Consegne prof

Microservizi di stato (ora, data)  
Microservizio meteo  
Microservizio numero persone (IOLO)

Attuatore(acceso/spento)  
Attuatore(acceso/spento, byte)  
Attuatore(acceso/spento, 3 byte)

Decisore e profiler per ultimi

Visualizzatore(interfaccia di polling o di interrupt)

Restlet, springboot, publish-subscribe

OSGI Continuos integration  Guarda i jar dentro una cartella e li esegue;

Systema deployment per i servizi  
Architettura sborona con Maven, interrupt al posto del polling

# Ricerca

REST architecture:   
un’architettura REST è un set di risorse collegate, che formano un sottoinsieme del web, mostrati come microservizi.  
<https://restlet.com/use-cases/api-first/rest-apis/>

Restlet: REST API, libreria per creare e gestire un’architettura REST, sistema di comunicazione tra i microservizi;  
URI: Uniform Resource Identifier, la stringa per identificare una risorsa e accedervi via internet (in parole povere un URL).

Spring MVC: Framework(libreria) per creare applicazione web secondo un modello Model View Controller, i primi due termini rappresentano input e output, il controller elabora gli input e passa i dati alle view. In spring gli input sono le classi modello da gestire e le view sono file JSP(Compilati in HTML)

<https://www.html.it/pag/44655/spring-mvc-introduzione-2/>

Springboot: E’ un configuratore dello Spring framework, crea gli XML necessari senza che sia tu a scriverli. Crea un progetto che “è server di sé stesso”, non ha bisogno di un gestore di servizi, il “.jar” prodotto include nativamente Apache Tomcat (o altri gestori) che avviano il tuo servizio. Serve a non dover gestire il deploy dei vari servizi.  
Per installare springboot su eclipse:  
<https://www.codejava.net/ides/eclipse/install-spring-tool-suite-for-existing-eclipse-ide>  
Per scrivere un microservizio springboot con Maven:  
<https://www.springboottutorial.com/creating-spring-boot-project-with-eclipse-and-maven>  
Le annotations sono dichiarazioni delle funzioni per mappare il loro funzionamento e sono obbligatorie per compilare.

Maven: “Accumulation of knowledge”, gestisce le builds, le dependencies + versions e la documentation di un progetto, gira tutto intorno ad un file principale di configurazione: “pom.xml”

Per quanto riguarda OSGI, esso utilizza un formato differente da spring, infatti non usa i jar ma i bundle. Questo potrebbe portare a problemi soprattutto con le dipendenze, anch’esse in formati diversi. Però sembrerebbe possibile scrivere microservizi con osgi. È un macello spesso e volentieri.

<https://www.theserverside.com/news/1363825/OSGi-for-Beginners> ← cos’è osgi e come funziona

In realtà non riesco ancora a capire a cosa ci serve OSGi

Per installare il progetto Demo in qualsiasi pc:

1. Importare come maven project;
2. Eseguire una compilazione maven clean;
3. Eseguire una maven install per scaricare tutte le dipendenze e le librerie necessarie
4. Si può lanciare l’applicazione come spring app;