

Simulador Smart Campus

Uma tendencia atual é a implantação de Sistemas Inteligentes, conhecidos como “Smart ” + área de aplicação. Nesse contexto, temos as *Smart Cities* (cidades inteligentes), *Smart Hospitals* (hospitais inteligentes), *Smart Campus* (campus universitário inteligente), *Smart Farms* (fazendas inteligentes) e assim em diversas áreas de aplicação. Uma vez que toda nova tecnologia chega primeiro em Universidades, o conceito de *Smart Campus* é um dos precursores em aplicações nas mais variadas áreas. Essas aplicações podem ser implementadas por meio de conceitos de Internet das Coisas (IoT – *Internet of the Things*), cujos objetos do dia a dia podem estar conectados a internet, fornecendo informações de uso geral ou mesmo sobre o seu comportamento. Essas áreas de “Smart”s e IoT são tradicionalmente trabalhadas (não exclusivamente) em áreas de domínio dos sistemas distribuídos.

Simular ambientes de aplicações reais é uma das técnicas que permitem que problemas não detectados inicialmente e ampliações de funcionalidades possam ser detectados. Os simuladores muitas vezes possuem o objetivo de validar em principais ideias e mostrar que determinados problemas podem ou não serem contornados. Em geral, a simulação precede a implantação física real de sistemas computacionais.

Não seria interessante estudar numa Universidade que contasse com projetos para implementar conceitos de *Smart Campus*? Pois essa é uma tarefa desafiadora? Como ficaram sabendo que vocês são alunos da disciplina de Sistemas Distribuídos, do curso de Ciência da Computação da UPF, sua tarefa é iniciar um simulador de *Smart Campus* para o UPF. O simulador vai se preocupar inicialmente com as características climáticas e de qualidade do ar do Campus Central.

No desenvolvimento do simulador, as seguintes aplicações devem ser elaboradas:

- Simulador de sensores: os sensores de condições climáticas deverão se comunicar via socket TCP enquanto que os de qualidade do ar devem se comunicar com socket UDP;
- Recepção e armazenamento de dados dos sensores: um concentrador de comunicação para recepção dos dados dos sensores e armazenamento no BD;
- Um serviço Web no padrão SOAP: servirá para cadastrar sensores (Sensor e Tipo);
- Um consumidor no padrão SOAP: acesso as funcionalidades do Web Service;
- Um Serviço Web no padrão Rest: fornecer acesso as informações dos sensores;
- Um consumidor no padrão Rest: consultar e visualizar as informações dos sensores.

Um banco de dados (BD) deve ser implementado para armazenar as informações coletadas pelos sensores e ser a fonte das informações consultadas pelos usuários. O BD pode ser implementado em qualquer tecnologia. Lembre-se que seu trabalho é de Sistemas Distribuídos, dessa forma o BD é um acessório (porém necessário). O BD deve conter pelo menos as seguintes tabelas.

Sensor

#codigo: Integer (chave primária)

Descricao: Varchar (30)

Latitude: Float

Longitude: Float

Tipo@: Integer (chave estrangeira para TipoSensor)

Dados

#codigo@: Integer (chave estrangeira para sensor)

#seq: Integer

DataHora: DateTime

Valor: Float

TipoSensor

#codigo: Integer (chave primária)

Descricao: Varchar (30)

Comunica: Char (‘U’ → UDP / ‘T’ TCP)

Mínimo: Integer (menor valor gerado)

Máximo: Integer (maior valor gerado)

Intervalo: Integer (intervalo em segundos de amostragem)

O trabalho será desenvolvido pelo seu grupo, usando software de desenvolvimento colaborativo e com controle de versões tais como Git (preferencial) ou bitbucket, entre outros. O professor da disciplina de Sistemas Distribuídos deve fazer parte do grupo de desenvolvimento.

A aplicação que simula os sensores deve ser escrita na linguagem Java. Deve-se gerar um valor aleatório, entre os valores máximo e mínimo para o tipo do sensor. A cada intervalo de amostragem, um valor deve ser gerado e enviado.

A aplicação que recebe os valores de sensores (deve ser apenas uma), deve ser escrita na linguagem Java. Deve-se receber os valores dos simuladores de sensores e armazenar no BD. Cada sensor de comunicação TCP deve ter uma *thread* específica para receber seus dados. Os sensores de comunicação UDP devem receber os dados na mesma *thread*.

O serviço Web, no padrão SOAP serve para acessar os dados do BD relativos aos sensores. Deve ser possível incluir, alterar, excluir ou consultar as tabelas Sensor e TipoSensor. O serviço pode ser implementado em qualquer linguagem de programação.

A aplicação que acessa o serviço Web no padrão SOAP deve ter uma interface gráfica (não Web), que proporciona ao usuário atualizar os dados das tabelas Sensor e Tipo Sensor. A aplicação pode ser implementada em qualquer linguagem de programação.

O serviço web, no padrão Rest serve para acessar os dados coletados pelos sensores, proporcionando que outras aplicações possam trazer benefícios aos usuários do Campus. Este serviço deve oferecer métodos GET e pelo menos um POST. A implementação desse serviço pode ser realizada em qualquer linguagem de programação.

Deve-se criar um consumidor do serviço Web no padrão Rest. Esse consumidor ficará responsável atualizar os usuários das informações coletadas pelos sensores. O consumidor deve estar numa aplicação com acesso pela Web, facilitando que todos os usuários do Campus possam obter informações coletadas. Gráficos e tabelas na visualização dos dados são bem vindas. A escolha da tecnologia para a implementação dessa aplicação é de livre escolha do grupo. Uma relação de funcionalidades dessa aplicação é:

- Verificar o último valor amostrado por cada um dos sensores;
- Escolher um sensor e visualizar os dados coletados num período (horas e dias);
- Visualizar as médias diárias dos dados dos sensores;
- Visualizar as médias de cada característica climática ou elemento de qualidade do ar (vários sensores) dos sensores no campus por dia.

A relação de funcionalidades descritas até aqui equivalem 70% da nota do trabalho. Os demais 30% serão computados se o grupo tiver funcionalidades extra (10% cada uma), como por exemplo:

- detector de falhas nos sensores;
- um *dashboard* que facilite a visualização dos dados coletados;
- uma página, com integração junto ao GoogleMaps para visualização em mapa da posição dos sensores no Campus;
- geração de um índice de qualidade do ar, a partir dos dados coletados. Referenciar a fonte dos parâmetros utilizados;
- uma aplicação móvel para fornecer aos usuários do Campus a visualização dos dados coletados pelos sensores;
- atualização aplicação web a cada novo dados coletado (via notificação);
- outros ... (consultar professor)

Apresentação

O líder da equipe deverá realizar uma apresentação técnica: mostrar os modelos usados, as aplicações desenvolvidas e as tecnologias usadas (não usar código-fonte) do que foi construído (10 minutos). Em seguida, deve-se proceder a demonstração das implementações. Uma máquina ficará com o servidor/serviços enquanto outra ficará com os clientes/consumidores/aplicativos desenvolvidos. Serão reservados dois projetores para a demonstração das implementações (40 min). A apresentação será no LCI e caso algum grupo necessite de acesso externo a algum serviço bloqueado, deve prever antecipadamente e solicitar ao professor para liberar o acesso. Em seguida, cada integrante será entrevistado individualmente para informar o que implementou e como foi realizada a implementação.

Serão realizadas duas avaliações: uma referente ao desenvolvimento do grupo e outra referente ao desenvolvimento individual de cada componente do grupo.

Grupo A	Grupo B
Victor	Diego
Fernando	Vitor
Herbert	Thiago
Leonardo	Tiago
Lucas	Willian
Willian	