

## **Sistema administrativo e marcação de pontos de funcionários**

O sistema desenvolvido em questão atribui o sistema administrativo e de pontos de funcionários cadastrados no sistema com interface gráfica. Há três projetos, onde um é servidor, e outros dois clientes, onde um é cliente ponto e outro é o cliente administrativo.

Principais funcionalidades do servidor:

- Salvamento de dados: Os dados enviados para o servidor são salvos em um bloco de notas, então mesmo se fechar o servidor, os dados são salvos a cada mudança em bloco de notas, para não precisar inserir novamente informações. Informações dos pontos estão também nesse bloco de notas, juntamente com as datas e horas de entrada e saída.
- Leitura de dados: Quando abrimos o servidor, ele automaticamente carrega os dados do bloco de notas.
- Criação de um bloco de notas: Caso não tenha um bloco de notas com as informações, o próprio servidor cria um bloco de notas na pasta do projeto, onde todas as informações do servidor e dos clientes são guardadas.
- Agrupamento de dados: Possui todos os dados requisitados pelos clientes.

Principais funcionalidades do cliente ponto:

- Marcar ponto: para marcar entrada ou saída de um funcionário, basta colocar uma ID válida no campo selecionado, se não for válida, o sistema mostra uma tela de erro para digitar apenas uma ID válida. Se for válida, ele dirá se é entrada ou saída do funcionário, dependendo do estado anterior de entrada do ID.

Principais funcionalidades do cliente administrativo:

- Comunicação com servidor: se dois ou mais programas administrativos forem abertos simultaneamente, todos se comunicam com o servidor e o atualiza direto, mesmo que os clientes não sejam atualizados, eles irão avisar de uma possível modificação ou remoção com avisos das suas respectivas funções.
- Listar funcionários cadastrados: Para isso, basta clicar em atualizar, e aparecerá uma lista com todos funcionários e algumas informações úteis de cada.
- Adicionar funcionário: é obrigatório a inserção de dados como nome, endereço, CEP, número da casa, forma de receber (em mãos, via depósito ou via correios). O sistema atribui uma ID automaticamente em sequência quando for confirmada a criação de um funcionário.

- Remover funcionário: para remoção de um funcionário, basta apenas digitar uma ID válida na lista e clicar em remover. Caso queira remover mais de um ao mesmo tempo, basta digitar as ID's separadas com um espaço em branco.
- Modificar funcionário: para modificar, o sistema atualiza uma JComboBox com todos funcionários do sistema até aquele momento, e basta modificar algum campo e clicar em modificar.

No programa, poderia ser implementado o sistema financeiro, para controlar os salários, folhas de pagamentos, controle geral dos pontos, exportar folhas de pagamentos, etc., assim o sistema ficaria de certa forma completo para uma empresa.

As principais dificuldades no desenvolvimento do sistema foram apenas na criação e manipulação dos sockets, pois se trocasse uma linha onde fosse chamada uma conexão com o servidor, pararia todo o servidor, pois ele estaria em espera, demorou algum tempo até perceber isto, pois os erros davam em outras funções ou até mesmo nos servidores clientes, quando o erro era no servidor. Já na questão do sistema em si, não foram detectados tantos problemas, pois o programa já tinha sido desenvolvido anteriormente por mim, e somente readaptado para trabalhar em rede.

## **Instruções para rodar o programa**

Para rodar os programas é simples, basta rodar os programas na ordem:

1. Servidor → Package Servidor → Inicio\_Servidor.
  - 1.1. Cliente Administrativo → Package Geral → Inicio\_Cliente.
  - 1.2. Cliente Ponto → Package Geral → Inicio\_Cliente.

Nota1: Pode rodar em qualquer ordem os pontos 1.1 e 1.2, mas para manipular os pontos, é necessário ter funcionários cadastrados para os ID's serem válidos.

Nota2: O servidor cria um arquivo .txt na pasta do projeto, chamado Funcionarios, que é onde guarda os dados do sistema. Para trocar a pasta, basta ir em:

1. Servidor → Package Servidor → Inicio\_Servidor.
  - 1.1. Na classe Inicio\_Servidor, existe uma public static String endereco, com uma string vazia, que direciona o arquivo diretamente para a pasta do projeto. Para mudar a pasta, basta mudar a string, como por exemplo: endereco = "C:\\Users\\USUÁRIOAQUI\\Desktop\\"; Este endereço salva ou cria (se necessário) o arquivo .txt na área de trabalho do USUÁRIOAQUI.