

Projeto de Arquivos

Gabriela Moreira

Cliente

- Buscar → “fOrderID,RowID” (f3,1)
- Inserir → “iCSV” (i1;3;13/10/2010...)
- Remover → “rOrderID,RowID” (r3,1)
- Salvar → “s”

Manipulação de dados (strings)

```

dado criaDado(string linhaRegistros){

    dado busca;

    try {
        if(linhaRegistros.find(';') > linhaRegistros.size()){
            busca.OrderID = stoi(linhaRegistros.substr(0,linhaRegistros.find(',')));
            busca.RowID = stoi(linhaRegistros.substr(linhaRegistros.find(',')+1,linhaRegistros.size()));
            busca.tamanho = -1;
        } else {
            int primeiro = linhaRegistros.find(';');
            busca.RowID = stoi(linhaRegistros.substr(0,primeiro));
            busca.OrderID = stoi(linhaRegistros.substr(primeiro+1,linhaRegistros.find('; ',primeiro+1)));
            busca.tamanho = linhaRegistros.size();
        }
    } catch (exception& ex) {
        cout << ">stoi exception: " << busca.OrderID << " " << busca.RowID << endl;
    }

    return busca;
}

```

Servidor – Busca

```
int encontraPosicao(dado busca, map <dado,int> arvore){

    if(arvore.find(busca) == arvore.end())
        return -1;

    dado encontrado = arvore.find(busca)->first;
    int posicao = arvore.find(busca)->second;

    arvore.erase(encontrado);
    if(arvore.count(busca) == 0){
        arvore[encontrado] = posicao;
        return posicao;
    }

    arvore[encontrado] = posicao;
    return 0;

}
```

```
posicao = encontraPosicao(criaDado(comando), arvore);

if(posicao == -1)
    new_sock << "404";

else if(posicao > 0){
    vector <string> registros = criaRegistro(pegaRegistro(posicao));

    string json = "{";
    for(int i=0;i<registros.size();i++)
        json += ("{" + campos[i] + ": " + registros[i] + "},");
    json[json.size()-1] = '}';

    new_sock << "S" + json;
} else
    new_sock << "RowIDNecessaria";
```

Servidor – Inserção

```
void escreveRegistro (string linhaRegistros){  
  
    ofstream saida ("Sample - Superstore Sales.csv", ios::app);  
    if(saida.is_open())  
        saida << linhaRegistros << endl;  
    saida.close();  
  
}
```

```
map <dado,int> insereIndice (string linhaRegistros, map <dado,int>  
arvore){  
  
    dado novoDado;  
    novoDado = criaDado(linhaRegistros);  
    arvore[novoDado] = arvore.size();  
  
    return arvore;  
  
}
```

Servidor - Remoção

- Verifica se o registro existe e é único - chama a função de busca;
- `arvore.erase(criaDado(registro));`

Servidor – Salvar dados para arquivos

```
map<dado,int>::iterator it;  
vector < vector<string> > colunas;  
  
dadoDoRegistro = criaDado(linhaRegistros);  
it = arvore.find(dadoDoRegistro);  
if(it != arvore.end()){  
    indices << dadoDoRegistro.OrderID << " " << dadoDoRegistro.RowID << " " << dadoDoRegistro.tamanho << " " << it->second << endl;  
    colunas.push_back(criaRegistro(linhaRegistros));  
}
```

Carregando a árvore

```
map <dado,int> leIndices (){

    dado indiceTemporario;
    int posicao;
    map <dado,int> arvore;
    ifstream arquivoIndices ("indices.txt");
    if (arquivoIndices.is_open()){
        while (! arquivoIndices.eof() ){
            arquivoIndices >> indiceTemporario.OrderID >> indiceTemporario.RowID >> indiceTemporario.tamanho >> posicao;
            arvore[indiceTemporario] = posicao;
            tamanhos.push_back(indiceTemporario.tamanho);
        }
        arquivoIndices.close();
    }

    return arvore;
}
```


Buscando um registro no arquivo

```
string pegaRegistro (int indice){  
  
    ifstream arquivoRegistros ("Sample - Superstore Sales.csv");  
    int posicao = 0;  
    for(int i=0;i<indice;i++)  
        posicao += tamanhos[i] + 1;  
  
    arquivoRegistros.seekg (posicao,ios::cur);  
    string linhaRegistros;  
    getline (arquivoRegistros,linhaRegistros);  
  
    return linhaRegistros;  
}
```

Busca com ID composto – Sobrecarga

```
bool operator< (dado d1, dado d2){
    if(d1.OrderID != d2.OrderID || d1.RowID == -1 || d2.RowID == -1)
        return (d1.OrderID < d2.OrderID);
    else
        return (d1.RowID < d2.RowID);
}

bool operator== (dado d1, dado d2){
    if(d1.RowID == -1 || d2.RowID == -1)
        return (d1.OrderID == d2.OrderID);
    else
        return (d1.OrderID == d2.OrderID && d1.RowID == d2.RowID);
}
```