

# Licence Informatique

## *Projet Semestre 5*

Le projet neOCampus (<https://www.irit.fr/neocampus/>) est un projet de l'Université Paul Sabatier qui vise notamment à contrôler les consommations énergétiques du campus. L'ensemble des fluides consommés sur le campus est mesuré au travers de compteurs et de capteurs localisés sur le campus. Ces fluides sont la consommation d'eau, d'air comprimé, d'électricité. Chaque compteur ou capteur envoie régulièrement sa valeur courante. Afin d'analyser et contrôler les consommations énergétiques du campus, il est donc nécessaire de recueillir ces données et permettre au moyen d'une interface de visualisation d'afficher ces différentes valeurs.

L'objectif de ce projet est donc de réaliser une interface de visualisation des données de consommation énergétique du campus. La visualisation de ces données pourra se faire de deux manières :

- Une visualisation « temps réel » : dans ce cas, l'interface de visualisation affichera la dernière valeur pour chaque capteur connecté. De plus, afin de surveiller les potentiels problèmes de consommation (fuite d'eau, radiateur défaillant, etc.) la valeur sera mise en évidence si elle dépasse les seuils autorisés pour ce capteur.
- Une visualisation « *a posteriori* » : dans ce cas, l'interface de visualisation pourra afficher et comparer des données de consommation de 1 à 3 capteurs en même temps sur une période déterminée.

Enfin, chaque capteur a des seuils minimum et maximum à ne pas dépasser. Si la valeur en cours d'un capteur dépasse ces bornes, l'interface devra alerter l'utilisateur en mettant en évidence le capteur en question (cf. interface de visualisation « temps réel »).

### **Protocole de communication**

Pour ce projet, les capteurs seront simulés par une interface de simulation que nous vous fournissons. La communication entre cette interface de simulation et votre interface de visualisation se fera au moyen de sockets. Votre application servira de serveur pour la réception des données envoyées par le simulateur. Au lancement de votre application, une boîte de dialogue devra demander à l'utilisateur sur quel port l'application doit se connecter. Une valeur par défaut devra être proposée à l'utilisateur.

L'interface de simulation peut gérer la simulation de plusieurs capteurs en même temps, en envoyant des messages indépendants pour chaque capteur. Un capteur peut envoyer trois types de message lors de la connexion ou de la déconnexion et lors de la mise à jour de la valeur courante du capteur.

### *Message pour la connexion d'un capteur*

Ce message sera au format suivant : **Connexion** *nom\_du\_capteur description\_du\_capteur*

Avec **description\_du\_capteur** au format suivant : *type:batiment:etage:lieu*  
et :

- **type** est le type de fluide géré par le capteur en question (les valeurs possibles pour le type sont *EAU, ELECTRICITE, AIRCOMPRIME, TEMPERATURE*) ;
- **batiment** est une chaîne de caractères qui donne le nom du bâtiment dans lequel se trouve le capteur ;
- **etage** est un entier pour indiquer à quel étage se situe le capteur (0 pour le rez-de-chaussée) ;
- **lieu** est une chaîne de caractères pour décrire plus précisément l'endroit où se situe le capteur à l'étage.

### **Exemples de message :**

Connexion Capteur1\_03 ELECTRICITE:U1:1:DevantSalle202  
Connexion Capteur302 EAU:U3:2:Entre215et214

### *Message pour l'envoi de la valeur d'un capteur*

Ce message sera au format suivant : **Donnee** *nom\_du\_capteur valeur\_du\_capteur*

Où *valeur\_du\_capteur* est une valeur sous la forme d'un réel.

### **Exemple de message :**

Donnee Capteur1\_03 3.33652

### *Message pour la déconnexion d'un capteur*

Ce message sera au format suivant : **Deconnexion** *nom\_du\_capteur*

### **Exemple de message :**

Deconnexion Capteur1\_03

## Stockage des données

Une fois connectée, votre application devra réceptionner toutes les informations émanant des capteurs. En plus d'offrir des outils de visualisation, votre application devra enregistrer l'ensemble de ces données dans une base de données. Celle-ci permettra notamment de pouvoir réaliser des analyses *a posteriori* sur les données recueillies.

## Interface graphique

Votre application devra contenir 3 grandes « panneaux » : un pour afficher la valeur des capteurs en « temps réel », un autre pour analyser les données *a posteriori*, et enfin un troisième pour gérer la liste des capteurs.

### Panneau de visualisation « temps réel »

Ce panneau de visualisation devra présenter une liste exhaustive de tous les capteurs connectés, et renseigner leur localisation (nom du bâtiment, étage, pièce) et leur valeur actuelle.

L'ensemble de ces données sera affiché sur l'interface sous la forme d'un tableau où chaque ligne du tableau correspondra à un capteur connecté, reprenant son nom, le type de fluide, sa localisation et sa valeur. L'utilisateur pourra filtrer les données temps réel soit en fonction du type de fluide, soit en fonction du bâtiment dans lequel se trouvent les capteurs.

De plus, si la valeur d'un capteur est en dehors des bornes définies pour ce capteur, la ligne du tableau correspondant à ce capteur sera mise en évidence dans le tableau (par exemple, affichée en gras ou affiché en rouge).

### Panneau d'analyse *a posteriori*

Ce panneau proposera une visualisation des données *a posteriori* : la visualisation des données se fera sous la forme de courbes où le temps sera représenté sur l'axe des abscisses et les valeurs du capteur seront affichées en ordonnée. Il sera possible d'afficher les courbes de 1 à 3 capteurs en simultanées.

Pour pouvoir afficher un graphique, l'utilisateur devra renseigner le type de fluide, puis le ou les capteurs pour lesquels il souhaite visualiser les données. La liste des capteurs sélectionnables sera mise à jour en fonction du fluide sélectionné préalablement. Enfin, l'utilisateur devra déterminer la période de temps (date de début et date de fin) pour laquelle il souhaite afficher ces données.

### *Panneau de gestion des capteurs*

Ce dernier panneau servira à afficher tous les capteurs ayant été connectés au moins une fois à votre application. L'ensemble des capteurs sera représenté sur la gauche de votre panneau sous la forme d'un arbre à 3 niveaux : bâtiment, étage et nom du capteur.

Lors de la sélection d'un capteur dans l'arbre, les informations relatives à ce capteur seront affichées sur la droite du panneau. Les informations à afficher sont le nom du capteur, le type de fluide, les informations sur la localisation du capteur, et la valeur de ses seuils.

Par défaut, chaque capteur aura des valeurs de seuil en fonction du fluide (cf. tableau ci-dessous). Cependant, les valeurs de seuil d'un capteur pourront être modifiées par l'utilisateur.

Fluide	Unité	Seuil min	Seuil max
EAU	m <sup>3</sup>	0	10
ELECTRICITE	kWh	10	500
TEMPERATURE	°C	17	22
AIR COMPRIE	m <sup>3</sup> /h	0	5

## Travail à rendre

### *Dossier de conception*

Dans un premier temps, vous devrez rendre un dossier de conception dans lequel vous présenterez et **justifierez** vos choix de conception. Ce document devra principalement décrire :

- Les différentes interfaces constituant votre application. Vous devrez décrire les composants que vous comptez utiliser et la manière dont ils seront assemblés pour construire votre interface ;
- Le modèle relationnel de votre base de données.
- Le diagramme de classe de votre application ;
- Les diagrammes de séquence expliquant le fonctionnement de votre système pour les principales actions possibles.

Ce rapport est à déposer sur Moodle au plus tard le **10 Décembre 2018**.

### *Exécutable du projet*

Votre projet devra être déposé sur Moodle au plus tard le **15 Janvier 2019**.

Vous devrez mettre à disposition :

- Un fichier README.txt expliquant comment exécuter votre application, et notamment comment créer la base de données nécessaire à la sauvegarde des données,
- Un fichier permettant de générer les tables de votre base de données (un fichier .sql),
- Un exécutable pour pouvoir lancer votre application,
- Le code source de l'ensemble de votre projet.

### *Soutenance de projet*

La soutenance du projet se déroulera **mi Janvier 2019**.

Pour cette soutenance, la présence de chaque membre du groupe est obligatoire.

Aucun support de présentation ne vous est demandé pour cette soutenance.

La soutenance consistera à faire une démonstration de votre application à votre tuteur en lui présentant les différentes fonctionnalités mises en œuvre.