Development and implementation of an API for the management of computing services in the cloud of CUValles



**Smart**CUValles

[1] Icon API

**Description of modules and basic functionality**

# Introduction

The SmartCUValles project aims to develop and implement a 3D interface on the web from which it is possible to monitor the environment (climate, wind speed, pollution, humidity, among others) and control actuators (projection cannons, blinds, doors, lights , among others) installed in a university center in order to generate greater energy savings.

The environment will be monitored by sensors installed in the CUValles and which in turn will be connected to network cards from which the information will be sent to an API (Application Programming Interface). The actuators will be connected in the same way to network cards that will report the state they are in (off, on, suspended). The union of these systems will allow to monitor the environment and make decisions about the actions that can be applied automatically or by the user from the interface.

The project is divided into modules and each one has a person responsible for its development. Below are described in a general way:

* Monitoring and control: The control and monitoring devices will send information every 2 minutes about what happens in the CUValles. The network cards connected to the sensors will send the obtained measurements and the cards connected to the actuators will send the states that indicate the current situation of the actuator.
* Cloud: It is necessary to develop a cloud computing infrastructure in which the database and project systems (API, 3D interface, monitoring system, and actuator system) can be housed.
* API: In order to communicate the systems of the project, it is necessary to develop and implement an API that allows to access the information in the database in a safe and reliable way through the Internet, that is, only through HTTP requests. In this way it will be possible to have a single information server for the entire university center and at the same time allow new applications for the benefit of CUValles access to information, this is how it is possible to have a development environment for new applications (systems).
* 3D interface on the web: The information captured by the sensors and the states of the actuators should be shown as graphs and status buttons respectively. The user will be able to observe all this information from this system.

Currently CUValles does not have total control of its information systems. The most common conflicts resulting in the information of their systems are:

* The information is repeated in the databases.
* The information is incomplete in some databases.
* The information is only updated in some databases.
* There is no communication between the systems.
* There is no centralized database of CUValles information.

The previous conflicts cause inconsistency and resonance in the stored information. The systems act independently and in the same way their databases are.

When the API becomes the sole information server of the systems, all previous problems will be solved.

# Objectives of the API

**Overall objective:**

Develop and implement an API to provide information in JSON format stored in a single database of CUValles in order to centralize and unify the information with which the existing and future CUValles systems interact, as well as provide future developers of an environment to create new applications for the benefit of the university center.

**Specific objectives:**

* Establish migration criteria and adaptation of existing systems to the API environment.
* Provide an environment for the development of new applications.
* Promote the API as the sole source of information for CUValles and its systems.
* Establish the criteria to eliminate existing problems with the information from the CUValles databases.
* Provide communication between Smart CUValles systems.
* Provide communication between existing and future CUValles systems.

# Detailed description of the API distribution

It is necessary to perform an analysis on the basis of two people, and firstly, focused on the distribution of the network and the second on the distribution of processes and information. They may sound somewhat similar, but their differences are described below.

* From the point of view of a distribution system it is notorious the existence of two forms of communication (distribution):
* Client-server processing: The client is the network card that sends the sensor or actuator information, as the case may be, and the processing server is the receiver in that communication. Take the name of the processing server because it is valid for the information received, you must first check if the URL exists, that is, if it was predefined in the API, but it was also the object in the API does not exist and therefore its information is not available. The information is not available. The information is not available.
* Processing server-information server: This communication architecture has a client-server background structure, but now it is the processing server. The result of the processing of the information in the server is obtained as a result of a MySQL query encoded in base64 that the information server is sent to be processed, after being executed the server that sends the request is answered and this in turn Respond to the client that started the process.

Now it is analyzed from the point of view of software, that is, the distribution of processes and information:

* Module 1, communication between elements: in this module to define how communication between the elements of the API should be, this implies the necessary conditions for the information to be sent by the client, the response, the exchange of information between the server Processing and the information.
* Module 2, validation of URLs: in this module the structures of virtual and physical URLs necessary for the existence of objects in the API must be described, an object is an information service, for example: a sensor object, exists in the database, in the API and therefore your information can be consulted by a client.
* Module 3, validation of information: once the existence of the URL in the API has been verified, it is necessary to verify that the information received complies with the parameters established in the design of the service offered and that it does not attempt to corrupt the database.
* Module 4, storage of the information: this module will only be executed by the central server, it will be in charge of administering the information found in the database.

# Conclusions

It is important to note that the modules described should not be exclusively the only ones in the system, during the analysis of the system it is possible that more modules result or even that the functionality of those already described changes.

# Description of software configuration management

The project has 5 documents to specify the requirements, design and implementation of the SmartCUValles API. The documents are:

* SRS: Specification of software requirements.
* SyRS: Specification of system requirements.
* StRS: Specification of user requirements.
* DDS: System design.
* PDP: Test plan.

The set of these documents specify that and how the system should be. In this way, if a change is presented, one should look in those documents to determine how viable it is to apply it. If after analyzing the documents the change is approved the same development of the registration and registration. The proposed change must be recorded, the description of the situation in which it is proposed, the case of being approved, the description of the situation; In case of being rejected, the decision must also be well justified.

In the analysis of the documentation, the aspects of those that must be carried out must be taken into account. The results have been modified.

A free repository system is used to manage the versions.

# References

[1] Image retrieved from: https://www.iconfinder.com/icons/532742/api\_code\_coding\_gear\_programming\_settings\_window\_icon under a free license for commercial or personal use.