

**TRABALHO DE DESENVOLVIMENTO DE UM PROJETO DE SISTEMAS DISTRIBUÍDOS**

**- 2019/2 -**

**Descrição:**

O trabalho (em grupos de DUAS ou TRÊS pessoas) consiste em desenvolver uma solução prática, que utilize comunicação de dados por troca de mensagens via Socket e processamento com Threads.

O trabalho deve apresentar:

- Comunicação por troca de mensagens UDP/TCP utilizando Sockets ou via WebServices;
- Ao menos uma classe comunicante desenvolvida na linguagem de programação Java;
- Todos os servidores deverão executar Threads para gerenciamento dos serviços disponibilizados;
- Documentação de código;
- Poderá ser desenvolvido via localhost, mas a apresentação será no laboratório distribuindo os serviços em diferentes máquinas.

**Definição das duplas/trio: 15 de outubro de 2019.**

A configuração dos grupos ficou:

- G1. Daniel Cunha / Lucas Cocco
- G2. Ricardo Fachinello / Tiago Seben
- G3. Cassiano Stefani / Juliano Oliveira / Matheus Cainelli

Os temas de desenvolvimento dos trabalhos focam na comunicação de dados entre diferentes sistemas e tipos de sistemas mantendo, de maneira segura, os dados comunicados. Para a realização do trabalho dividiu-se em três sub-áreas:

**A1. Robótica**

Através da comunicação de dados distribuído, um equipamento eletrônico deverá ser controlado de maneira remota.

**A2. Mundo Financeiro**

Através da comunicação de dados distribuído, gráficos, dados analíticos e notícias serão apresentados ao usuário.

**A3. Comunicação de dados com APIs de terceiros**

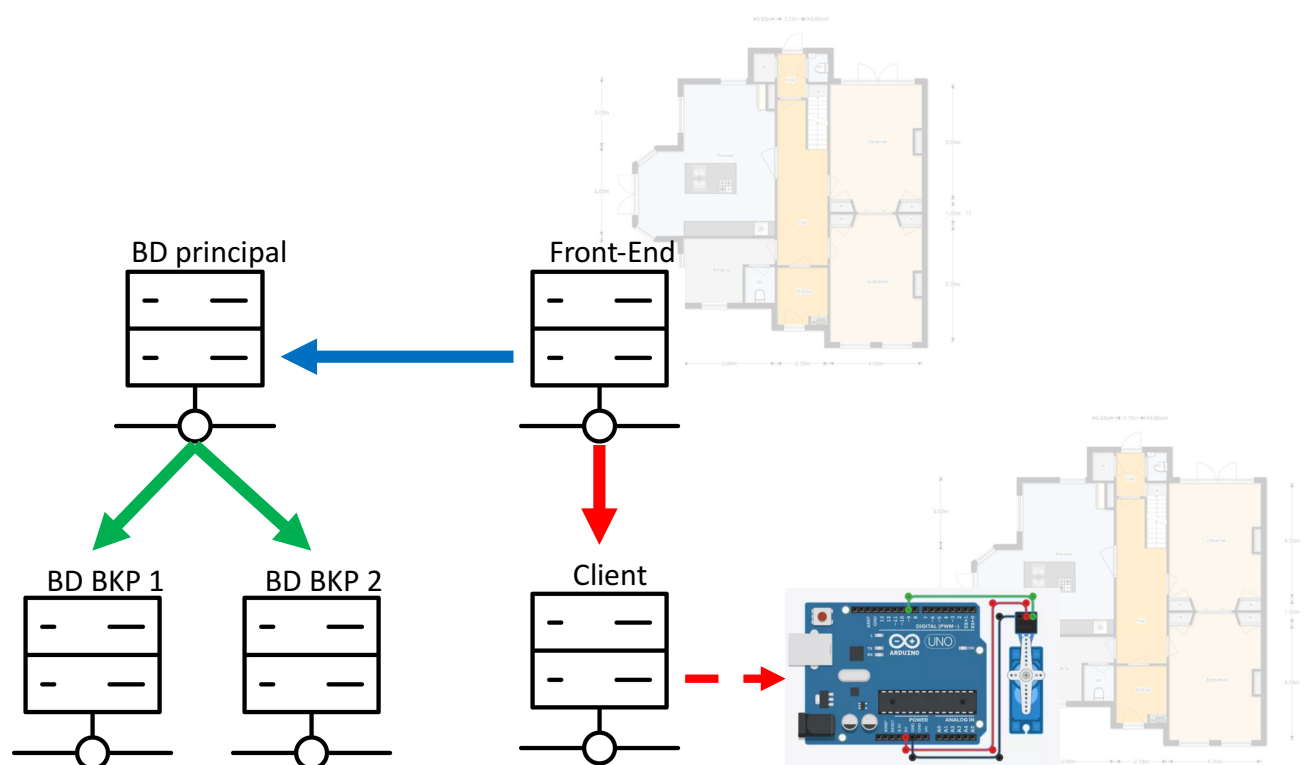
Através da comunicação de dados distribuído, serão apresentadas formas de consumo de dados de sistemas terceiros.

**Data de Apresentação da Primeira Versão: 12 de novembro de 2019**

O professor vai solicitar mudanças necessárias para o atendimento dos requisitos da aplicação.

**Data de Apresentação Final: 19 de novembro de 2019.**

## A1. Robótica



### Descrição:

A ideia deste trabalho é controlar um produto de automação residencial. Utilizando um microcontrolador (Arduino/RaspberryPi/ESP32) e LEDs (para simular lâmpadas) será possível controlar a iluminação da casa de forma remota.

**Grupo que irá desenvolver [G2] → Ricardo Fachinello / Tiago Seben**

### Detalhes:

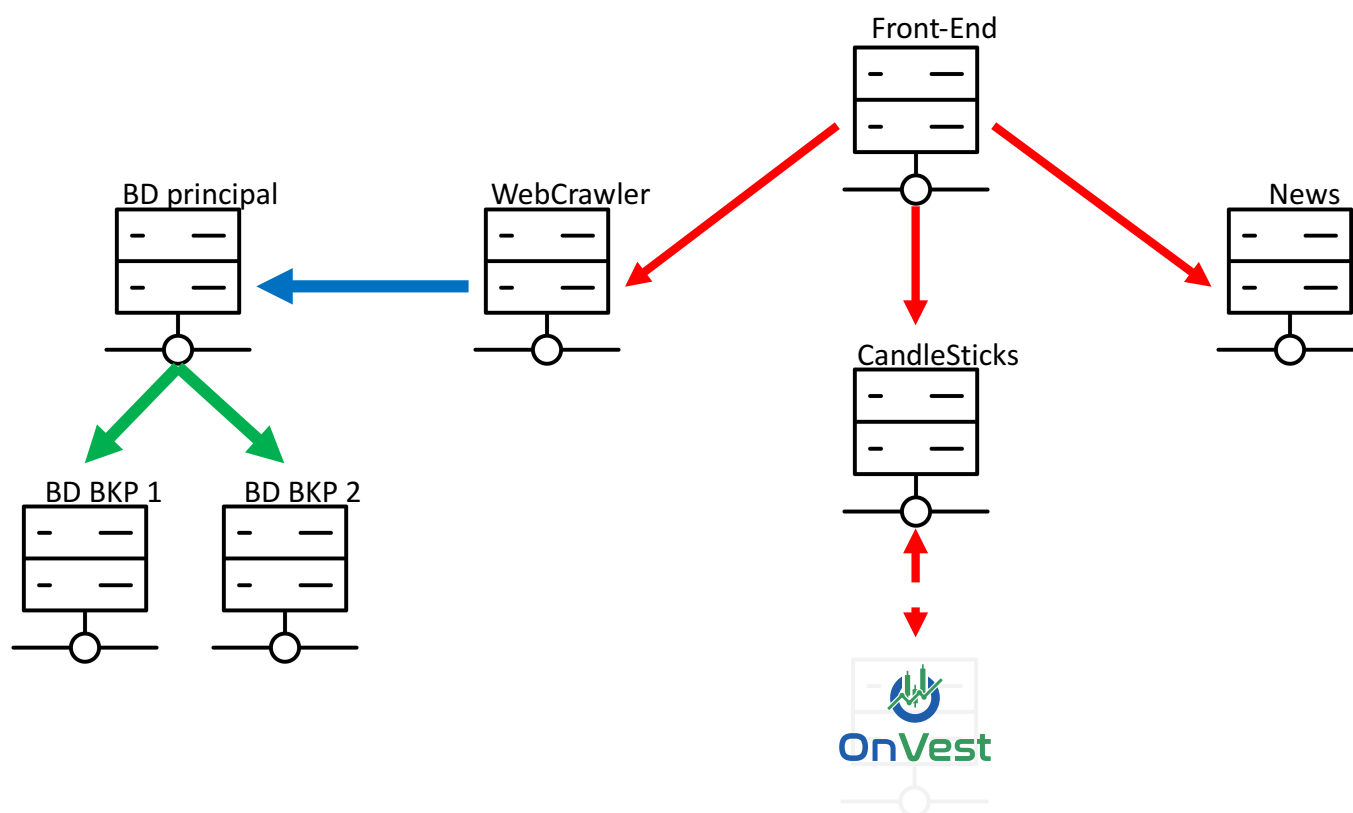
A máquina “Front-end” é responsável por disponibilizar uma camada de visualização para o usuário. Nela deverá ser apresentada uma planta-baixa de uma casa. O usuário irá clicar no cômodo que gostaria de ligar/desligar as luzes. A planta-baixa deverá ser montada com divisões e cores por cômodos. Poderá ser utilizada uma imagem para isso, mas receberão nota extra caso integrem a ferramenta FloorPlanner ou desenvolvam a planta-baixa em CSS. O front deve apresentar os cômodos informando se está com luzes ligadas ou não.

Toda escolha do usuário deverá ser registrada em uma base de dados (outra máquina), para manter o log de comandos com timestamp. Cada registro no log deve ser replicado em outras duas máquinas com o objetivo de manter backup de dados.

A máquina front-end irá se comunicar com outra máquina (Client). Esta máquina client irá fazer a comunicação com o microcontrolador para atuar na “casa”.

A casa deverá ser apresentada em uma maquete (um isopor com a planta desenhada/colada). Os LEDs deverão estar dispostos no isopor para quando o usuário ligar/desligar as luzes do(s) cômodo(s) podermos simular o ambiente.

## A2. Mundo Financeiro



### Descrição:

A ideia deste trabalho entregar ao usuário uma camada de análise de dados de empresas do mundo financeiro. O usuário terá a possibilidade de fazer 3 análises: gráfica, fundamentalista e de notícias.

**Grupo que irá desenvolver [G1] → Daniel Cunha / Lucas Cocco**

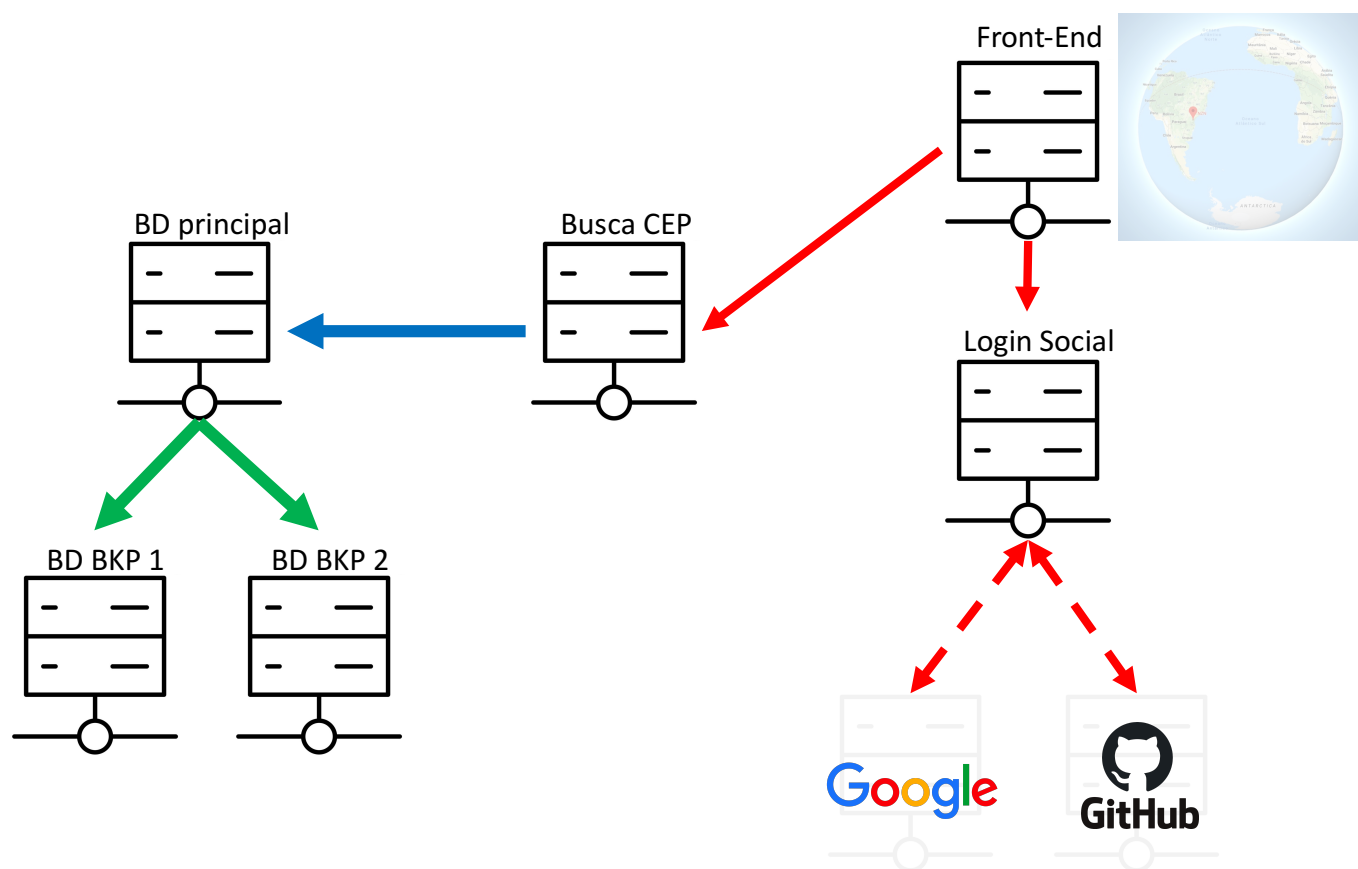
### Detalhes:

A máquina "Front-end" é responsável por disponibilizar uma camada de visualização para o usuário. Nela deverá ser apresentada a possibilidade de o usuário escolher o que deseja analisar: gráfico, fundamentos ou notícias. Com base na escolha do usuário o sistema distribuído irá agir. Quando o usuário escolher a visualização de forma gráfica o front-end irá se comunicar com a máquina "CandleSticks". Esta máquina tem como trabalho buscar dados de uma empresa (escolhida pelo usuário no front-end) no site da OnVest (ex: [www.onvest.com.br/grafico/PETR4](http://www.onvest.com.br/grafico/PETR4)) e enviar para que o front-end possa montar o gráfico candle-stick para o usuário analisar. O gráfico deve ser um candle stick, podendo ser qualquer gráfico disponível para uso na internet, como exemplo: TradingView (API fechada) e o ApexCharts.js.

Se o usuário escolher "News", o front-end irá se comunicar com a máquina News. Esta máquina tem como objetivo retornar ao front-end um JSON com as 10 últimas notícias sobre a empresa que o usuário quer analisar. Para esta tarefa pode ser utilizado o Google News.

Caso o usuário escolha a análise fundamentalista, o front-end irá se comunicar com o WebCrawler. Esta máquina irá buscar os dados da empresa em análise no site do Fundamentus. A máquina irá gravar em uma base de dados os dados retornados no site (<http://www.fundamentus.com.br/detalhes.php?papel=petr4>). Os dados que devem ser armazenados são: Papel, LPA, VPA, ROE, ROIC, e Valor do Mercado. Toda vez que o usuário solicitar acesso aos dados fundamentalistas o sistema deverá registrar os dados em uma base de dados (outra máquina). Cada registro na base de dados deverá ser replicado em outras duas máquinas com o objetivo de manter backup de dados.

### A3. API de Terceiros



#### Descrição:

A ideia deste trabalho é fazer uma comunicação e consumo de dados de ambientes de terceiros.

**Grupo que irá desenvolver [G3]** → Cassiano Stefani / Juliano Oliveira / Matheus Cainelli

#### Detalhes:

A máquina "Front-end" é responsável pelo controle de login social e pela apresentação de um endereço no mapa. O Login Social deve, obrigatoriamente, ter ao menos 2 opções, tais como: Facebook, Google, LinkedIn, GitHub, Pinterest, Dropbox, Instagram, etc.

Quando o usuário estiver logado ele deverá ter a possibilidade de visualizar uma rota entre dois pontos. Para isso, ele deverá informar 2 CEPs. Ambos os CEPs deverão ser consultados na API dos correios para captura do endereço. Toda busca de CEP deverá ser registrada em uma base de dados (outra máquina). Cada registro na base de dados deve ser replicado em outras duas máquinas com o objetivo de manter backup de dados.

Os dois endereços retornados da busca do CEP deverão ser enviados para a máquina front-end, que irá apresentar o Google Maps com a rota entre os dois endereços.

### **Sobre a Avaliação do Trabalho**

A avaliação do trabalho será dividida em duas partes: grupo e individual.

A avaliação em grupo se dará da seguinte forma:

- **Front-End**  
Será analisada a forma de apresentação dos dados e interação do usuário com o sistema. O front-end deverá ser feito de forma gráfica e amigável para o uso.
- **Base de dados com backup**  
Será avaliado o backup de cada instância de dados.
- **Interação com terceiros (Robô, OnVest, Correios, Arduino, etc)**  
Serão avaliadas as comunicações efetuadas ao longo do uso da aplicação.
- **Linguagens de programação**  
Obrigatoriamente deve-se utilizar Java em, ao menos, uma classe da aplicação. O uso correto da linguagem correta para cada etapa será levado em consideração, ex: CSS, JS, HTML para Front-End; Python para análise de dados; Java, PHP, JS para back-end; etc.
- **Forma de apresentação do fluxo**  
Será avaliada a forma como um “log” de execução será apresentado em cada máquina do sistema, informando início de comunicação, conteúdo e fim.
- **Apresentação do trabalho como um todo**  
Uso do sistema e completude nas funcionalidades.

A avaliação individual se dará da seguinte forma:

- Após a apresentação dos 3 grupos para a turma toda, apenas 1 indivíduo de cada grupo irá apresentar o que fez. 1 indivíduo de cada grupo deverá ser destinado a coordenador, para informar o que cada um fez.
- Os trabalhos deverão ser armazenados no GitHub contendo todos os componentes do grupo como colaborador, juntamente com o professor (usuário guimadalozzo). A avaliação individual também levará em consideração os commits/pushes realizados pelo integrante do grupo.