



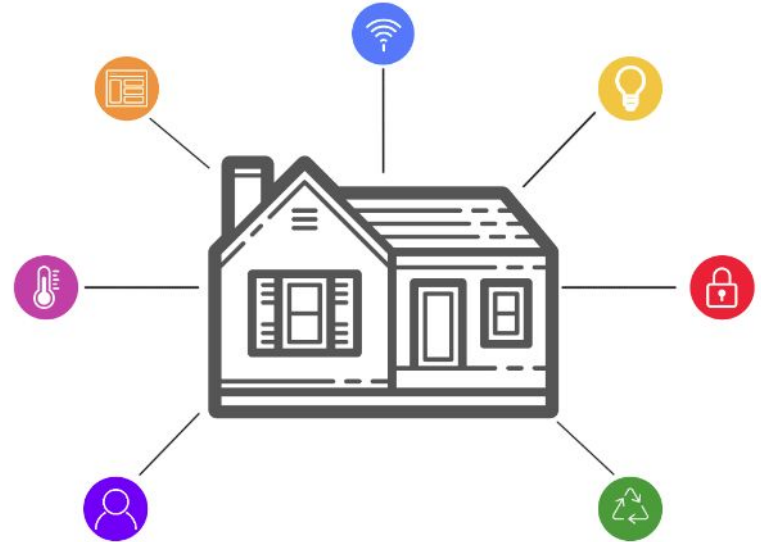
Domotics

Software Engineering 2017/2018 - Group 302

Ana Rita Santiago, 72099
António Ramos, 30928
Filipe Macário, 64618
Francisco Cunha, 76759
Raquel Ramos, 76538
Ricardo Silva, 68067

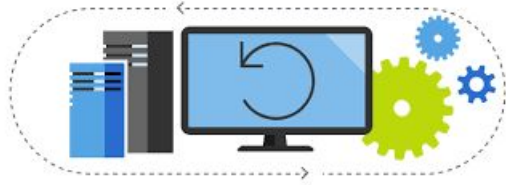
Project's Presentation

Domotics is a platform that allows the management and monitoring of smart homes. The user has the possibility to control the environment of his house using our dashboard. It is also possible to set alerts so that important events of the month are not missing or if there is a change, you can change it.



System's Scenarios

System - entity that makes “intelligent decisions” based on the information received from the sensors.



UC-5

UC-6

UC-7

UC-8

UC-1: Visualize alerts in the dashboard.

UC-2: Visualize real-time values in tables.

UC-3: Visualize previous values in charts.

UC-4: Define routine or specific alarms.

UC-5: Get values from sensors.

UC-6: Display values from sensors in the dashboard.

UC-7: Modify values that are above/below the threshold.

UC-8: Send alerts to dashboard.

User - a person who uses the dashboard and the system.



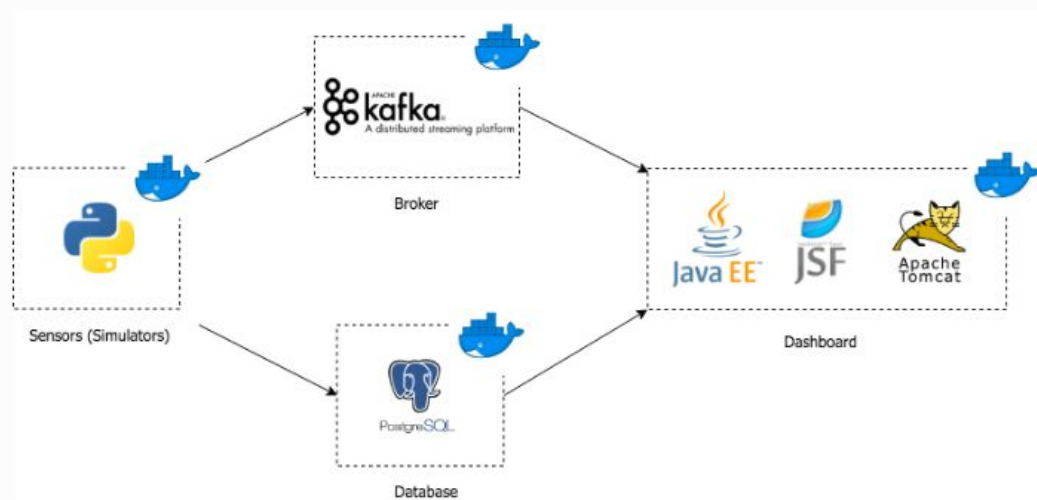
UC-1

UC-2

UC-3

UC-4

Architecture



For our project it was necessary to define 3 groups of main components that interconnected each other simply and efficiently. So it was chosen to use **PostgreSQL** as database, **Kafka** as broker and **JavaServer Faces** for dashboard's implementation.

All components are defined into docker containers and our platform runs over **Apache Tomcat**. Besides that the input data comes from home's sensors and we use **Python** to import info and **Java** to export it.

Why those technologies?

- relational database
- most advanced and extensible system.
- enables interaction with object-oriented languages

PostgreSQL



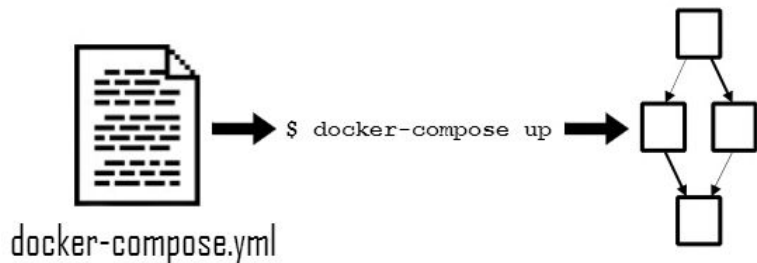
- allows the exchange of messages between different topics

- Java specification for building web application interfaces
- event oriented



Deployment

Full system with sensors, broker, database and consumer



System with Kafka broker and database PostgreSQL installed

1. Edit file `docker-compose_vm.yml`: change command to use broker (`<kafkamachine>:9092`) and change type of value sent from sensors (`<simtype>`).

2. `docker-compose -f docker-compose_vm.yml up.`



Questions and Answers

