Systems Integration

3rd year • 1st semester • Academic year 2019/2020

Project

IPLeiria SmartCampus

Implementing Smart Information and Communication Technology (ICT) for well-being, energy efficiency and surveillance in public buildings

Iteration 1 – Temperature and Humidity

What is the project about?

Internet of Things will have a big impact in our society in terms of security, surveillance, well-being and even on climate change. Cities, plants, buildings, offices, homes, cars will be turned into smart places and appliances that can save energy costs, by measuring and controlling electricity consumption and even correcting our bad behaviors, such as letting lights and temperature systems turned on after left the infrastructure. Homes and buildings will bring more comfort to our life by automatically setting the environment (e.g. temperature, humidity, media appliances and alike) according to our preferences, and detect just-in-time resource wasting such as water and gas leaks. IoT will also impact on access control and surveillance by relying in real time monitoring. The impact is so large that even the climate change problem would benefit from it.
If we consider our school campus or the entire Polytechnic of Leiria schools' campus, the employment of IoT could have an interesting impact at our scale. Hence, this project deals with implementing smart information and communication Technology (ICT) concepts in public buildings, of course, following an evolutionary approach as we will improve the system with new features and capabilities year by year.

The **IPLeiriaSmartCampus** solution consists in the development of an integrated platform that relies on IoT in order to improve energy efficiency, well-being and surveillance in all the Polytechnic of Leiria campus.
In this first project iteration only the Campus 2 is to be considered but the result solution must be flexible enough to be possible to be applied to any Polytechnic of Leiria Campus. More specifically, the first project iteration deals with temperature and humidity of Library only. Hence, the main aim of the first project iteration is to record temperature and humidity of the **Library** building. Once the whole campus is covered by WiFi network it could be a good idea to rely on this network in order to grab sensors data. Due to the lack of suitable hardware (sensor nodes), in this first project iteration you must be able to deal with simulated sensors, which means that a kind of bridge device (and software) will be needed in order to push simulated sensor's data into the WiFi network.

As far as simulated sensors is concerned, you must assume that library building already includes an old and local temperature and humidity monitoring system that beeps when temperature or humidity falls outside safety (considering books and other contents) values. It is known that this system is running in the receptionist computer and apart from beeping it also logs temperature and humidity into a local binary file. When the local system logs data to the binary file (data.bin) it also presents on console the following information:

SENSOR ID: 1
TEMPERATURE: 22.42
HUMIDITY: 47.76
BATTERY: 99
TIMESTAMP: 1571568195 (2019-10-20 11:43:15)

SENSOR ID: 2
TEMPERATURE: 22.83
HUMIDITY: 47.74
BATTERY: 98
TIMESTAMP: 1571568195 (2019-10-20 11:43:15)

As the file structure is not known, you must rely on this printed information in order to guess and implement an application in order to also forward this information to the WiFi network. Considering the printed information, it is also possible to guess that the local monitoring system includes 2 physical sensors (id: 1 (upstairs) and id: 2 (downstairs)), one per building floor and battery data ranges from 0 to 100%. An Unix executable file is provided to mimic the operation of the library monitoring system in your laptop and you can assume that you have rights to install your "connector" into the receptionist computer (in this case, in your laptop).

The simulated sensor's data must be forward into a system where multiple (micro) applications can exist to visualize, monitor and analyze that data.
For now, two **local** desktop applications must be developed. The first one must be able to show real-time temperature and humidity data per floor in tabular and chart formats. There is no need to locally store sensor data. However, while active, the application must keep and show the extracted sensor data since the moment it is opened until the moment it is closed.

The second micro application must be able to configure and trigger alerts, according to the conditions: >, <, = and between and each configured alarm could be enabled or disabled. All required configuration must be stored in an appropriate local file and generated alerts must be shared.

Apart from (micro) applications, sensor's data must be forwarded into a database and a globally API must be available to give the chance to school community to access and use the collected data, alerts, etc., following different interaction approaches, probably in community applications or applications developed in other curricular units. API must include, at least, register a 'personal' sensor, **add new** sensor data, **invalidate** sensor data (for the case of malfunctioning sensors), **get** sensor data per floor with or without date interval, get a list of all the sensors and get alarms data. Remember that API and other kind of data sharing are only available to qualified users/clients.
Optionally, the API may include also actions to store temperature sensors' data coming from community mobile devices (personal sensors).

Your work is to implement the solution and write the project report. Remember that the System Architecture of the overall solution **must** be included in the project report and well explained. It is mandatory! As another tip, we advise you to first draw the system architecture and only after that start implement it.

Project Assessment

The project evaluation criteria are as follows:

Criterion	% [0-100]
Library Sensors to WiFi bridge	20 %
Data Show Micro Application	10 %
Alerts Micro Application	20 %
Data Storage Micro Application	10 %
Global API	20 %
Project report	20 %

Guidelines

Groups must be formed by 4 students, which can be from different practical classes. The name and number of the group members must be submitted in the courses page (http://ead.ipleiria.pt - Moodle) until the end of October.
The project delivery must be performed until the date published in the official assessment calendar. It has a mandatory project presentation (and oral discussion) that will occur in a later date, also published in the course assessment calendar.

Besides the source code of the entire solution, it is also mandatory to deliver a **written report** following the published template, which also includes appendixes. The template for the report will be available in the course web page. Remember to tag your source code (using text headers) because, as this project is an evolutionary project, your source code could be elected to be used as a starting point in the next school year.

Project Delivery

The project delivery must be done in the course web page (http://ead.ipleiria.pt) where students must submit a link to an external ZIP file (or .7z file) with all the project material. The link to the project delivery should use an external cloud service, e.g. Dropbox (shared file), WeTransfer, etc.
Therefore, students must guaranty that all the following material is included in a ZIP file (.7z) when delivering the project, and with the following organization structure:

- Text file (**identification.txt**) with all the information about the team members (number, name and e-mail) and course name, etc.
- Folder **Project**: all the source code of each module must be included in this folder. The source code of each module should be in a different folder to clearly identify each project. For each project, you must include: source files and other resource (files, executables, dll's, etc.)
- Folder **Report**: the written report in docx format. Don't forget that in the end of the report (appendix) it is mandatory to identify which features has each team member implemented.
- Folder **Data**: database files (.mdf and .ldf). The database must have data, in order to be able to test the project. Also, it is recommended to deliver also a database script (.sql file) including the database structure and data /database records. If using a database in the cloud, the necessary credential to connect to the database should be also provided.
- Folder **Other**: other files required by the project that were used by the team members but were not included in the previous folders.