**Università degli Studi di Camerino**

Scuola di Scienze e Tecnologie

Corso di Laurea in Informatica (Classe L-31)

# LS Genio Platform

Laureando
**Vincenzo Nucci**

**Matricola 092861**

Laureando
**Matteo Tiberi**

**Matricola 092913**

Relatore
**Dott. Rosario Culmone**

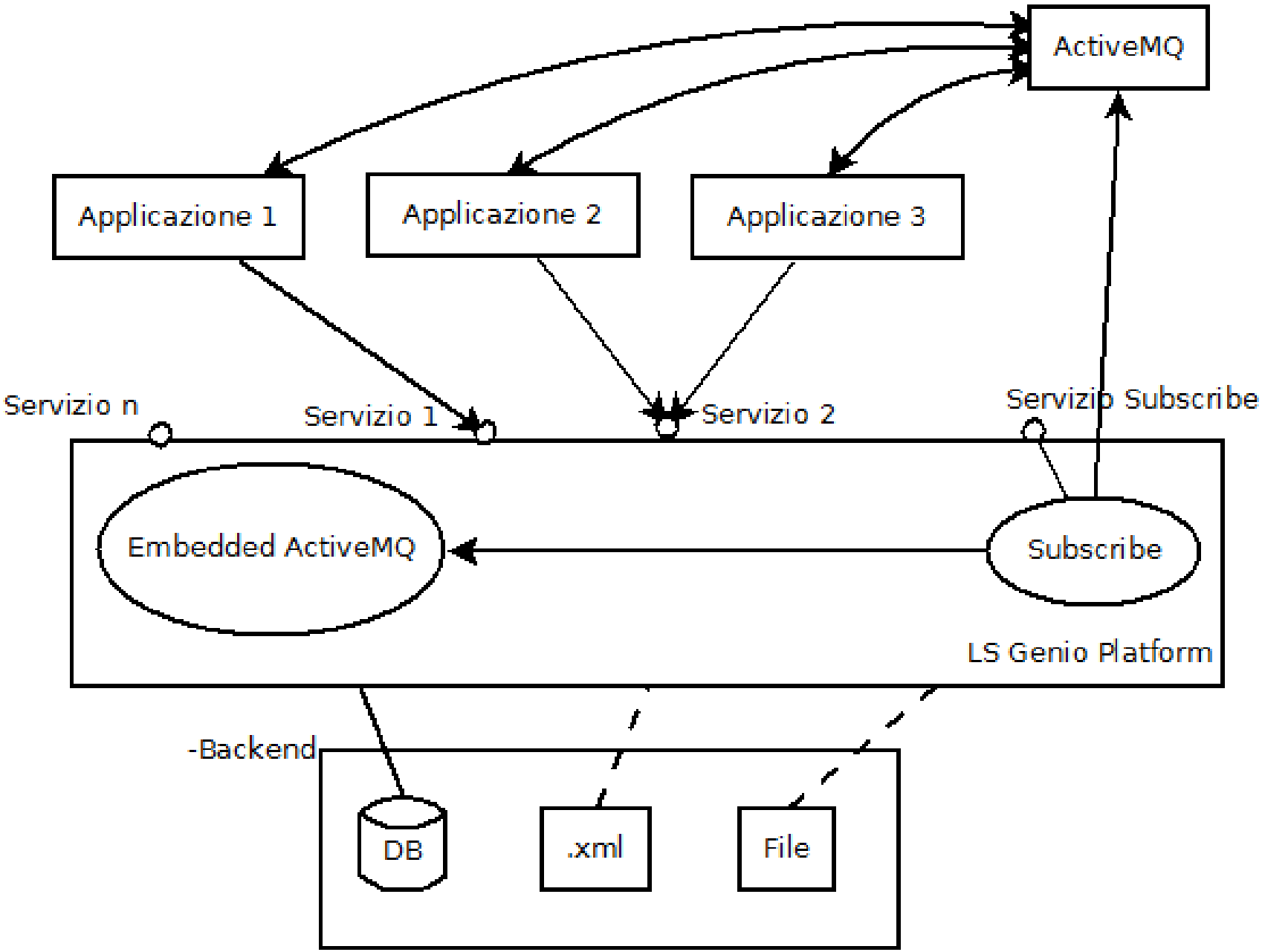
Correlatore
**Dott. Leonardo Vito**

A.A. 2016/2017

# Obiettivi

- Piattaforma REST indipendente da sorgenti dati
  - Autenticazione tramite token
  - Interfaccia web
- Servizio di sottoscrizione "subscribe"
  - Notifica dei messaggi PUSH
- Integrazione dei servizi con NAV
- Servizio di monitoraggio dei dati
  - Controllo valore oltre soglia

# Architettura piattaforma



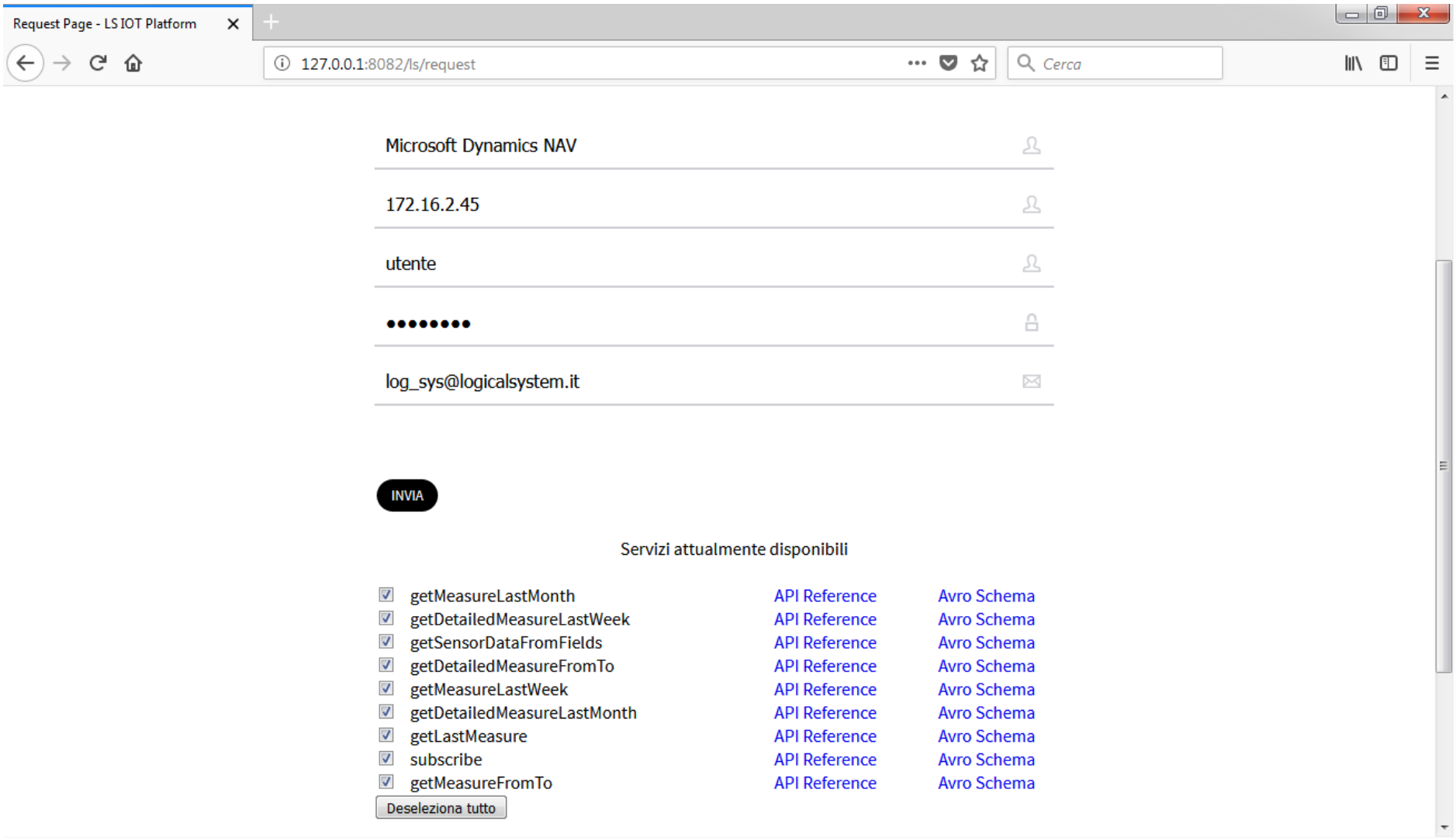
# Struttura tabelle backend

| tmvgenio |
| --- |
| *canplc smallint(6) default '0' |
| *numpez double |
| *oldpez double |
| *nocich int(11), default '0' |
| *ferma smallint(6), default'0' |
| *manaut smallint(6) |
| *allarm smallint(6) |
| *acceso smallint(6) |
| *valor1 double |
| *valor2 double |
| *valor3 double |
| *valor4 double |
| *valor5 double |
| *valor6 double |
| *lasttm double |
| *lastch datetime, default '0000-00-00 00:00:00' |

| tvatest |
| --- |
| *autoid int(11) auto_increment |
| *rescod int(11) |
| *pcount double |
| *canplc smallint(6) |
| *sigava char(4) |
| *numava int(11) |
| *codcom int(11) |
| *codint int(11) |
| *profas smallint(6) |
| *manaut smallint(6) |
| *allarm smallint(6) |
| *statcg smallint(6) |
| *lasttm double |
| *datava date |
| *datora datetime |

| tvavext |
| --- |
| *autoid int(11) auto_increment |
| *rescod int(11) |
| *pcount double |
| *canplc smallint(6) |
| *sigava char(4) |
| *numava int(11) |
| *codcom int(11) |
| *codint int(11) |
| *profas smallint(6) |
| *manaut smallint(6) |
| *allarm smallint(6) |
| *statcg smallint(6) |
| *valor1 double |
| *valor2 double |
| *valor3 double |
| *valor4 double |
| *valor5 double |
| *valor6 double |
| *valor7 double |
| *valor8 double |
| *datava date |
| *datora datetime |

# Pagina web per la richiesta di abilitazione



# Pagina web catalogo Smart Object

Smart Objects - LS IOT Platform

127.0.0.1:8082/ls/plc

Cerca

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 208 | 3 | | 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 |

# Esempio di un servizio - getlastmeasure

```
@Path("getlastmeasure/{sensorId}/{annotation}/{enhanced}")
@GET
@Produces(MediaType.APPLICATION_JSON)
public String getLastMeasure(@HeaderParam("token") String token, @PathParam("sensorId") int sensorId,
        @PathParam("annotation") boolean annotation, @PathParam("enhanced") boolean enhanced)
{
    try {
    if (!checkToken(token, "getLastMeasure"))
        throw new IllegalArgumentException("Token: "+token+" not valid/not authorized");
    Connection con;
    String query = "";
    logger.debug("GET ["+token+"] last measure of sensor: " + sensorId);

    con = ((AbstractSQLConnection)conf.getProperty("mysql")).connect();
        if(annotation)
            if(enhanced)
                query = "select tvmgenio.*,measann.idvalue, measann.descr, measann.type, measann.measurementuri, "
                        + "measann.observationprocedure, measann.onturi from tvplcset left join tvplcfam using(famplc) "
                        + "left join tvmgenio using (canplc) left join measann using (famplc) where (canplc="+sensorId+")";
            else
                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,enhanced,true);

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

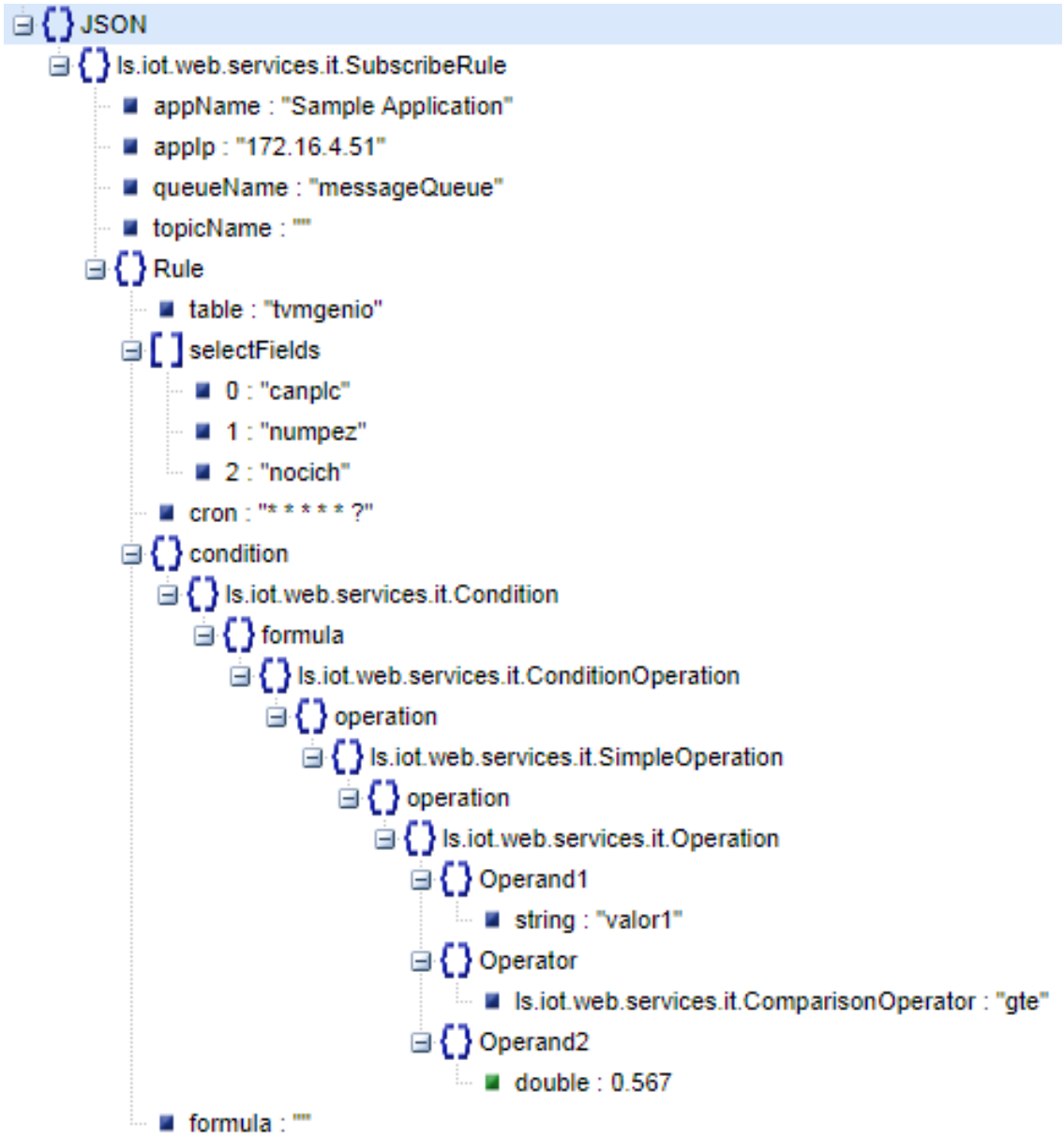
# Valori di ritorno di getlastmeasure

GET 127.0.0.1:8082/ls/iot/getlastmeasure/106/true/true

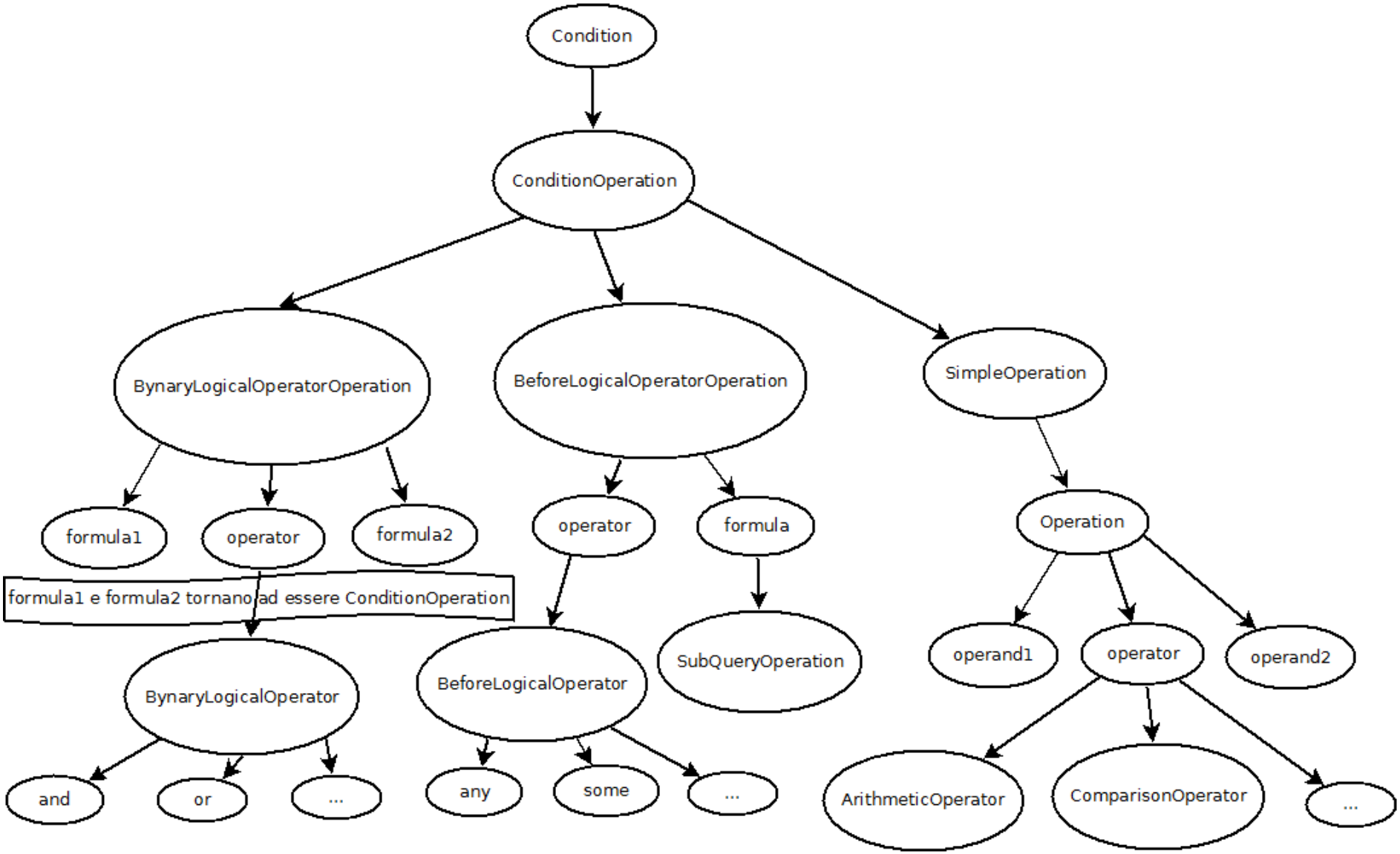
Pretty Raw Preview JSON


```
        "canplc": 106,
        "canplcannotation": null,
        "numpez": 1759,
        "numpezannotation": null,
        "oldpez": 1743,
        "oldpezannotation": null,
        "nocich": 0,
        "nocichannotation": null,
        "ferma": 0,
        "fermaannotation": null,
        "manaut": 0,
        "manautannotation": null,
        "allarm": 0,
        "allarmannotation": null,
        "acceso": 0,
        "accesoannotation": null,
        "valor1": 130,
        "valor1annotation": {
            "id": "valor1106",
            "description": "Extrusor Temperature",
            "type": "Measurement",
            "phenomenonTime": "2018-02-20 16:56:54",
            "observedProperty": "http://www.logicalsystem.it/it/onto/ExtrusorTemperature",
            "procedure": "http://www.logicalsystem.it/it/register/process/sensorGenio.xml",
            "featureOfInterest": "canplc 106",
            "resultTime": "2018-02-20 16:56:54",
            "result": {
                "value": "130.0",
                "uom": "http://purl.obolibrary.org/obo/UO_0000027"
```

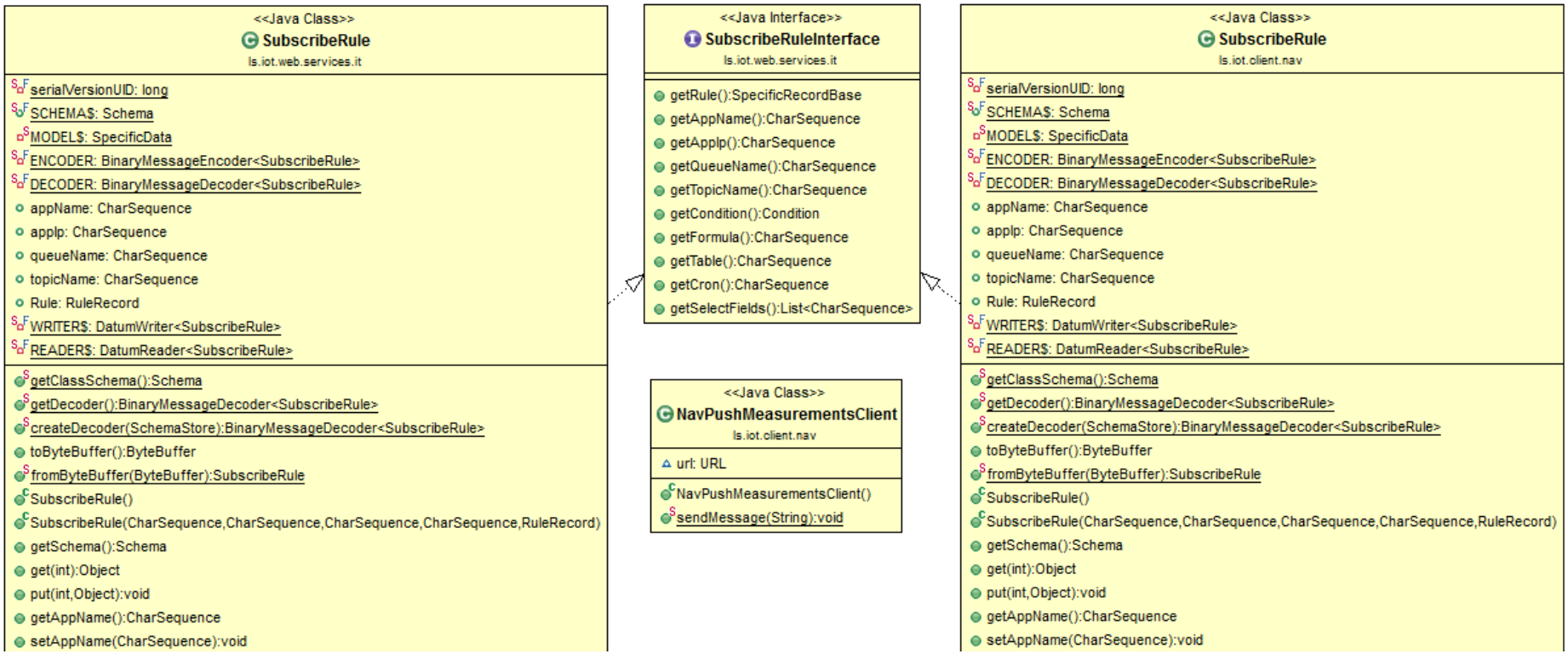
# Subscribe Rule di una applicazione



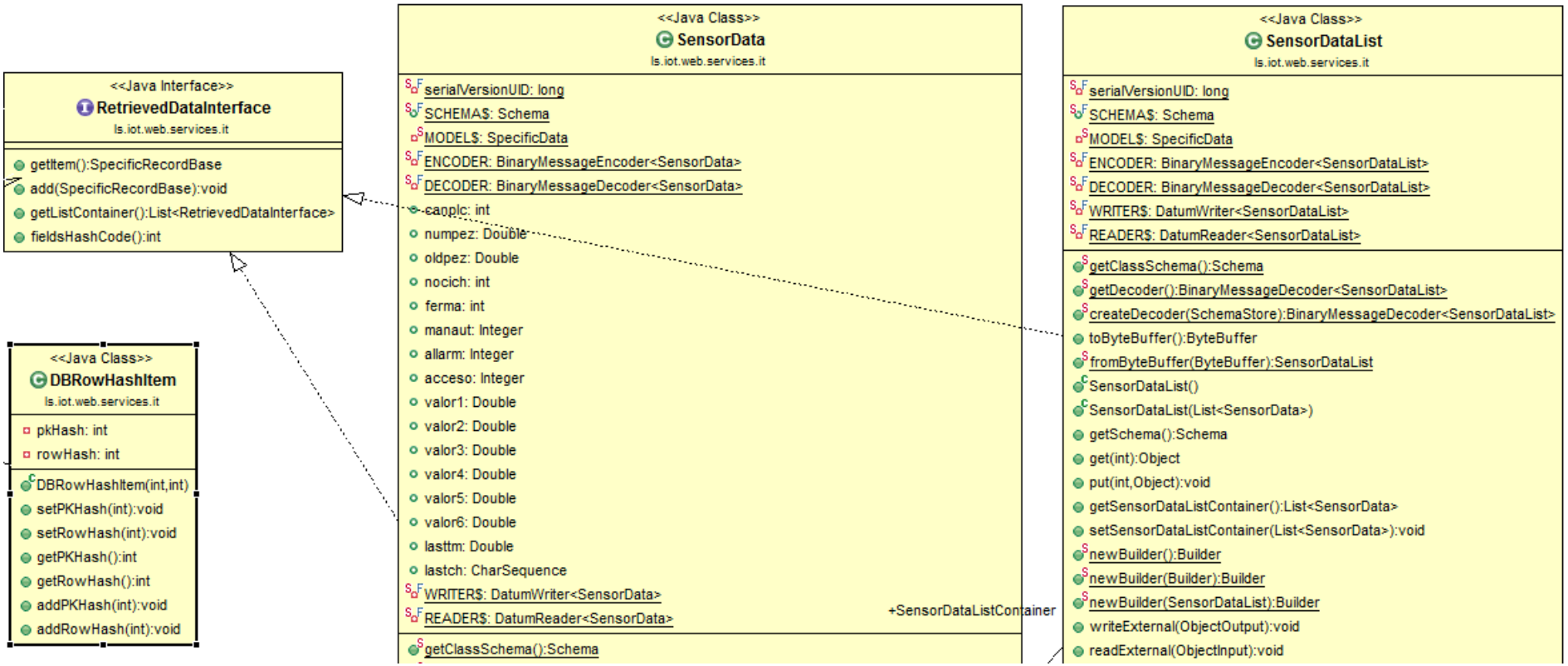
# Albero della condition



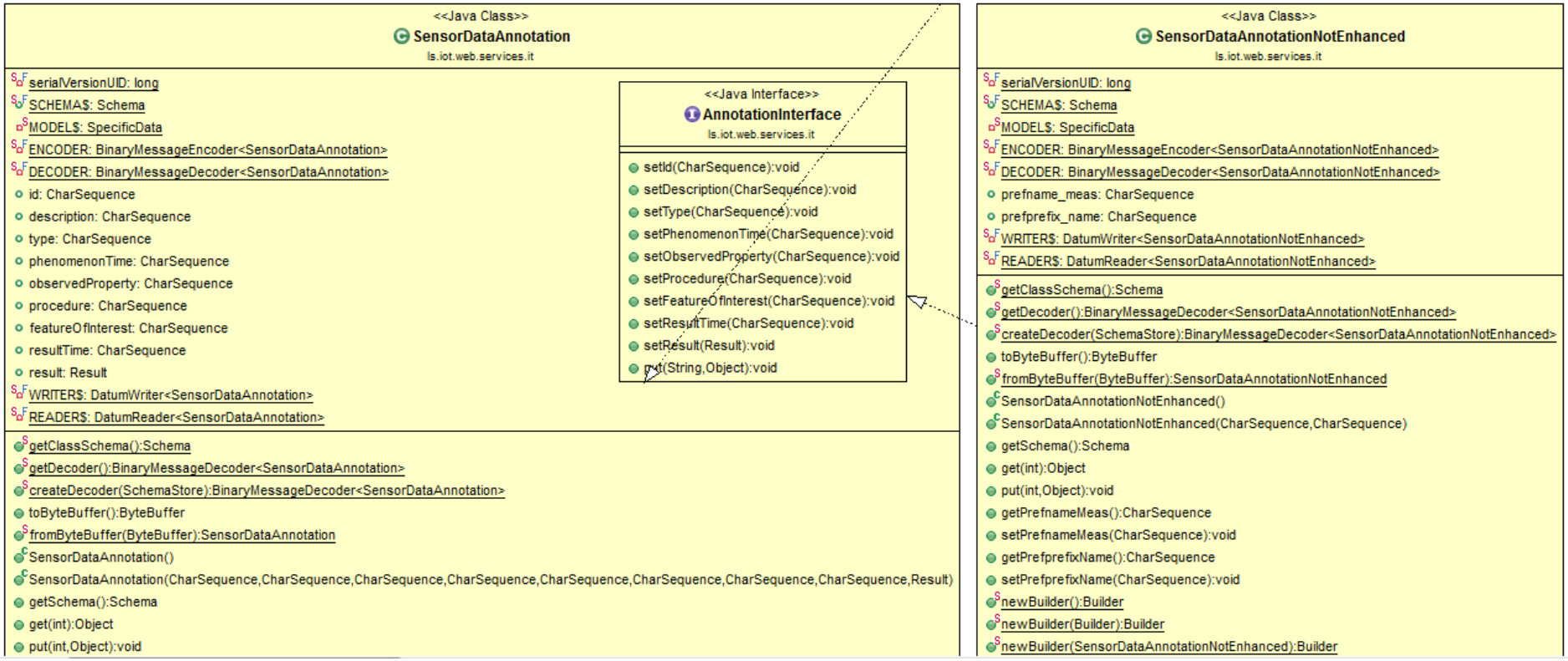
# Class Diagram SubscribeRuleInterface



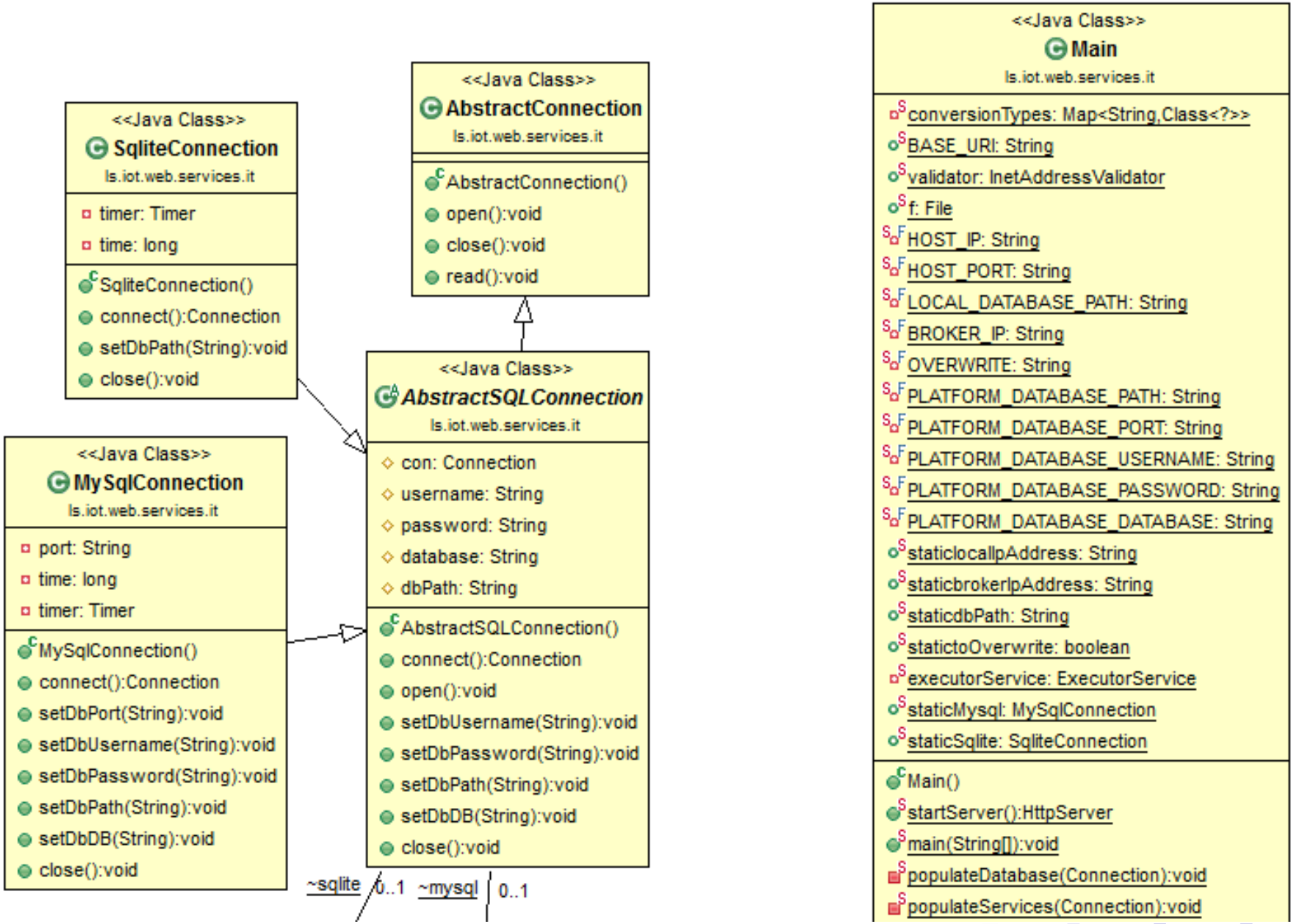
# Class Diagram RetrievedDataInterface



# Class Diagram AnnotationInterface



# Class Diagram AbstractConnection



# Schema Avro SensorData

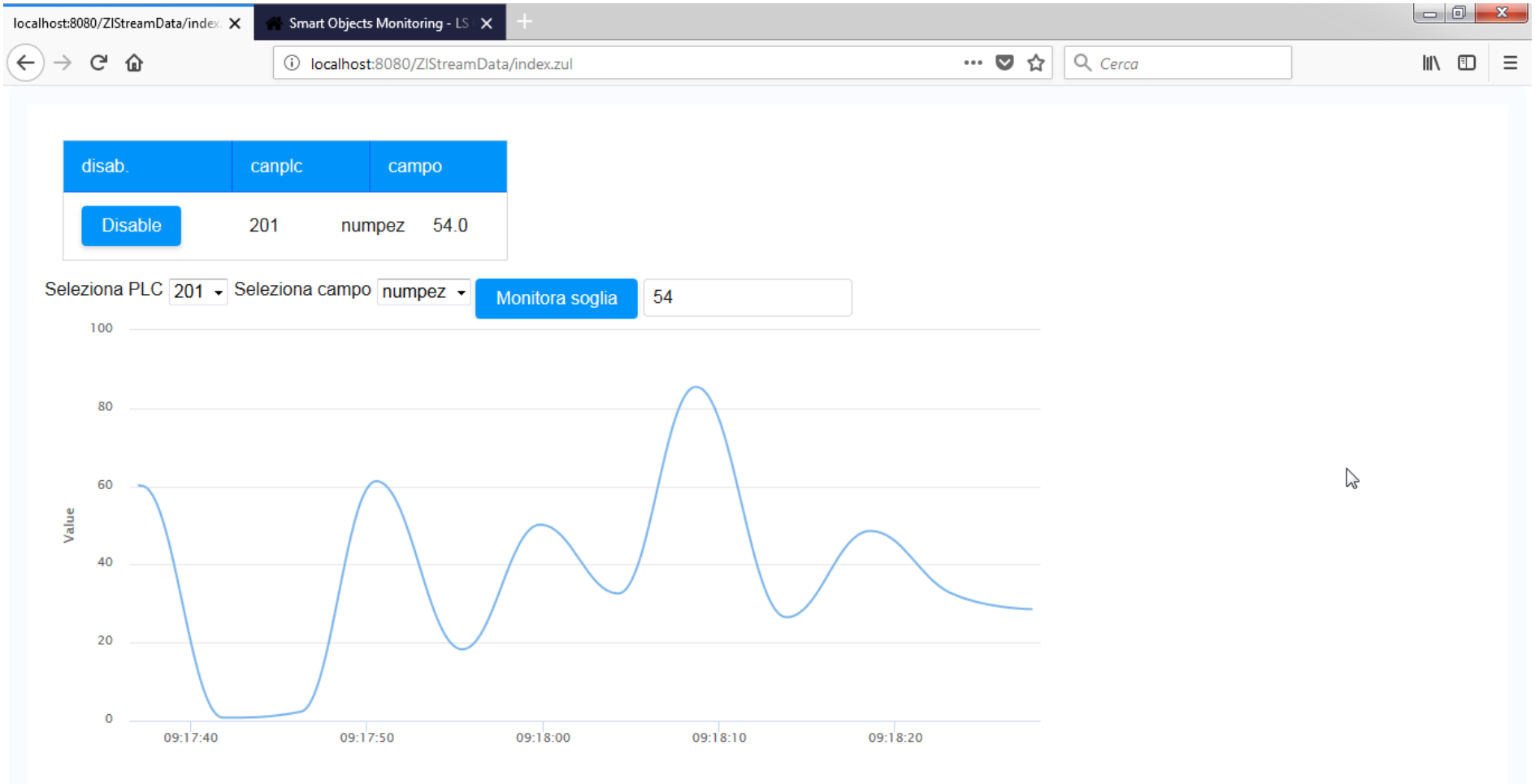

```
{
  "namespace":"ls.iot.web.services.it",
  "name":"SensorDataList",
  "type":"record",
  "fields":[
    {
      "name":"SensorDataListContainer",
      "type":{
        "name":"SensorDataListContainer",
        "type":"array",
        "namespace":"ls.iot.web.services.it",
        "items":{
    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

# Schema Avro SensorDataAnnotation


```
{
   "type":"record",
   "name":"SensorDataAnnotationNotEnhanced",
   "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."
      }
   ]
}
```

```
{
   "type":"record",
   "name":"SensorDataAnnotation",
   "namespace":"ls.iot.web.services.it"
   "fields":[
      {
         "name":"id",
         "type":"string",
         "doc":""
      },
      {
         "name":"description",
         "type":"string"
      },
      {
         "name":"type",
         "type":"string"
      },
      {
         "name":"phenomenonTime",
         "type":"string"
      },
      {
         "name":"observedProperty",
         "type":"string"
      },
      {
         "name":"procedure",
         "type":"string"
      },
      {
         "name":"featureOfInterest",
         "type":"string"
      },
      {
         "name":"resultTime",
         "type":"string"
      },
      {
         "name":"result",
         "type":{
            "name":"Result",
            "type":"record",
            "fields":[
               {
                  "name":"value",
                  "type":[
                     "double",
                     "int",
                     "string"
                  ]
               },
               {
                  "name":"uom",
                  "type":"string"
               }
            ]
         }
      }
   ]
}
```

# Pagina web per il grafico



# Codice Job Flink

```
@Override
public void run() {
    try {
        final StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();

        DataStream<SensorData> messages = env.addSource(source);
        DataStream<Double> counts = messages
                .filter(new MyFilterFunction<SensorData>(canplc))
                .map(new MyMapFunction<SensorData, Double>(field))
                .timeWindowAll(Time.seconds(10), Time.seconds(5))
                .apply(new Avg())
                .filter(new MyAvgFilter<Double>(threshold));
        System.out.println("superati i filtri e mappature");
        counts.addSink(new MySinkFunction<Double>(canplc, field, threshold));
        System.out.println("aggiunto sink");

        JobExecutionResult re = env.execute();
    } catch (Exception e) {
        System.out.println("Flink job terminated");
        e.printStackTrace();
    }
}
```

# Difficoltà incontrate

- Integrazione subscribe con NAV
  - Utilizzo Web Service SOAP

# Tecnologie utilizzate

- Java
- Framework Jersey e Grizzly
- Apache Avro
- Apache ActiveMQ
- Framework ZK
- Apache Flink

# Risultati raggiunti

- Piattaforma indipendente
  - Classi astratte e interfacce
  - Database SQLite per autenticazione token
- Servizio subscribe debolmente accoppiato
  - Tramite message broker
- Servizio monitoraggio dei dati
  - Grafico per visualizzare andamento
  - Apache Flink per controllo soglia

# Obiettivi

- Interazione di Microsoft Dynamics NAV con la piattaforma LS-Genio Mashup e definizione di un "setup" per l'utente
- Realizzazione di un ontologia delle misurazioni e delle misure

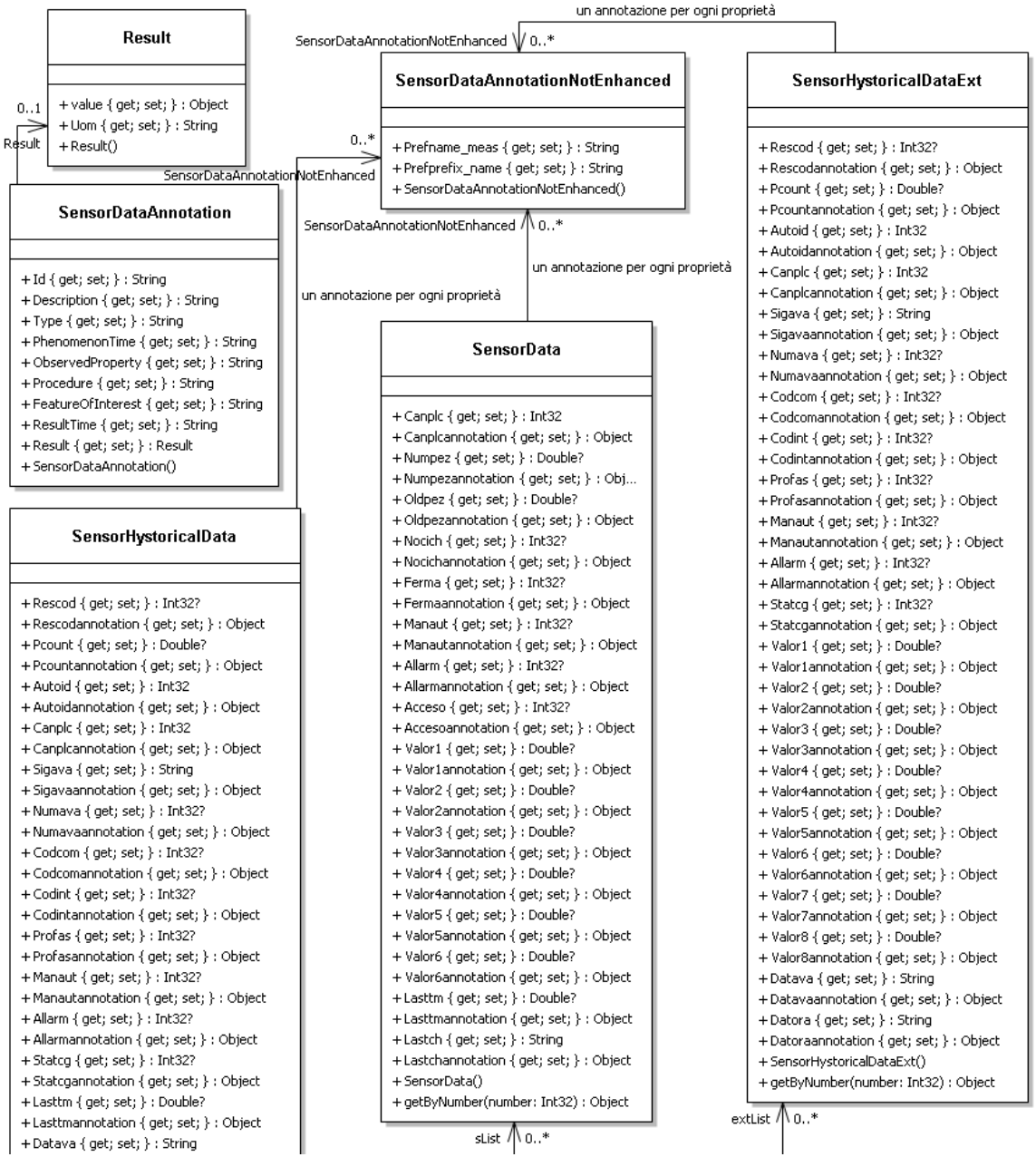
# Problematiche e risoluzioni

- Software Microsoft Dynamics NAV che possiede numerose limitazioni, ostacolando l'interazione con la piattaforma
  - Risolto mediante implementazione di un client C#, integrato poi su NAV
- Difficoltà nel trovare un modello ontologico relativo al case study
  - Risolto mediante adattamento allo standard ISO 19156:2011

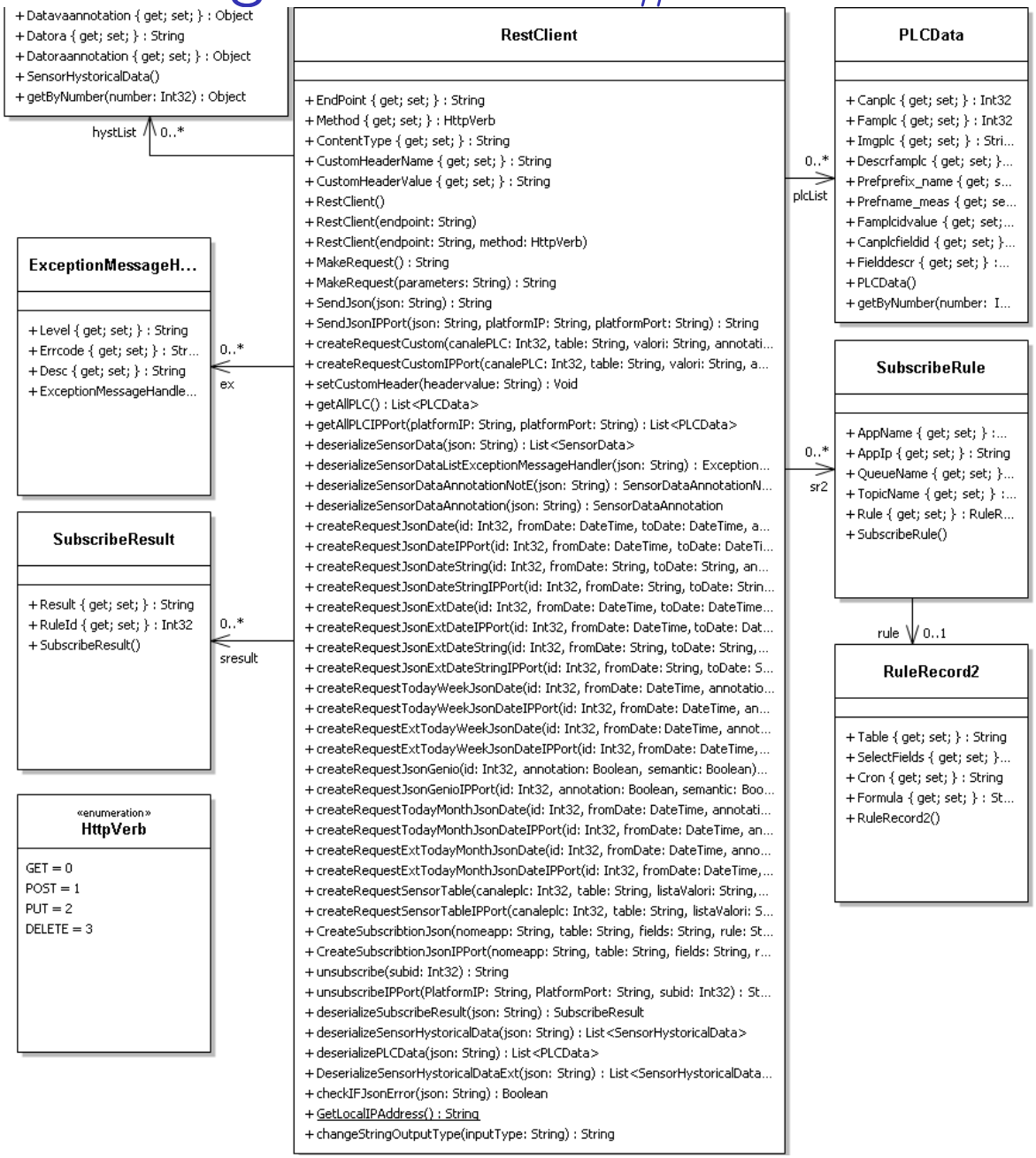
# Tecnologie e software utilizzati

- C#
- Apache Avro
- Microsoft Dynamics NAV e C/AL code
- Microsoft PowerBI
- Protégé
- MySQL

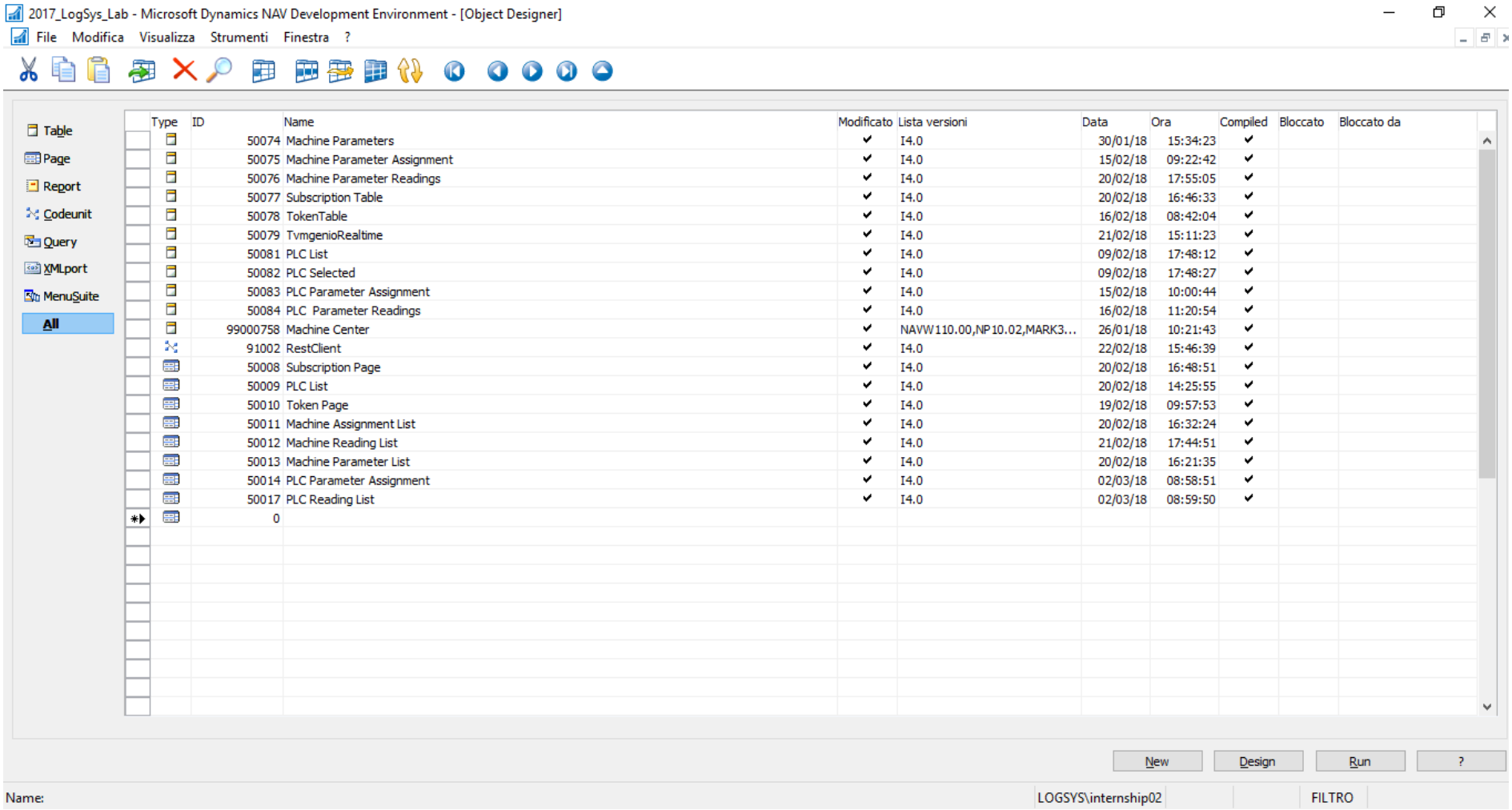
# Class Diagram Client C# 1



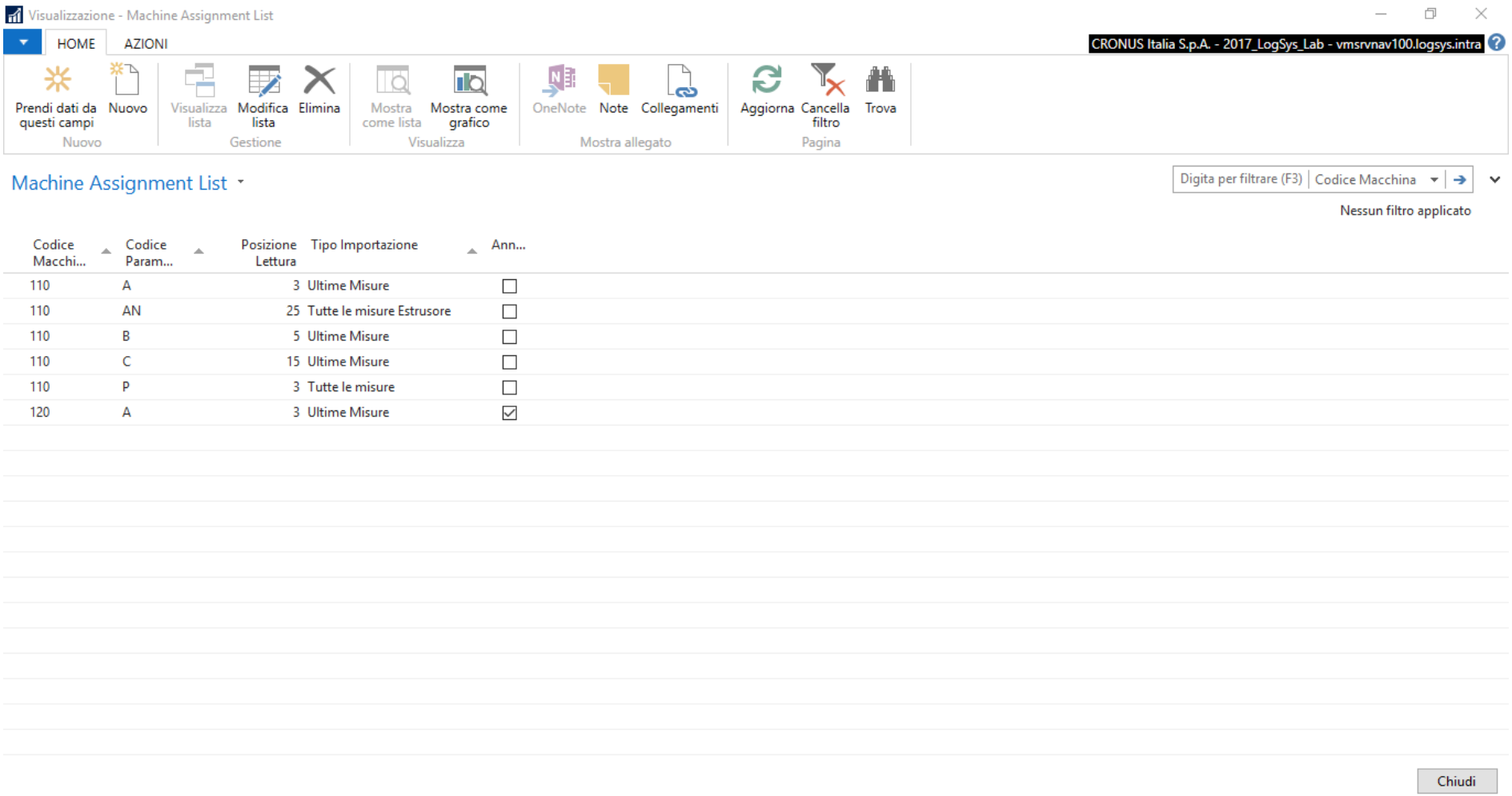
# Class Diagram Client C# 2



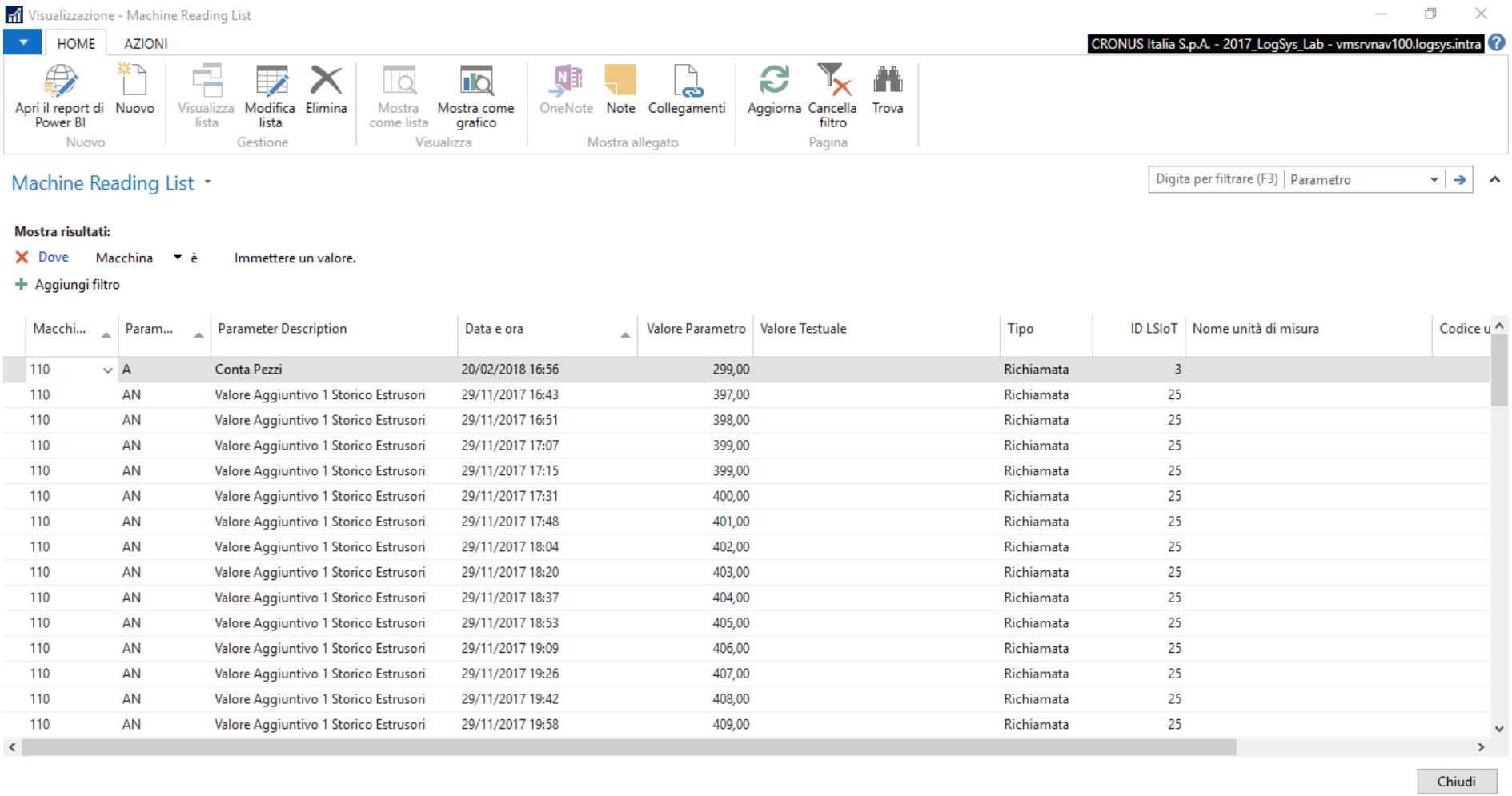
# Ambiente di sviluppo (C/SIDE) NAV



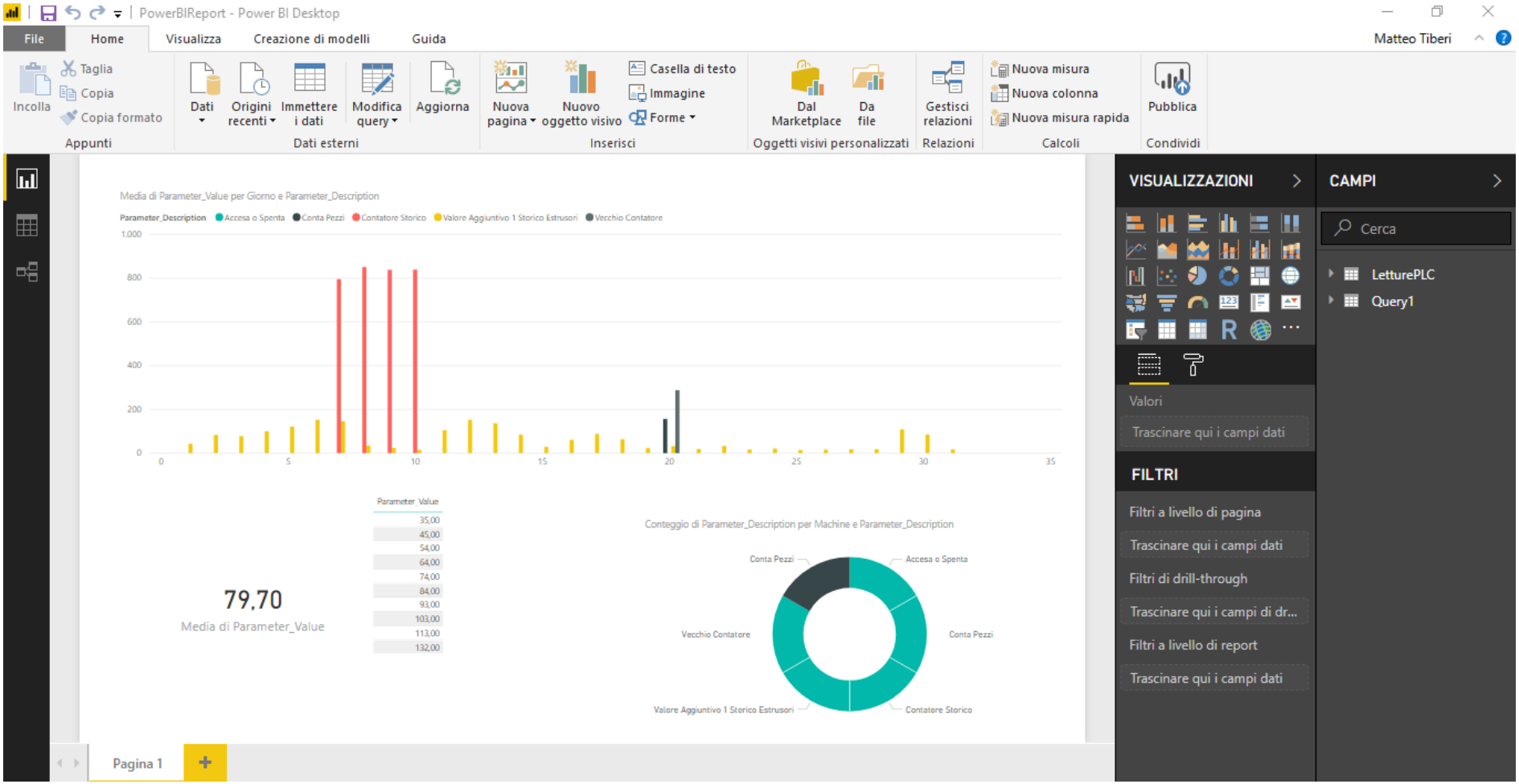
# La pagina Machine Assignment List



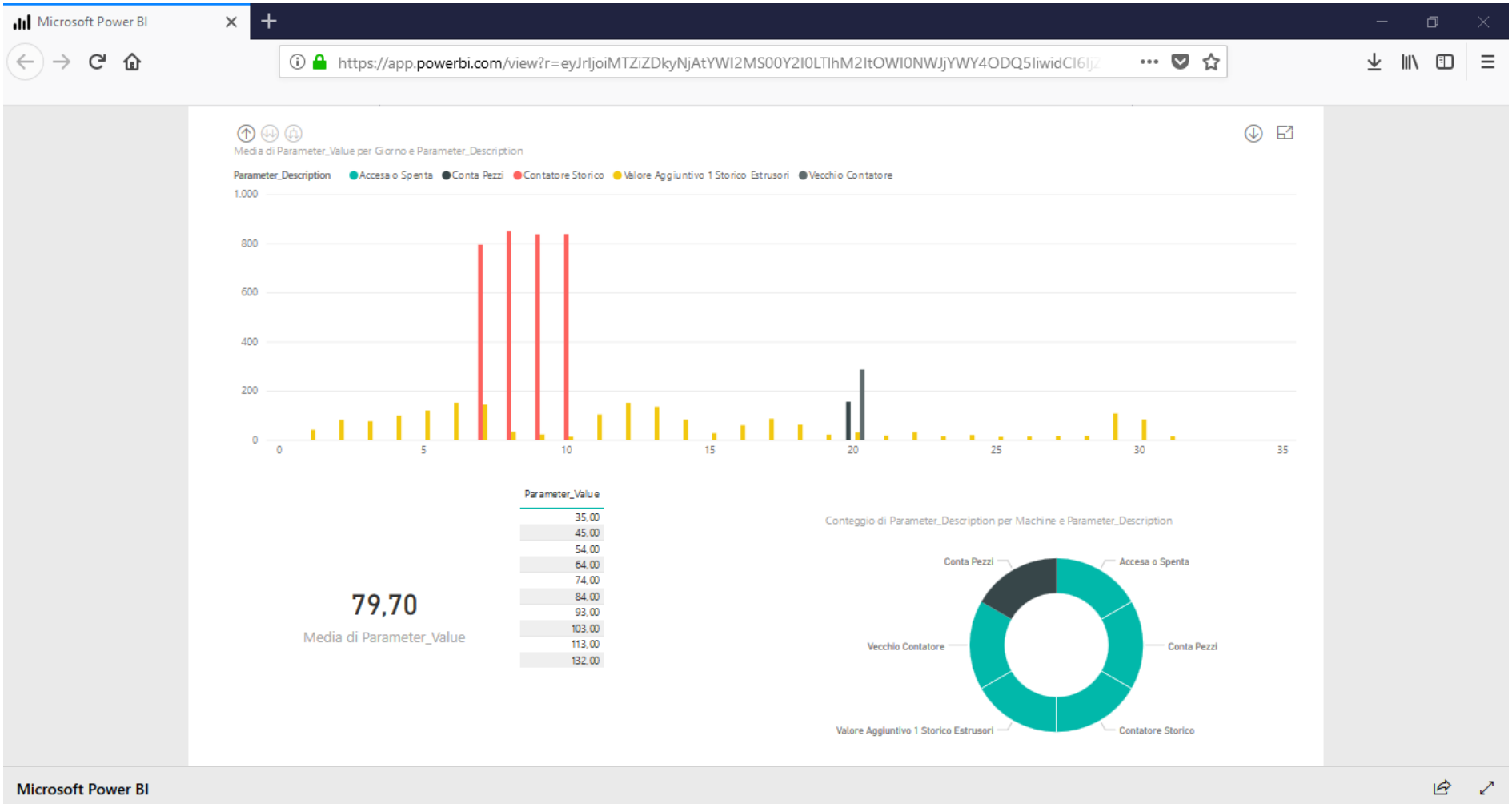
# La pagina Machine Reading List



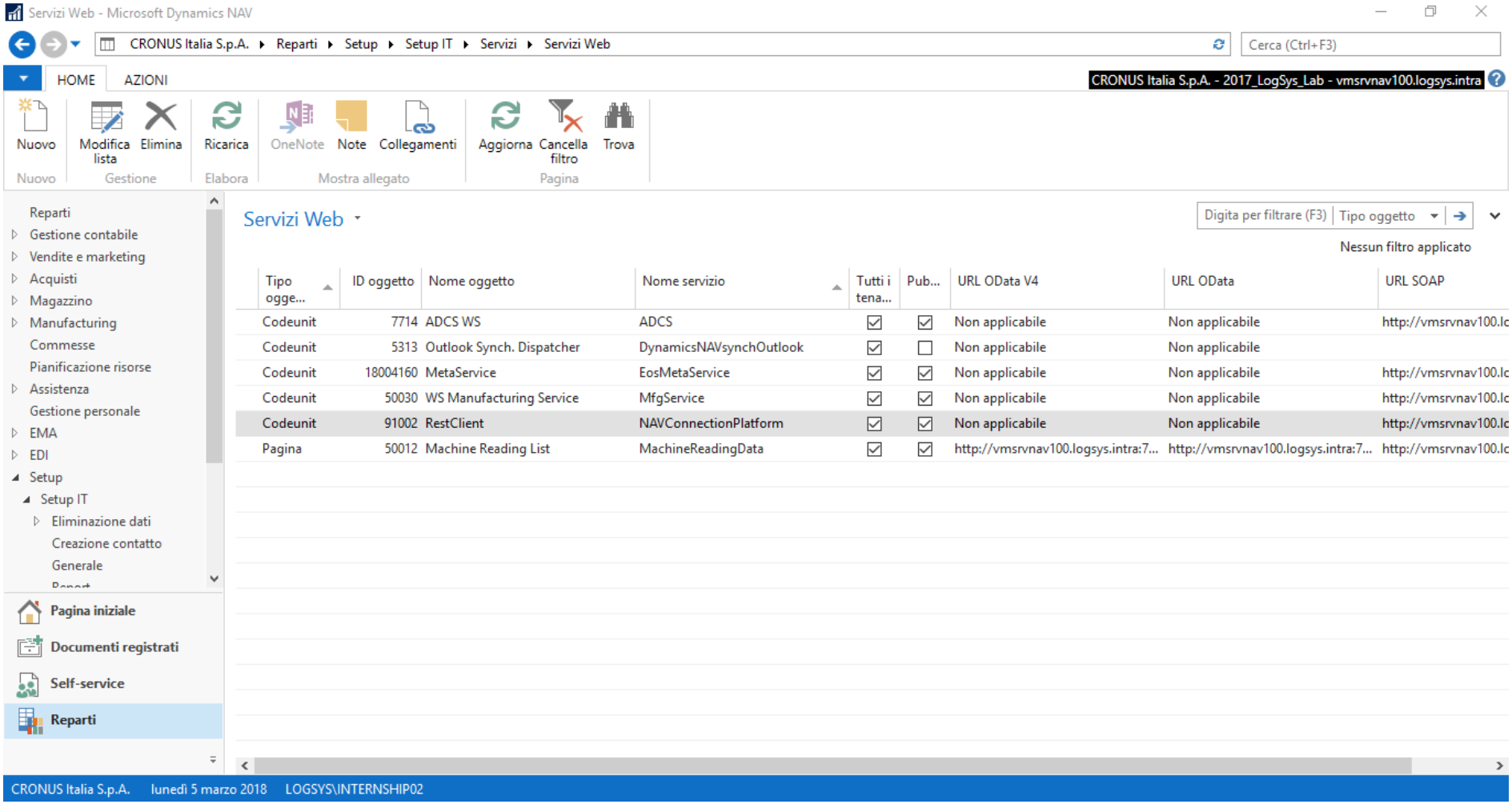
# Il report PowerBI nell'applicativo



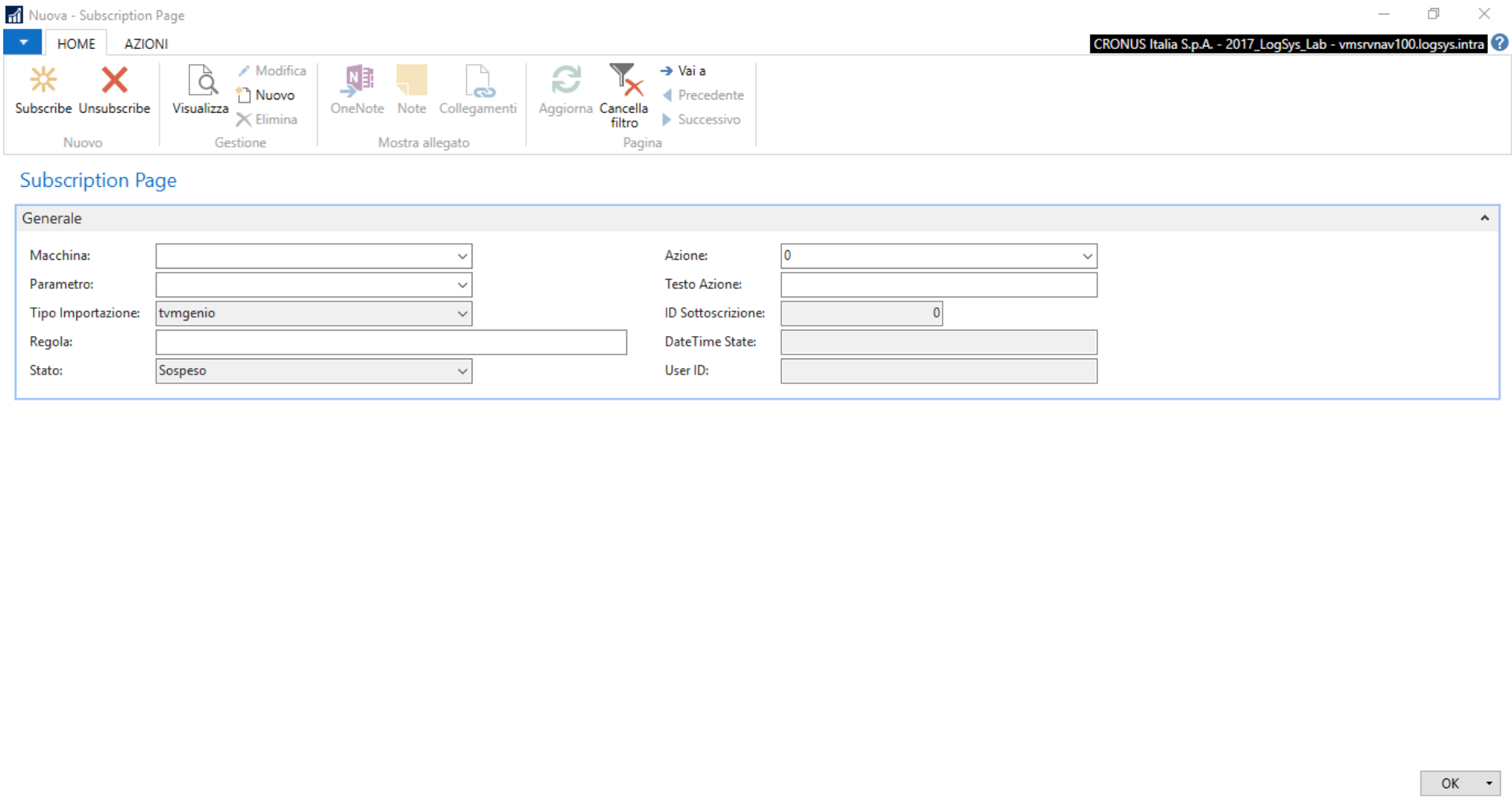
# Il report PowerBI esportato nel web



# NAV servizi web



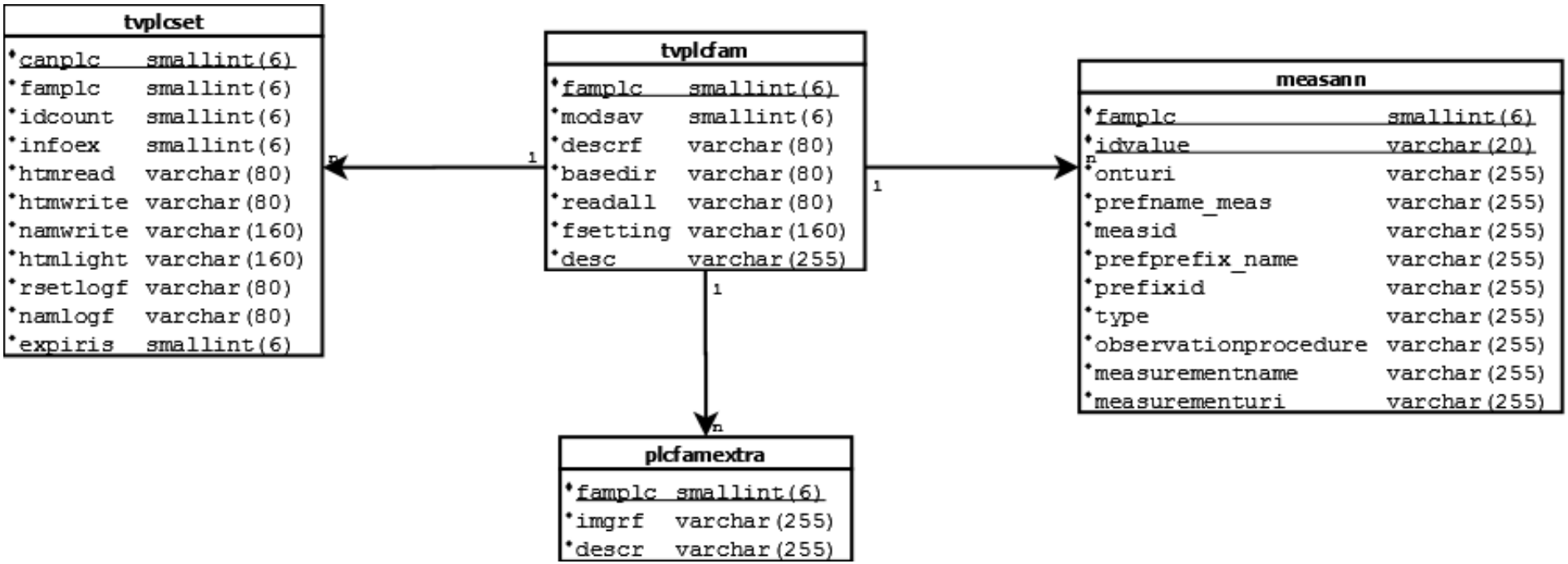
# La pagina SubscriptionPage



# Standard Observation and Measurement ISO 19156:2011

- Standard basato sul concetto di osservazione, con implementazioni in formato XML e JSON
  - Pensato per l'ambito geospaziale, il modello risulta astratto e applicabile nel case study
- Concetto di osservazione generico specializzato in base al risultato (es. Measurement)
  - Solo alcune specializzazioni sono utilizzate nel case study
- Al risultato di una osservazione specializzata viene poi associata un ontologia delle misure

# Tabelle Backend Annotazioni



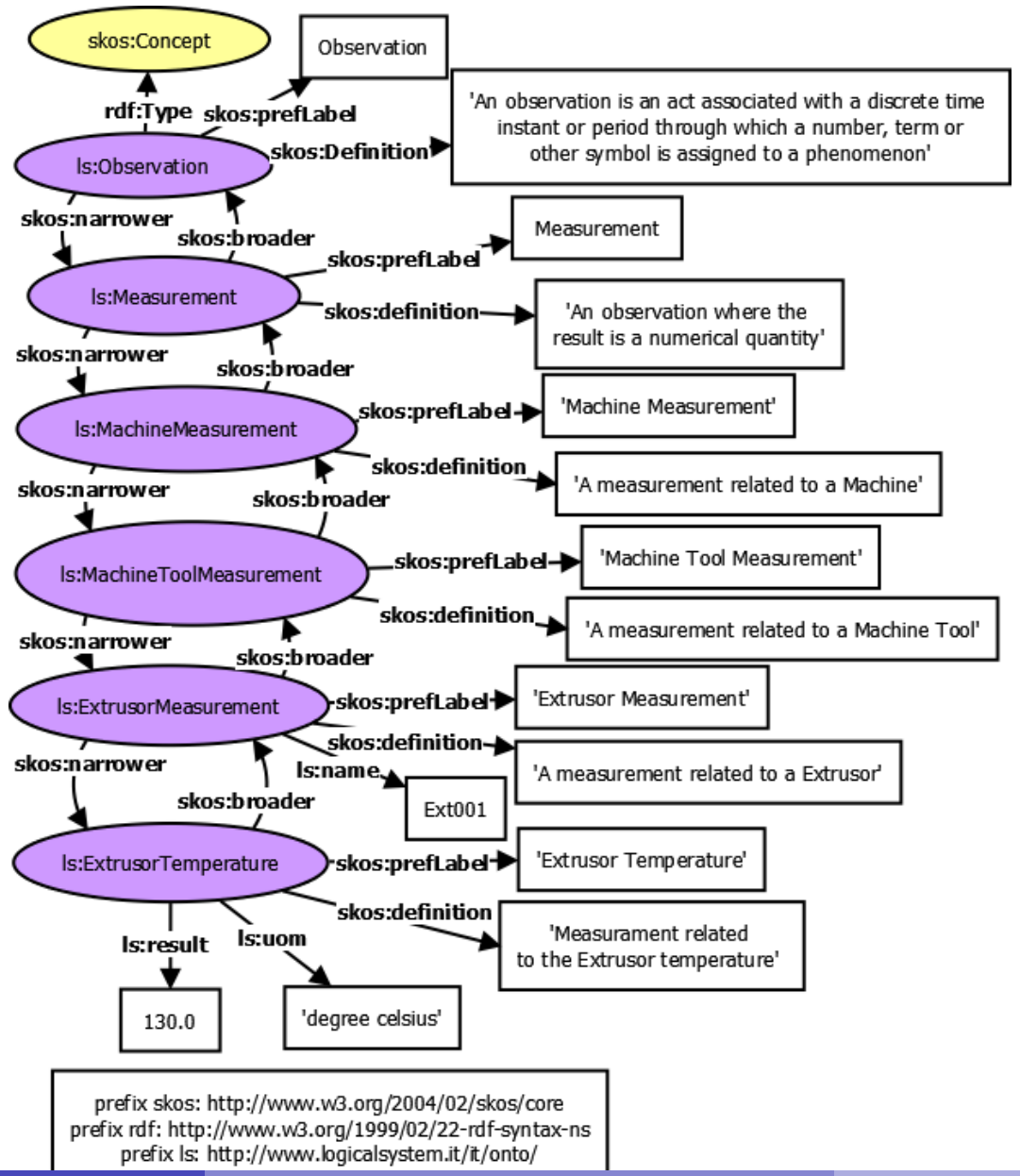
# Esempio XML

```
<om:OM_Observation xmlns:om="http://www.opengis.net/om/2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:gml="http://www.opengis.net/gml/3.2" gml:id="valor1106"
xsi:schemaLocation="http://www.opengis.net/om/2.0
http://schemas.opengis.net/om/2.0/observation.xsd">
  <gml:description>
     Valore della temperatura per l'estrusore Ext001 relativo al PLC 106
  </gml:description>
  <gml:name>valor1106</gml:name>
  <om:type xlink:href="http://www.opengis.net/def/observationType/OGC-OM/2.0/OM_Measurement"/>
  <om:phenomenonTime>
    <gml:TimeInstant gml:id="ot1t">
       <gml:timePosition>2018-02-20 16:56:54</gml:timePosition>
    </gml:TimeInstant>
  </om:phenomenonTime>
  <om:resultTime xlink:href="#ot1t"/>
  <om:procedure xlink:href="http://www.logicalsystem.it/it/register/process/sensorGenio.xml"/>
  <!--  a notional URL identifying a  procedure ...  -->
  <!--  environmental conditions during measurement  -->
  <om:observedProperty xlink:href="http://www.logicalsystem.it/it/onto/ExtrusorTemperature"/>
  <!--  a URN identifying the observed property  -->
  <om:featureOfInterest>canplc106</om:featureOfInterest>
  <!--
     a notional WFS call identifying the object regarding which the observation was made
  -->
  <om:result xlink:href="http://purl.obolibrary.org/obo/UO_0000027">130.0</om:result>
  <!--
     The XML Schema type of the result is indicated using the value of the xsi:type attribute
  -->
</om:OM_Observation>
```

# Esempio JSON


```
{
    "id": "valor1106",
    "description": "Valore della temperatura per l'estrusore Ext001 relativo al PLC 106",
    "type": "Measurement",
    "phenomenonTime": "2018-02-20 16:56:54",
    "observedProperty": {
        "href": "http://www.logicalsystem.it/it/onto/ExtrusorTemperature"
    },
    "procedure": {
        "href": "http://www.logicalsystem.it/it/register/process/sensorGenio.xml"
    },
    "featureOfInterest": "canplc 106",
    "resultTime": "2018-02-20 16:56:54",
    "result": {
        "value": 130.0,
        "uom": "http://purl.obolibrary.org/obo/UO_0000027"
    }
}
```

# Grafico misurazioni e misure



# Pagina web ExtrusorTemperature

Ontologies Classes Object Properties Data Properties Annotation Properties Individuals Datatypes Clouds

**Contents**

- All Ontologies
- Classes (1510)
- Object Properties (118)
- Data Properties (3)
- Annotation Properties (165)
- Individuals (18)
- Datatypes (25)

**ncit**

- Classes (1124)
- Object Properties (99)
- Data Properties (2)
- Annotation Properties (118)
- Datatypes (25)

**uo**

- Classes (383)
- Object Properties (3)
- Data Properties (2)
- Annotation Properties (35)
- Datatypes (3)

**skos**

- Classes (7)
- Object Properties (20)
- Data Properties (3)
- Annotation Properties (21)
- Individuals (18)
- Datatypes (2)

Ontologies Classes Object Properties Data Properties Annotation Properties Individuals Datatypes Clouds

Individual: 'Extrusor Temperature'

**Usage (1)**

- 'Extrusor Measurement' skos:narrower **'Extrusor Temperature'**

**skos:definition**

- "Measurement related to the temperature of the Extrusor" @en

**skos:prefLabel**

- "Extrusor Temperature"

**uom**

- "degree celsius"

**skos:broader**

- 'Extrusor Measurement'

OWL HTML inside

# Conclusioni

- L'integrazione tra NAV e la piattaforma ha avuto esito positivo tramite uso del client C#
  - Permettendo agli utenti un semplice utilizzo dei servizi
- L'ontologia delle misurazioni e delle misure è stata implementata
  - In modo da avere una descrizione dei dati ottenuti dai servizi