PUC Minas – Coração Eucarístico
Turma 91.30.100
Algoritmos e Estruturas de Dados III
Professor responsável
Walisson Ferreira de Carvalho

Trabalho Prático de Algoritmos e Estruturas de Dados III

Parte III

Gabriel da Silva Cassino

Matrícula 796535

Índice

Considerações iniciais sobre as Partes I , II e III

O programa desde etapa I é o mesmo, por esta razão foi mantido o diagrama da etapa I e II. Dada a necessidade de detalhamento foi adicionado um Índice a este protótipo de documentação.

Considerações iniciais sobre a Etapa III

Foi desenvolvido a compactação e descompactação em modo demostrativo, com enfoque principalmente no LZW, e por conseguinte Huffman.

A aplicação recebe um caminho predefinido e executa a compactação, e logo após a descompactação dando feedback em milissegundos ou segundos dependendo do algoritmo em execução. A compactação se mostrou eficiente permitindo no LZ78 a execução da etapa I.
A parte III possui execução A e B (Processo A e Processo B).

A Processo A executa Huffman em 2 modos e LZW, já Processo B apenas LZ78 sem opção de escolha apenas demostrativo. Ambos possuem um modo de demonstração de arquivo pequeno autogerado pelo código.

Introdução e Resumo Parte I e II

O programa desenvolvido em Java por meio de duas classe principais auxiliadas por outras classes complementares realiza a leitura de um arquivo .csv tanto internamente, quanto em qualquer pasta do computador em uso e a princípio, é compatível com Windows e Linux.

Ao final da leitura, gera um arquivo de dados .db que pode passar pelas operações de:
-Criar um registro ou vários registros em lote(arquivo .csv);
-Leitura de um registro ou de todos os registros;
-Atualização de um registro;
-Exclusão lógica de um registro.

Fonte da base de dados usada

Para a criação do arquivo de dados foi utilizada a base de dados sobre acidentes de trânsito proveniente do site https://www.kaggle.com/datasets/oktayrdeki/traffic-accidents contendo 209306 registros. Tal escolha de uma base tão vasta tem por objetivo aprimorar técnicas e desenvolver novas.

A parte I foi mantida em estagio1 e a parte II está subdividida em: estagio2, pasta Indexed e subpastas, e pasta Test_<abcd>

Diagrama do Programa

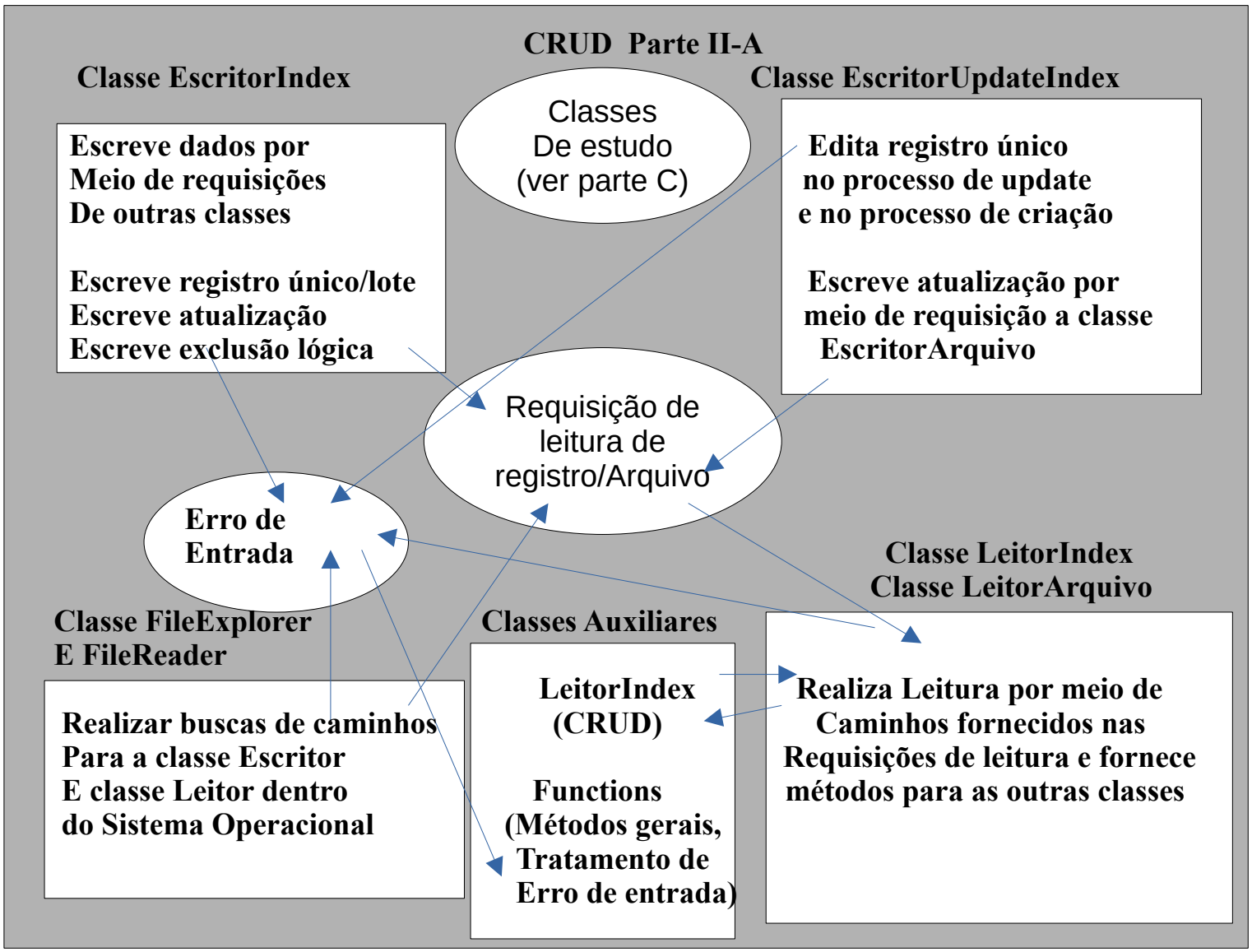


Diagrama Parte II – Programa, Arquivos, Sistema Operacional e Dispositivo - A

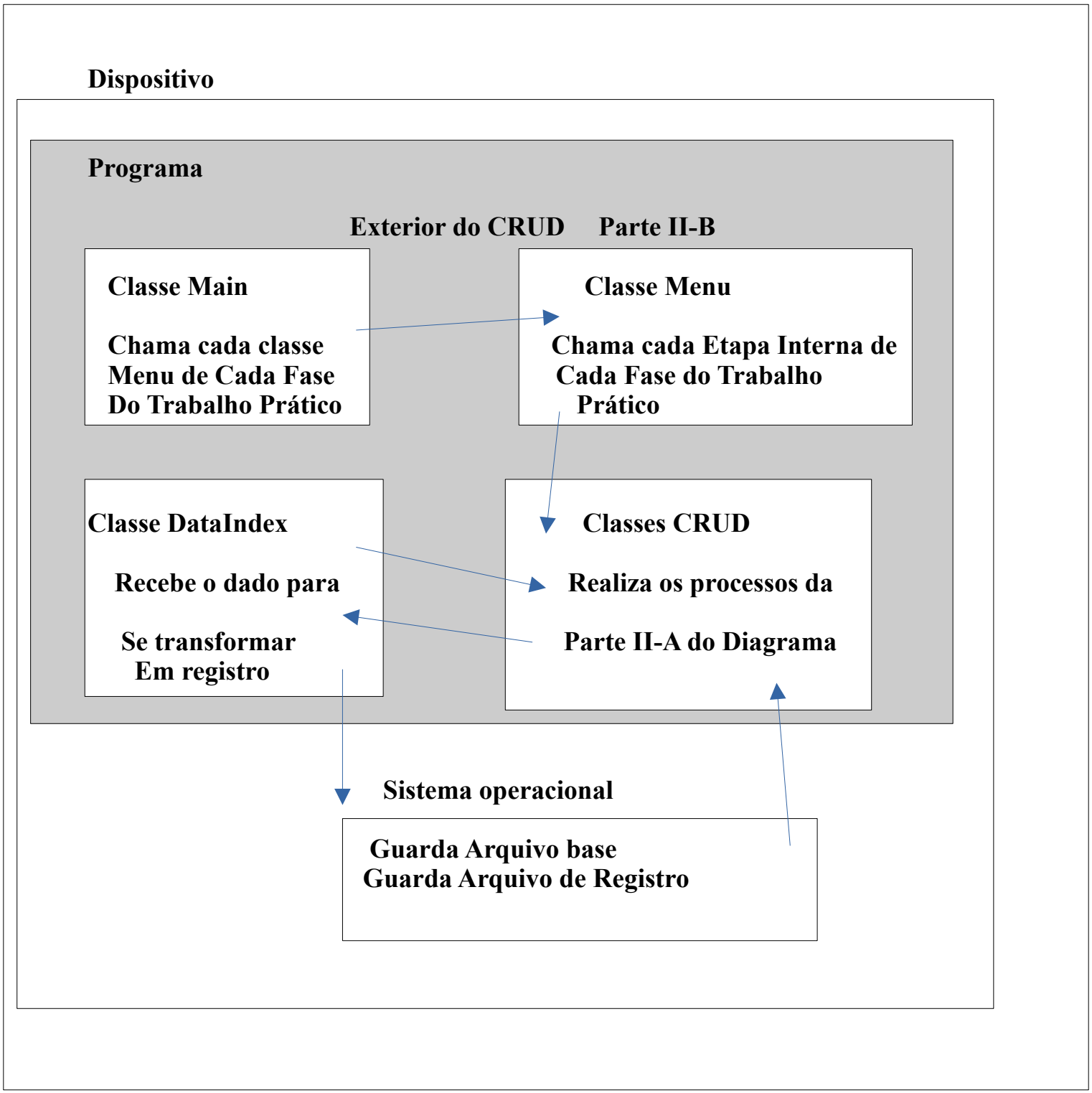
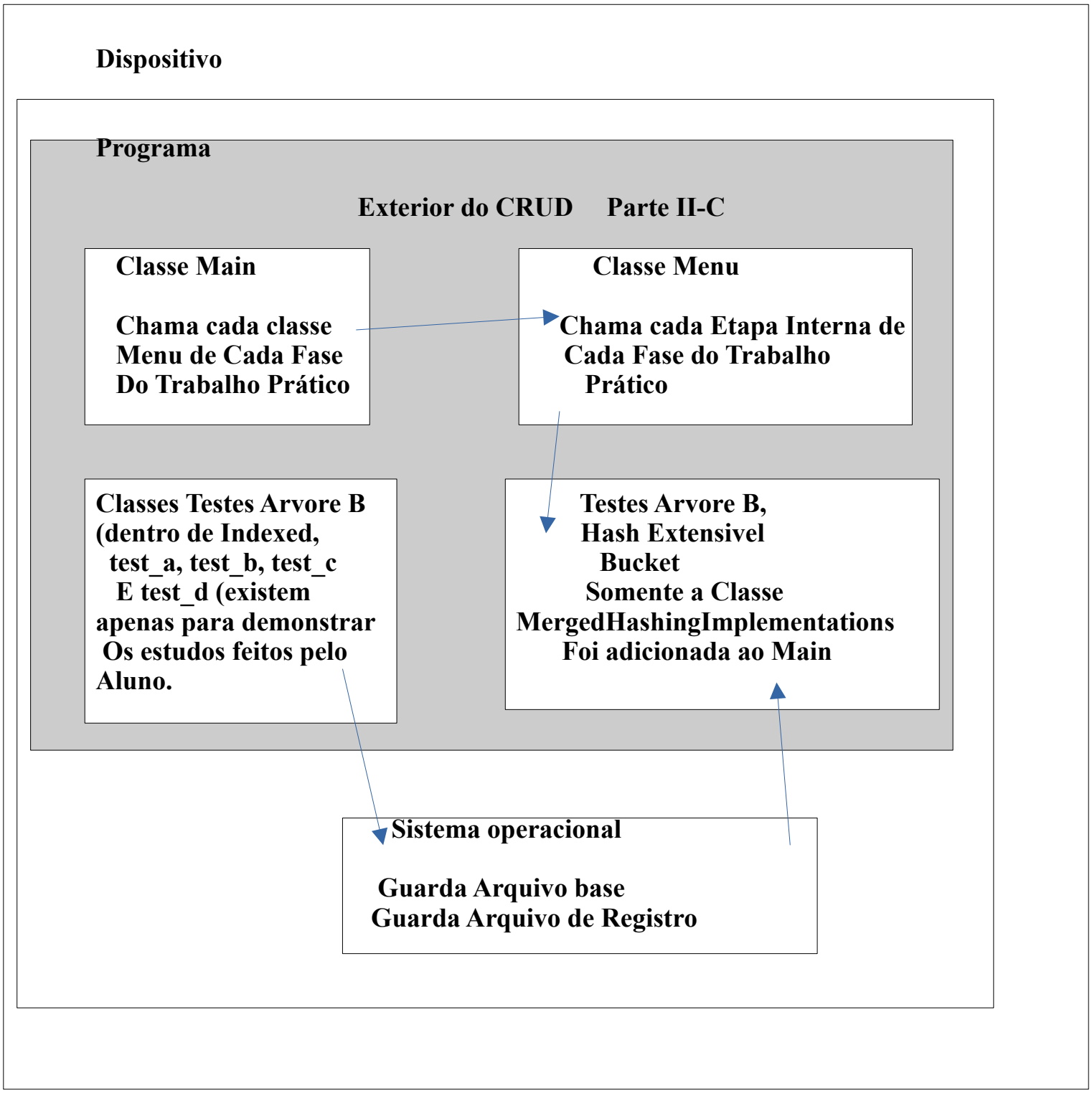


Diagrama Parte II – Programa, Arquivos, Sistema Operacional e Dispositivo - B

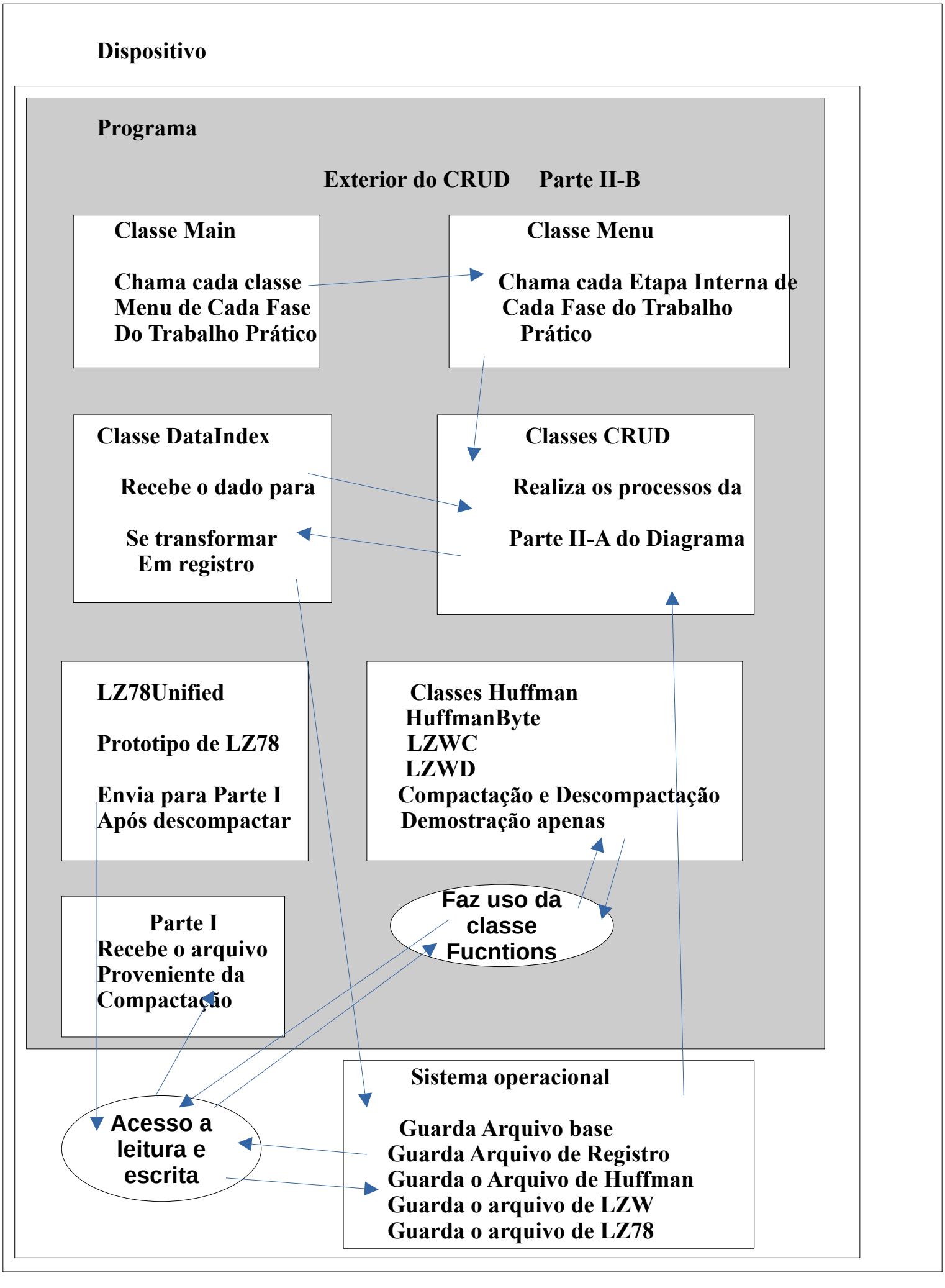


O que foi possível implementar da Ementa da Parte II:

- Arquivo Sequencial com Indice Externo a ele com a Lógica de CRUD Implementada, porém não completamente testada;

Há arquivos de tentativa de:
- Arvore B de Indice;
- Arvore de Lista;
- Teste de Bucket e Hash Extensivel para Indexar Arquivo, sendo este o único que possui métodos de criação e inserção funcional sem ocorrer Java.lang.StackOverflowError ou não inserção por Java.io.EOFException.

Diagrama Parte III – Programa, Arquivos, Sistema Operacional e Dispositivo



Detalhes Parte III

Classes de Demonstração

Classe Huffman: Executa a compressão e descompressão baseada no arquivo CSV;
Classe HuffmanByte: Executa também a compressão e descompressão baseada no arquivo CSV, mas pode enviar o arquivo .db a qualquer parte que o usa;
Classe LZWC: Executa apenas a compressão baseada no arquivo CSV, porém pode chamar a Classe LZWD descompressão para gerar o arquivo CSV, e pode enviar o arquivo .db a qualquer parte que o usa;
Classe LZWD: Apenas executa a descompressão para gerar o arquivo CSV, e pode enviar o arquivo .db a qualquer parte que o usa.

Classe efetiva

LZWUnified:Executa a compressão e descompressão baseada no arquivo .db ou .csv diretamente, podendo enviar o arquivo .db a qualquer parte que o usa(no momento envia a Parte I baseado em um arquivo predefinido).

Arquivo Sample

Todas as classes gerarão um arquivo de teste de eficiência como demonstração. Verificou-se por hora que qualquer das classes recuperam o arquivo porém a compactação só é eficiente quando o arquivo é grande(5% a 65% do arquivo original), quando é pequeno ultrapassa o dobro(104% a 450% do arquivo original), e no LZW pode haver acréscimo de até 4% após descompactação.