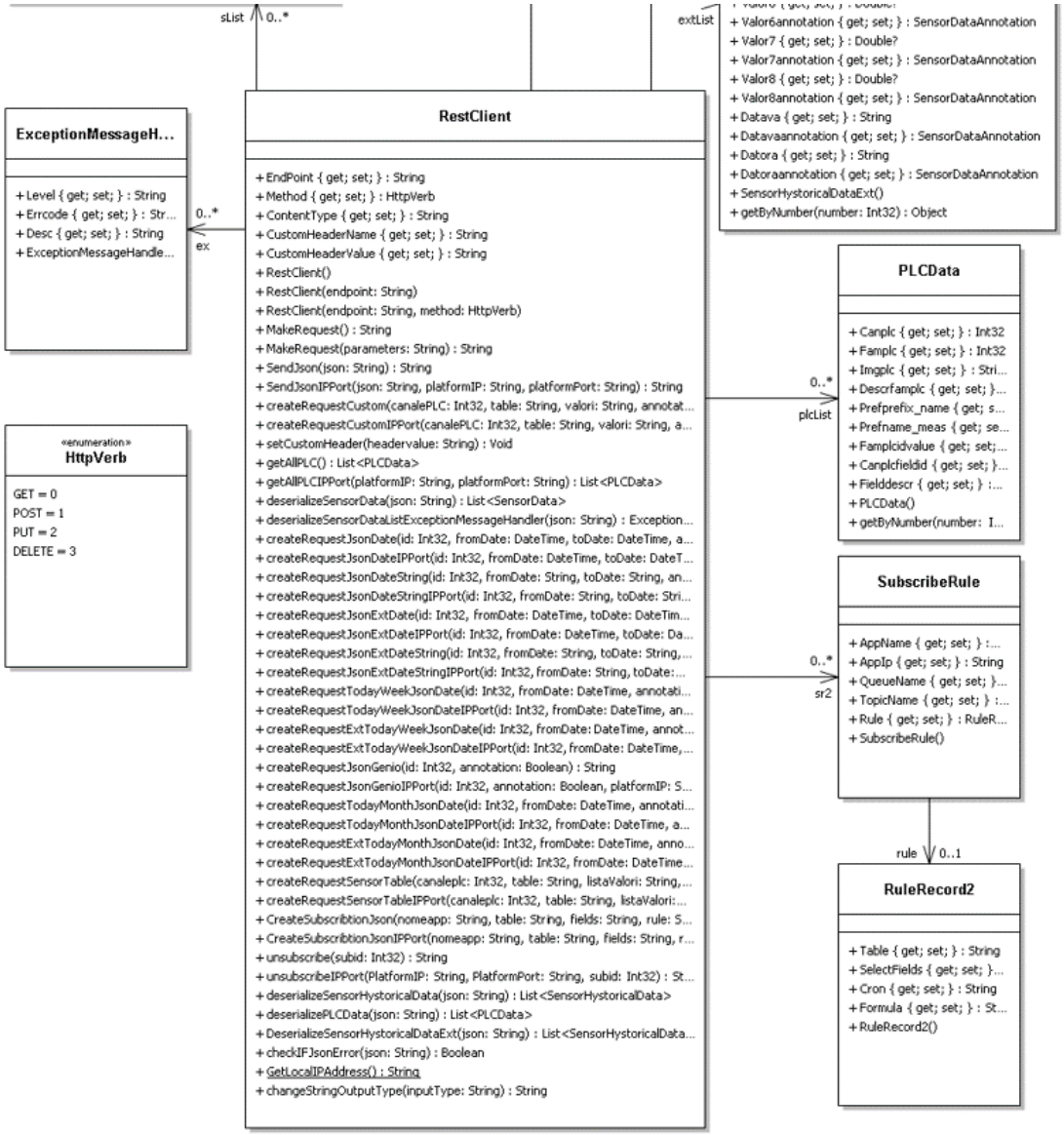
Client NAV


Gestore ordini vendite - Microsoft Dynamics NAV
CRONUS Italia S.p.A. ▸ Pagina iniziale ▸
Cerca (Ctrl+F3)
AZIONI
CRONUS Italia S.p.A. - 2017_LogSys_Lab - vmsrvnav100.logsys.intra
Offerta vendita
Fattura vendita
Ordine vendita
Ordine di reso vendita
Nota credito vendita
Nuovo documento
Prospetto prezzi vendita
Prezzi
Prezzo
Naviga
Storico
Aggiorna
Pagina
Registrazioni vendite
Task
Sconti riga
Vendite
Cliente
Vendite
Report
Gestione ruolo utente
Ordini vendita
Ordini di vendita Dynamics CRM
Offerte vendita
Ordini vendita programmati
Fatture vendite
Ordini di reso vendita
Note credito vendita
Articoli
Clienti
Registrazioni magazzino
Registrazioni vendite
Registrazioni incassi
Pagina iniziale
Documenti registrati
Self-service
Reparti
Gestore ordini vendite
Attività
Imposta pile
Filtro
Cancella filtro
Per rilascio
3
Offerte vendita - Aperte
56
Ordini vendita - Aperti
Nuova offerta vendita
Nuovo ordine vendita
Ordini vendita rilasciati non spediti
6
Pronto da spedire
0
Parzialmente spedito
28
In ritardo
328,4
Numero medio di giorni di ritardo
Naviga
Resi
0
Ordini di reso vendita - Aperti
1
Note credito vendita - Aperte
Nuovo ordine di reso ve...
Nuova nota credito ven...
Ordini di vendita trailing
Mostra
Durata periodo
Opzioni
Setup
Filtro
Tutti gli ordini|Mese|Nr. di ordini|. (Aggiornato alle 08:51:57.)
100
90
80
70
60
50
40
30
20
10
0
ott 2018
nov 2018
dic 2018
gen 2019
feb 2019
CRONUS Italia S.p.A.
venerdì 2 marzo 2018
LOGSYS\INTERNSHIP02

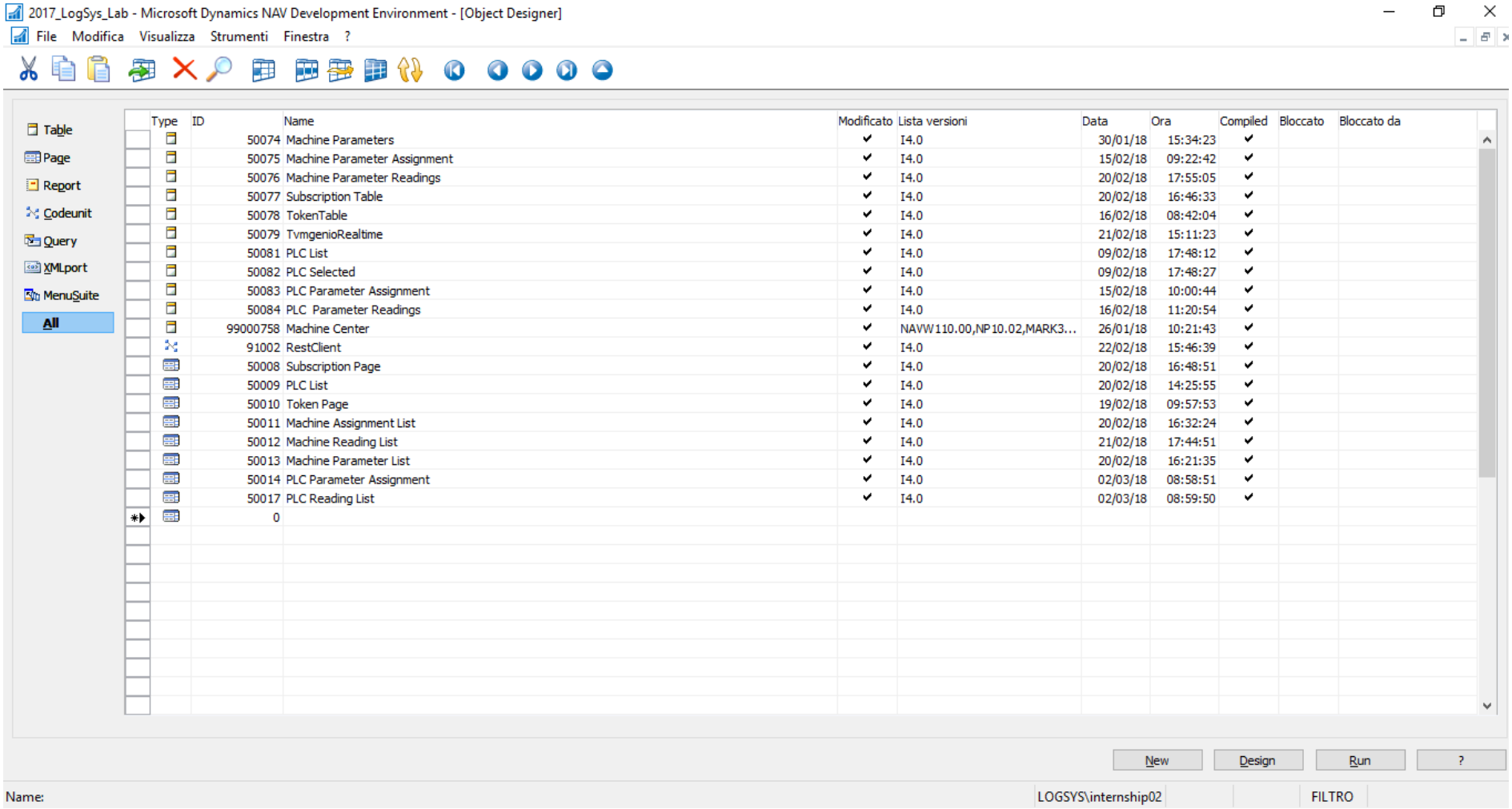
Class Diagram Client 1



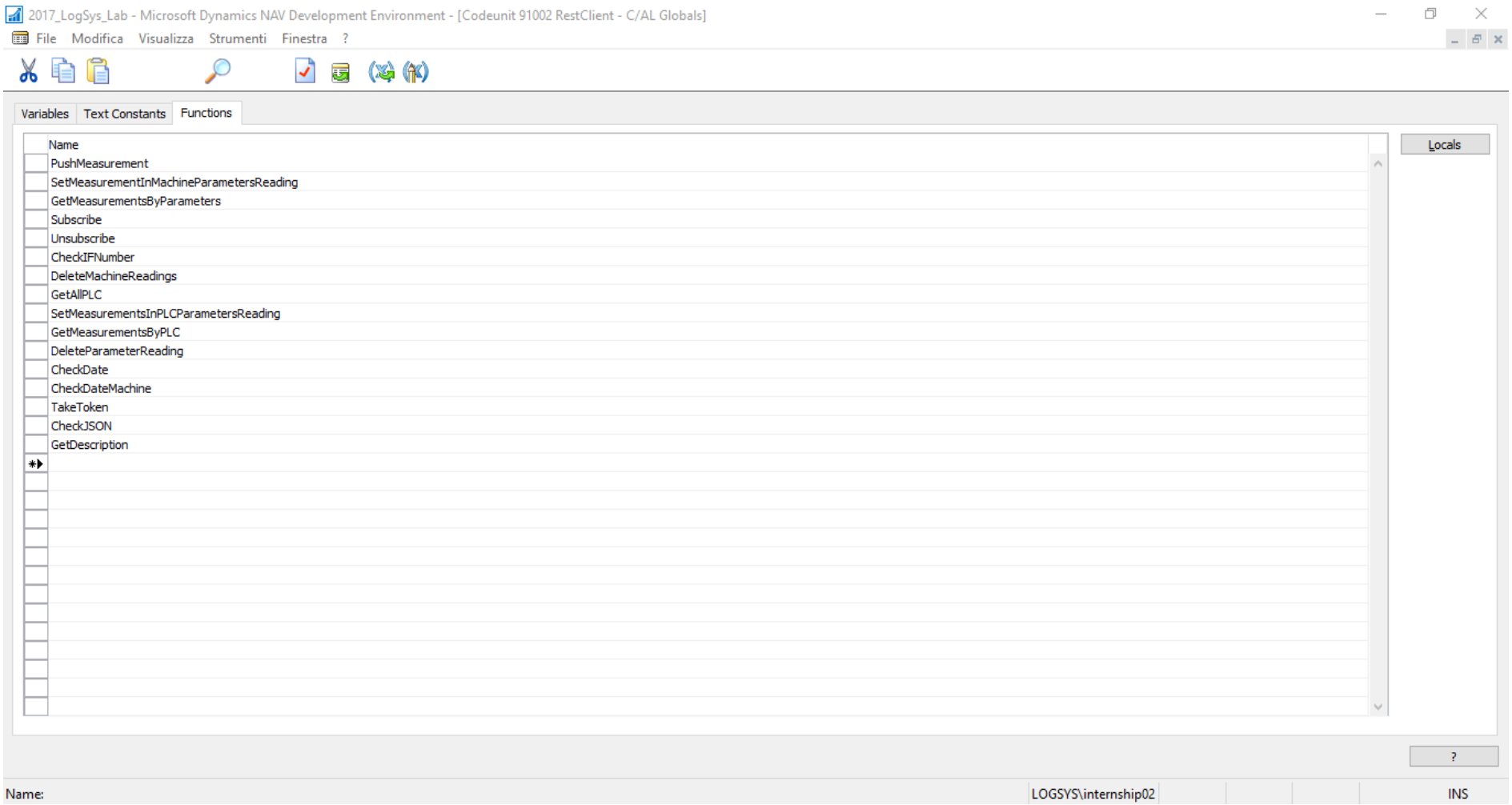
Class Diagram Client 2



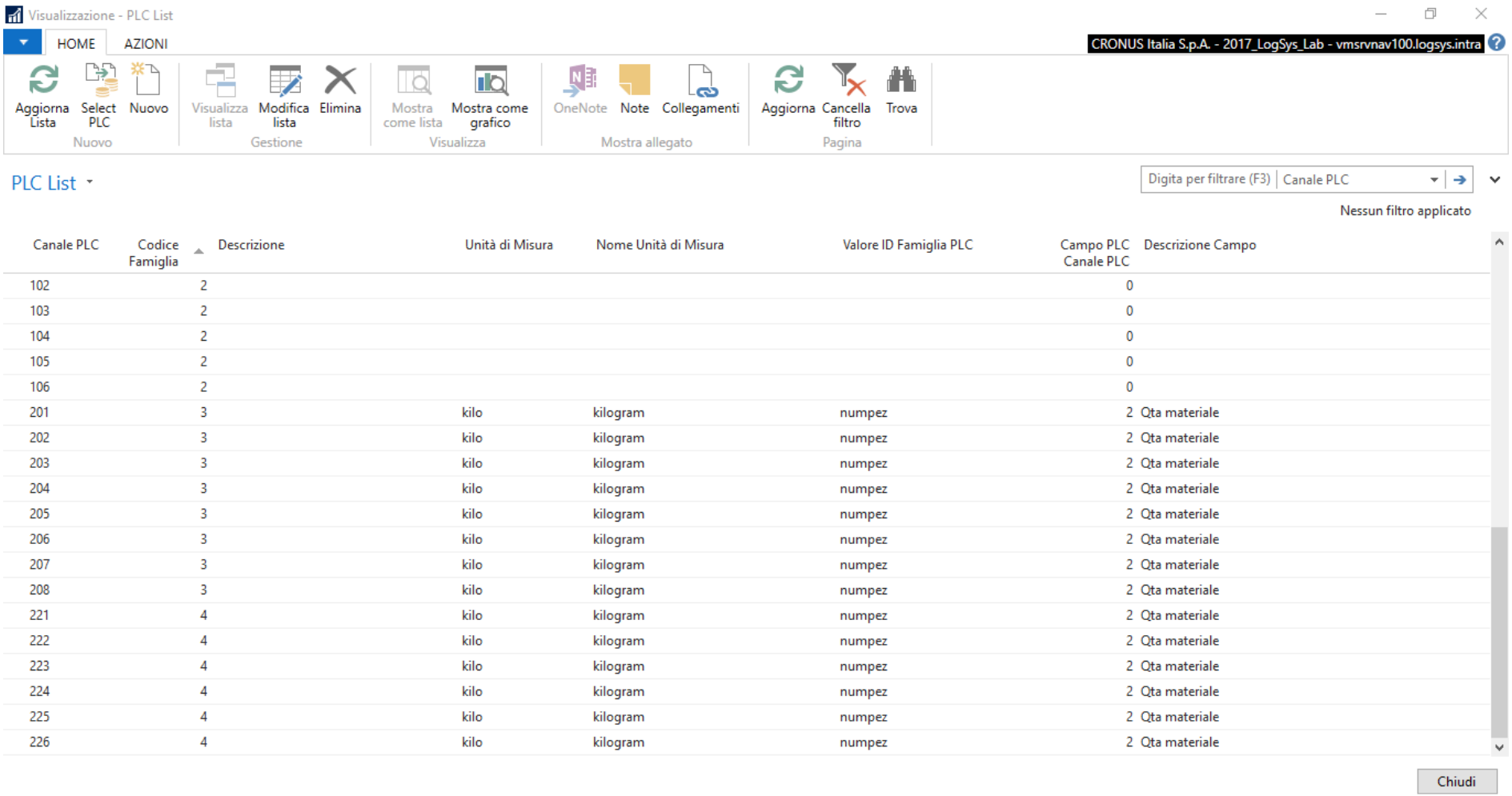
Ambiente di sviluppo (C/SIDE) NAV



Lista delle funzioni della codeunit

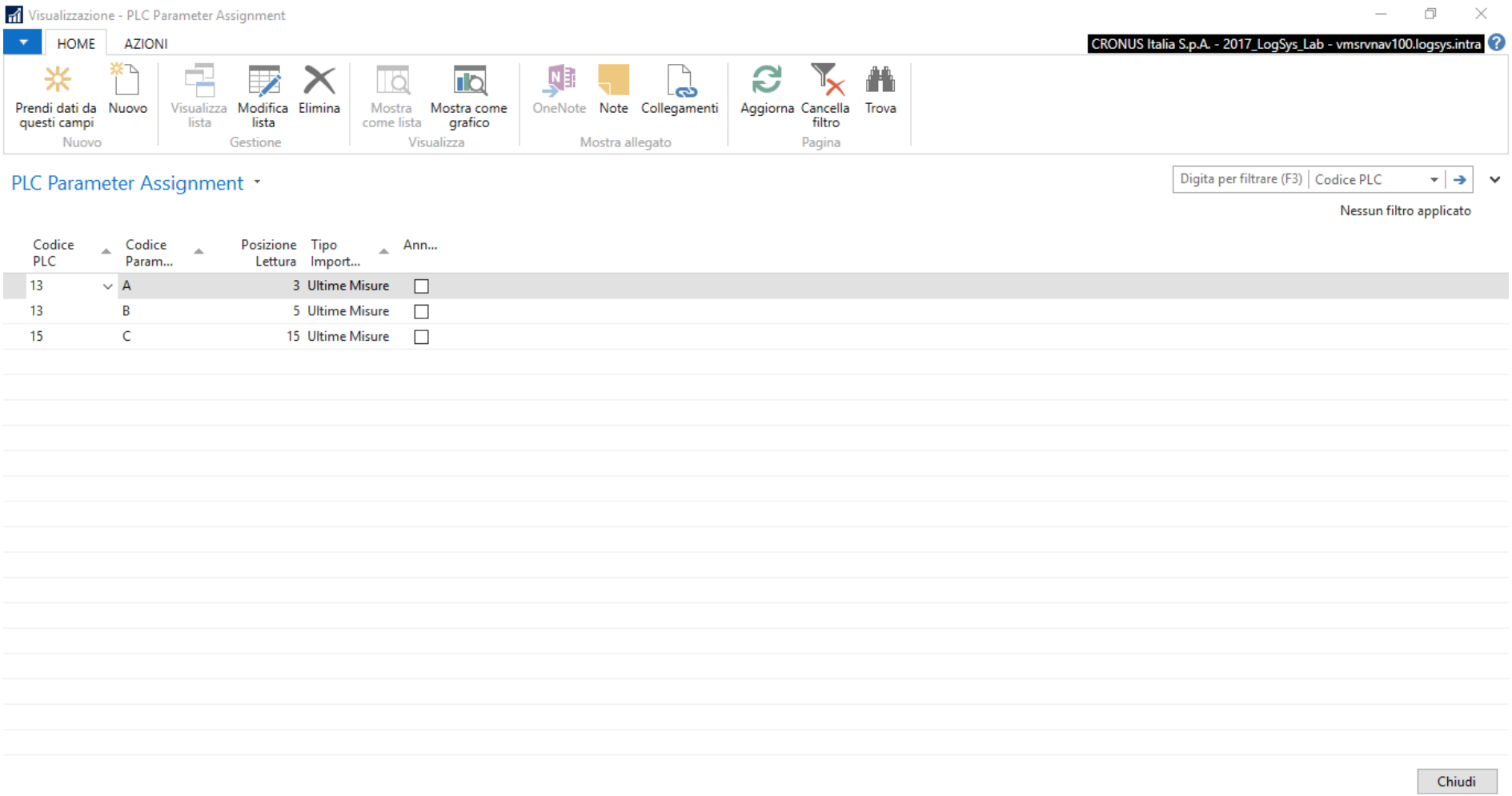


Lista PLC

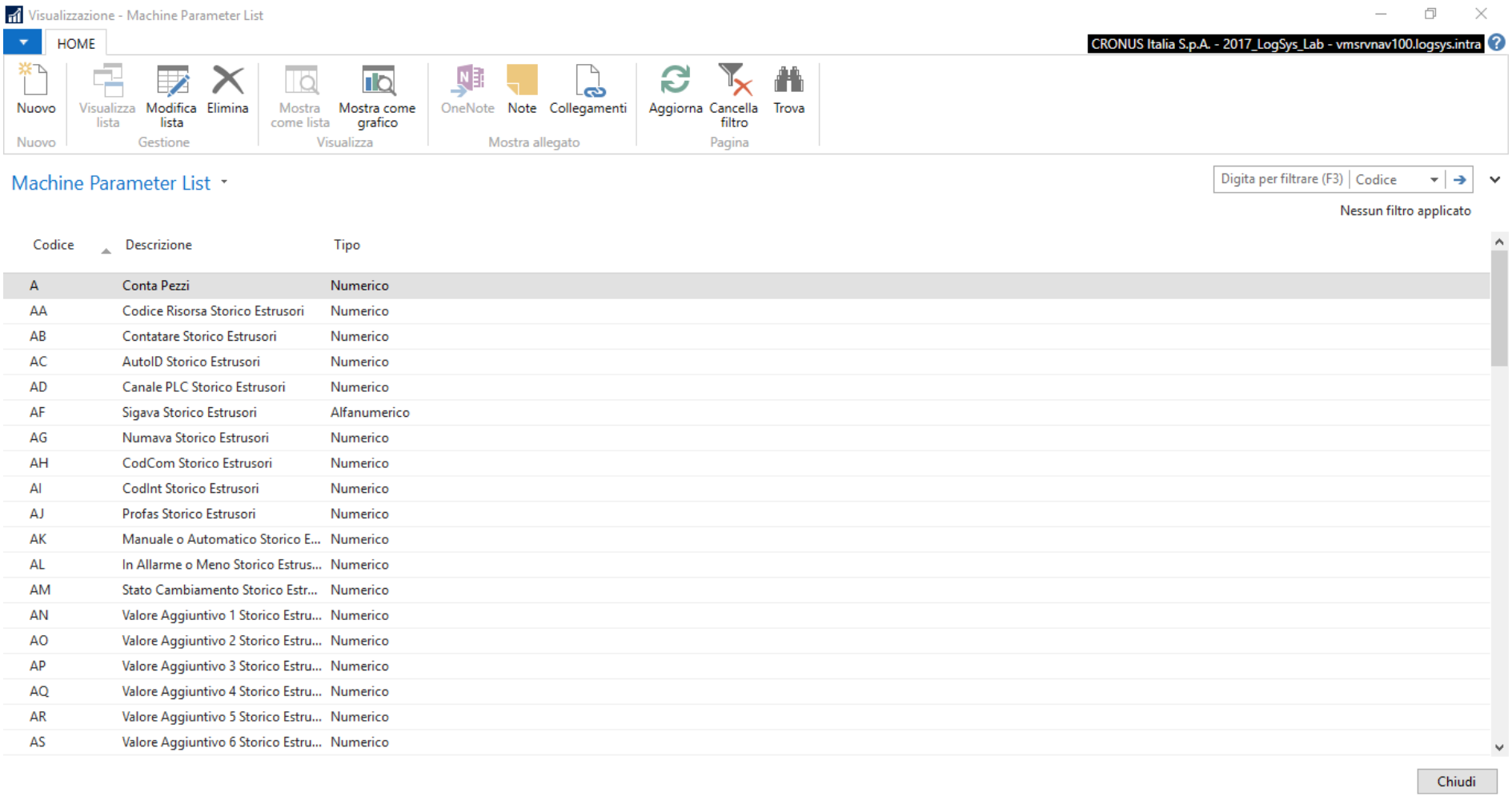


Canale PLC	Codice Famiglia	Descrizione	Unità di Misura	Nome Unità di Misura	Valore ID Famiglia PLC	Campo PLC Canale PLC	Descrizione Campo
102	2					0	
103	2					0	
104	2					0	
105	2					0	
106	2					0	
201	3		kilo	kilogram	numpez	2	Qta materiale
202	3		kilo	kilogram	numpez	2	Qta materiale
203	3		kilo	kilogram	numpez	2	Qta materiale
204	3		kilo	kilogram	numpez	2	Qta materiale
205	3		kilo	kilogram	numpez	2	Qta materiale
206	3		kilo	kilogram	numpez	2	Qta materiale
207	3		kilo	kilogram	numpez	2	Qta materiale
208	3		kilo	kilogram	numpez	2	Qta materiale
221	4		kilo	kilogram	numpez	2	Qta materiale
222	4		kilo	kilogram	numpez	2	Qta materiale
223	4		kilo	kilogram	numpez	2	Qta materiale
224	4		kilo	kilogram	numpez	2	Qta materiale
225	4		kilo	kilogram	numpez	2	Qta materiale
226	4		kilo	kilogram	numpez	2	Qta materiale

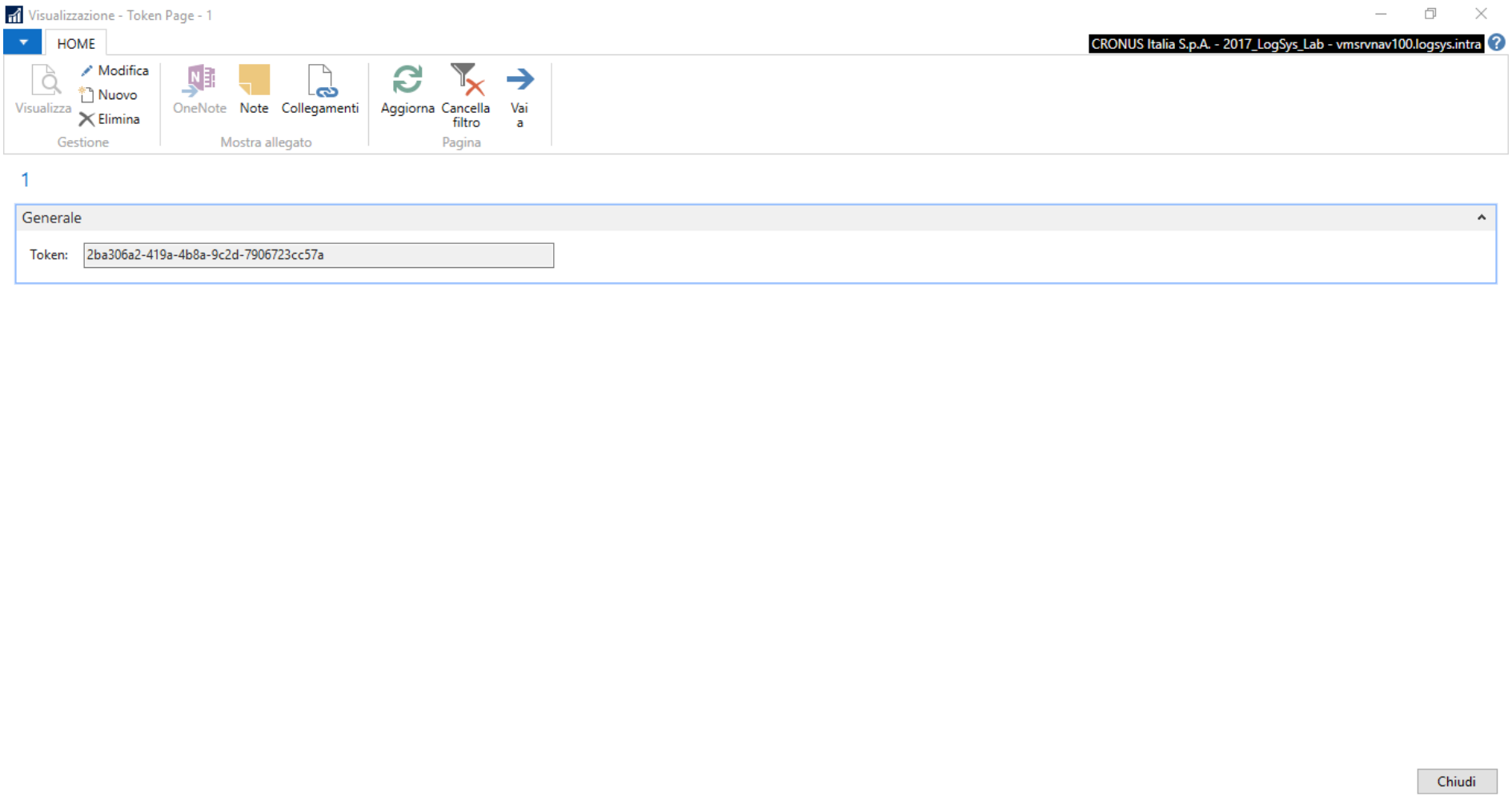
PLC Assignment List



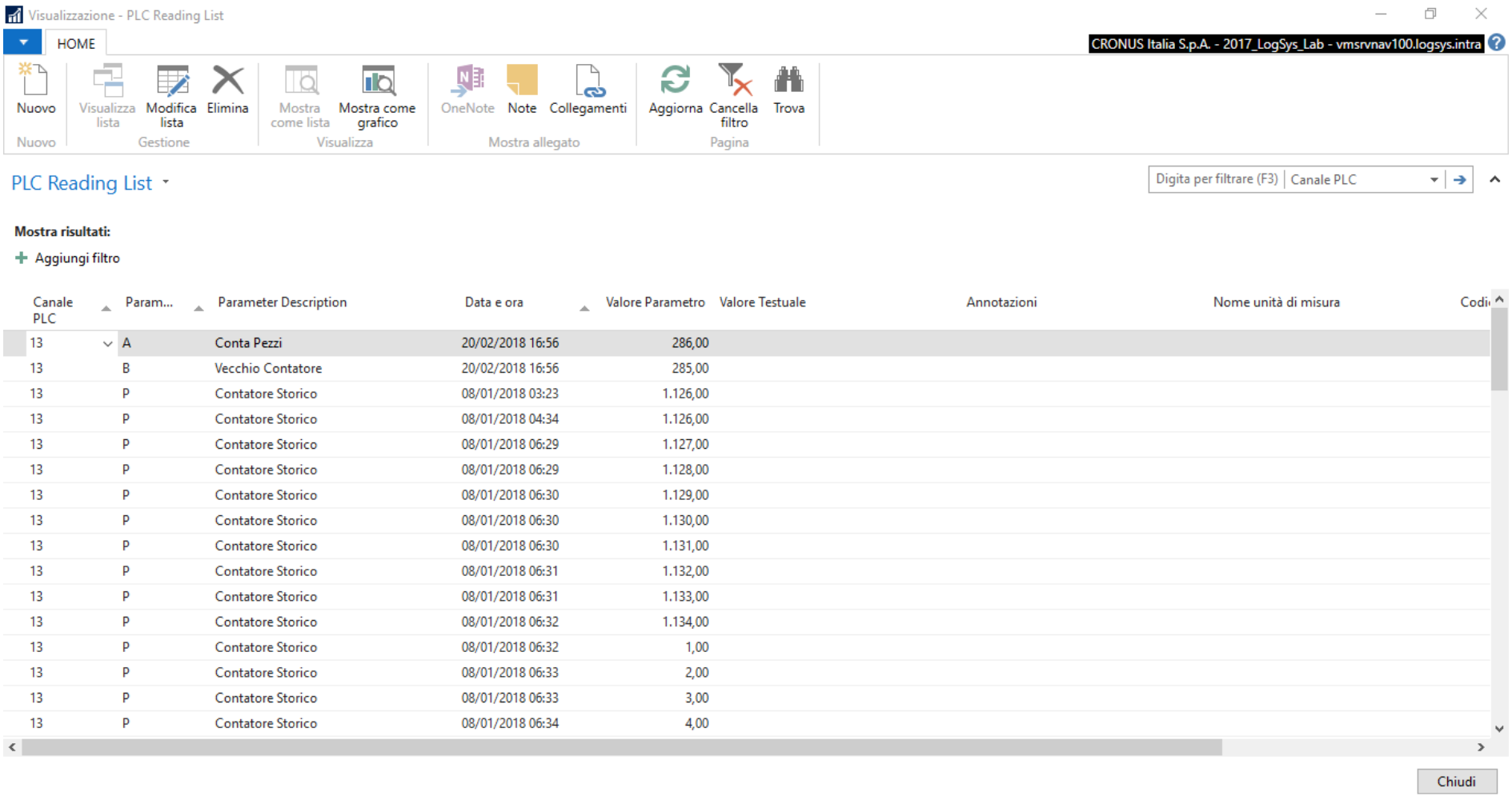
Lista con i parametri



Pagina del token



PLC Reading List



Metodo SetMeasurementPLC

2017_LogSys_Lab - Microsoft Dynamics NAV Development Environment - [Codeunit 91002 RestClient - C/AL Editor]

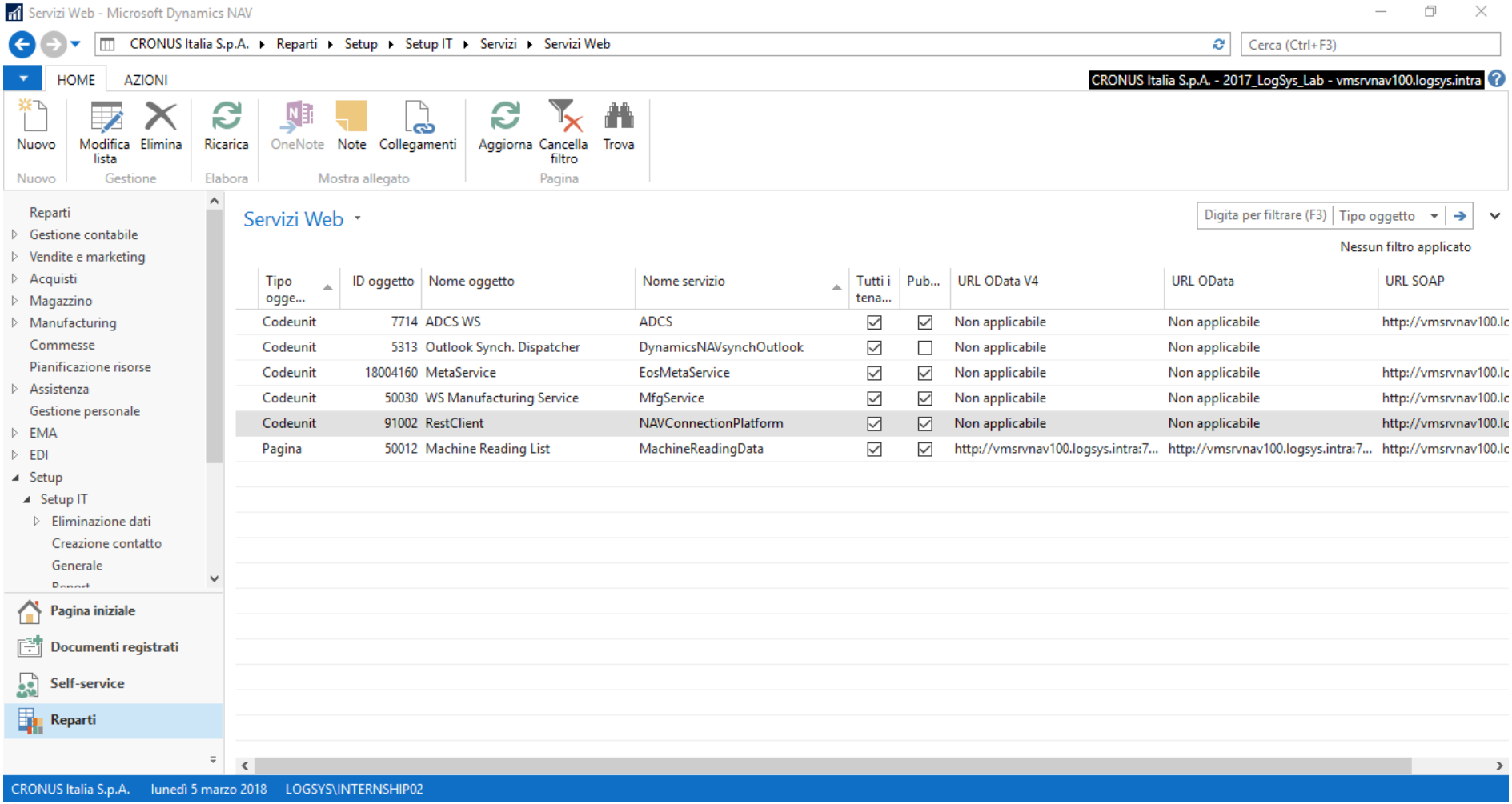
File Modifica Visualizza Strumenti Finestra ?

```
404 SetMeasurementsInPLCParametersReading()
405 CLEAR(lPLCParameterAssignment);
406 CLEAR(lPLCSelected);
407 lPLCSelected.SETFILTER("PLC Channel",lPLCParameterAssignment.CodPLC);
408 IF lPLCSelected.FINDSET THEN BEGIN
409   REPEAT
410     EVALUATE(lPLCChannel,lPLCSelected."PLC Channel");
411     lPLCParameterAssignment.SETFILTER(CodPLC,lPLCSelected."PLC Channel");
412     IF lPLCParameterAssignment.FINDSET THEN
413          GetMeasurementsByPLC(lPLCChannel, lPLCParameterAssignment)
414   UNTIL lPLCSelected.NEXT = 0;
415   END;
416
417 LOCAL GetMeasurementsByPLC(lCanPLC : Integer;lPLCParameterAssignment : Record "PLC Parameter Assignment")
418 CLEAR(lParametertoSend);
419 CLEAR(lParametertoSendAllData);
420 CLEAR(lParametertoSendExt);
421 CLEAR(lClient);
422 CLEAR(lMyList);
423 lAnnotation := FALSE;
424 lAnnotationHystorical := FALSE;
425 lAnnotationHystoricalExt := FALSE;
426 lSensorHystoricalData := lSensorHystoricalData.SensorHystoricalData();
427 lSensorHystoricalDataExt := lSensorHystoricalDataExt.SensorHystoricalDataExt();
428 IF lPLCParameterAssignment.FINDSET THEN BEGIN
429   REPEAT
430     IF (lPLCParameterAssignment.CodPLC = FORMAT(lCanPLC)) THEN BEGIN
431     IF lPLCParameterAssignment."Import Type" = 0 THEN BEGIN
432       lParametertoSend+=FORMAT(lPLCParameterAssignment."Reading Position");
433       lParametertoSend+=',';
```

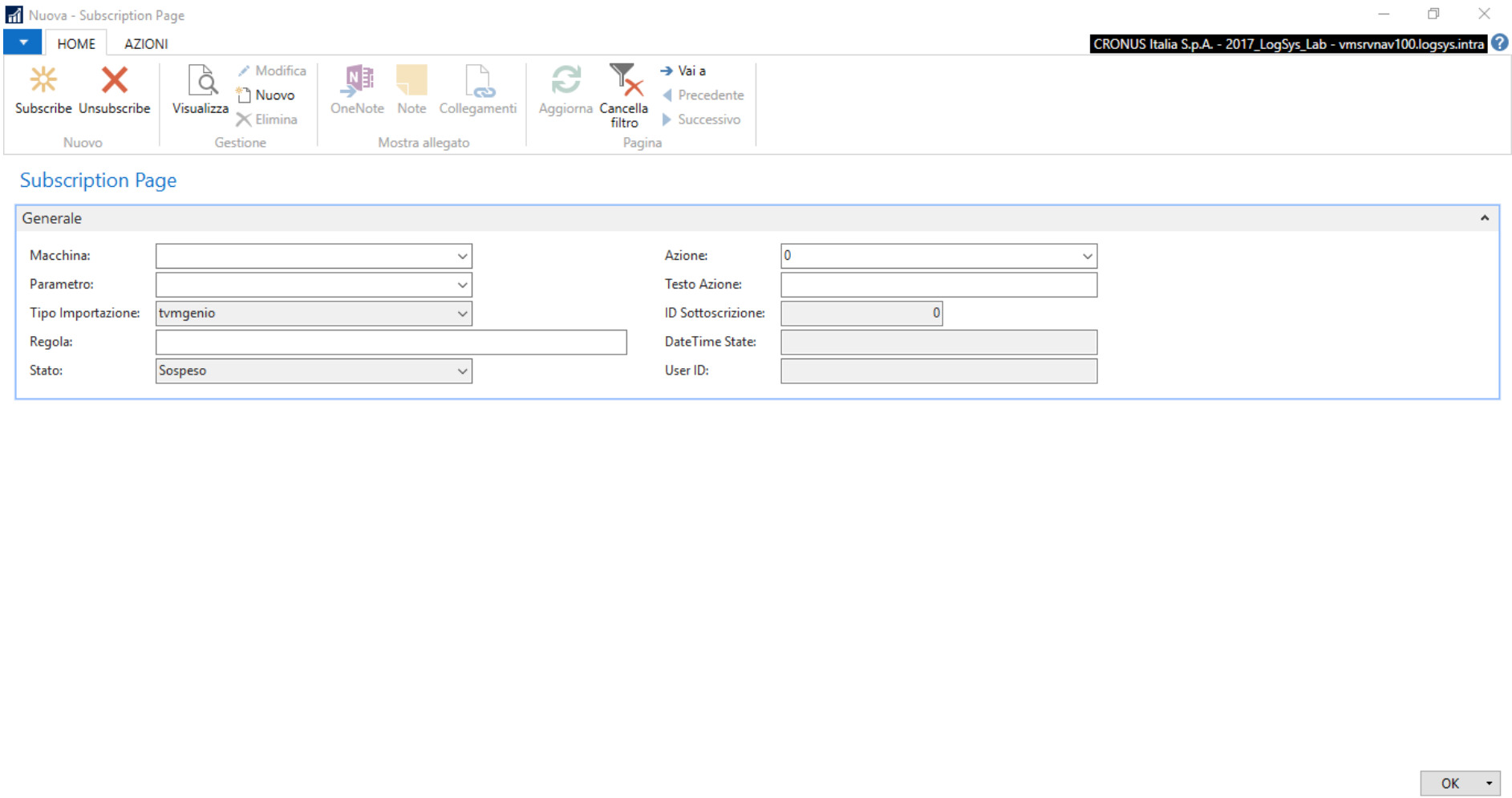
100 %

GetMeasurementsByPLC: Rg 13 Col 69 LOGSYS\internship02

NAV servizi web



NAV SubscriptionPage



Metodo PushMeasurement 1

Metodo PushMeasurement 2

2017_LogSys_Lab - Microsoft Dynamics NAV Development Environment - [Codeunit 91002 RestClient - C/AL Editor]

File Modifica Visualizza Strumenti Finestra ?

```
47  lMachineParameterReadings.Machine := lSubscriptionTable.Machine;
48  lMachineParameterReadings.Parameter := lSubscriptionTable.Parameter;
49  lMachineParameterReadings."Date Time" := lDateTimeParse.Parse(lSensorData.Lastch);
50  lMachineParameterReadings."Parameter Value" := lSensorData.getByNumber(lSubscriptionTable."Reading Position");
51  lMachineParameterReadings.Type := 1;
52  lMachineParameterReadings."ID LSIoT" := lSubscriptionTable."Reading Position";
53  lMachineParameterReadings."Parameter Description" := GetDescription(lSubscriptionTable.Parameter);
54  lMachineParameterReadings.INSERT;
55  END;
56  END ELSE IF lSubscriptionTable."Import Type" = 1 THEN BEGIN
57  lMyList := lClient.deserializeSensorHystoricalData(Json);
58  FOR gI := 0 TO (lMyList.Count -1) DO
59  BEGIN
60  lSensorHystoricalData := lMyList.Item(gI);
61  lMachineParameterReadings.Machine := lSubscriptionTable.Machine;
62  lMachineParameterReadings.Parameter := lSubscriptionTable.Parameter;
63  lMachineParameterReadings."Date Time" := lDateTimeParse.Parse(lSensorHystoricalData.Datora);
64  lMachineParameterReadings."Parameter Value" := lSensorHystoricalData.getByNumber(lSubscriptionTable."Reading Position");
65  lMachineParameterReadings.Type := 1;
66  lMachineParameterReadings."ID LSIoT" := lSubscriptionTable."Reading Position";
67  lMachineParameterReadings."Hystorical AutoID" := lSensorHystoricalData.Autoid;
68  lMachineParameterReadings."Parameter Description" := GetDescription(lSubscriptionTable.Parameter);
69  lMachineParameterReadings.INSERT;
70  END;
71  END ELSE BEGIN
72  lMyList := lClient.deserializeSensorHystoricalDataExt(Json);
73  FOR gI := 0 TO (lMyList.Count -1) DO
74  BEGIN
75  lSensorHystoricalDataExt := lMyList.Item(gI);
76  lMachineParameterReadings.Machine := lSubscriptionTable.Machine;
```

100 %

PushMesuraments: Rg 29 Col 25 LOGSYS\internship02

Metodo PushMeasurement 3

2017_LogSys_Lab - Microsoft Dynamics NAV Development Environment - [Codeunit 91002 RestClient - C/AL Editor]

File Modifica Visualizza Strumenti Finestra ?

```
58    FOR gI := 0 TO (lMyList.Count -1) DO
59    BEGIN
60    lSensorHystoricalData := lMyList.Item(gI);
61    lMachineParameterReadings.Machine := lSubscriptionTable.Machine;
62    lMachineParameterReadings.Parameter := lSubscriptionTable.Parameter;
63    lMachineParameterReadings."Date Time" := lDateTimeParse.Parse(lSensorHystoricalData.Datora);
64    lMachineParameterReadings."Parameter Value" := lSensorHystoricalData.getByNumber(lSubscriptionTable."Reading Position");
65    lMachineParameterReadings.Type := 1;
66    lMachineParameterReadings."ID LSIoT" := lSubscriptionTable."Reading Position";
67    lMachineParameterReadings."Hystorical AutoID" := lSensorHystoricalData.Autoid;
68    lMachineParameterReadings."Parameter Description" := GetDescription(lSubscriptionTable.Parameter);
69    lMachineParameterReadings.INSERT;
70    END;
71    END ELSE BEGIN
72    lMyList := lClient.deserializeSensorHystoricalDataExt(Json);
73    FOR gI := 0 TO (lMyList.Count -1) DO
74    BEGIN
75    lSensorHystoricalDataExt := lMyList.Item(gI);
76    lMachineParameterReadings.Machine := lSubscriptionTable.Machine;
77    lMachineParameterReadings.Parameter := lSubscriptionTable.Parameter;
78    lMachineParameterReadings."Date Time" := lDateTimeParse.Parse(lSensorHystoricalDataExt.Datora);
79    lMachineParameterReadings."Parameter Value" := lSensorHystoricalDataExt.getByNumber(lSubscriptionTable."Reading Position");
80    lMachineParameterReadings.Type := 1;
81    lMachineParameterReadings."ID LSIoT" := lSubscriptionTable."Reading Position";
82    lMachineParameterReadings."Hystorical AutoID" := lSensorHystoricalDataExt.Autoid;
83    lMachineParameterReadings."Parameter Description" := GetDescription(lSubscriptionTable.Parameter);
84    lMachineParameterReadings.INSERT;
85    END;
86    END;
87  END;
```

100 %

PushMesuraments: Rg 29 Col 25 LOGSYS\internship02