DOCUMENTAÇÃO VERCIONADO

**Mario Cesar da Nóbrega – 2211250**

**Gabriela Pereira – 2211288**

**Lucas Tourinho Mamede -221133**

**Guilherme Abade -2211337**

Versão VERSÃO FINAL

# Índice dos Componentes

## Lista de Classes

Aqui estão as classes, estruturas, uniões e interfaces e suas respectivas descrições:

**Fila**  pagenum

**Node**  pagenum

**Snapshot**  pagenum

# Índice dos Arquivos

## Lista de Arquivos

Esta é a lista de todos os arquivos e suas respectivas descrições:

**C:/Users/mario/OneDrive/Documentos/Documentação Vercionador/VERCIONADO.C**  pagenum

# Classes

## Referência da Estrutura Fila

### Atributos Públicos

* **Node** \* **inicio**
* **Node** \* **fim**

### Atributos

#### Node\* Fila::fim

#### Node\* Fila::inicio

#### A documentação para essa estrutura foi gerada a partir do seguinte arquivo:

* C:/Users/mario/OneDrive/Documentos/Documentação Vercionador/**VERCIONADO.C**

## Referência da Estrutura Node

### Atributos Públicos

* char **arquivo** [256]
* struct **Node** \* **prox**

### Atributos

#### char Node::arquivo[256]

#### struct Node\* Node::prox

#### A documentação para essa estrutura foi gerada a partir do seguinte arquivo:

* C:/Users/mario/OneDrive/Documentos/Documentação Vercionador/**VERCIONADO.C**

## Referência da Estrutura Snapshot

### Atributos Públicos

* int **identificador**
* **Fila** \* **arquivos**
* struct **Snapshot** \* **prox**

### Atributos

#### Fila\* Snapshot::arquivos

#### int Snapshot::identificador

#### struct Snapshot\* Snapshot::prox

#### A documentação para essa estrutura foi gerada a partir do seguinte arquivo:

* C:/Users/mario/OneDrive/Documentos/Documentação Vercionador/**VERCIONADO.C**

# Arquivos

## Referência do Arquivo C:/Users/mario/OneDrive/Documentos/Documentação Vercionador/VERCIONADO.C

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/stat.h>

### Componentes

* struct **Node**struct **Fila**
* struct **Snapshot**

### Definições de Tipos

* typedef struct **Node** **Node**

### Funções

* **Fila** \* **criarFila** ()
* void **enfileirar** (**Fila** \*fila, const char \*arquivo)
* char \* **desenfileirar** (**Fila** \*fila)
* void **iniciar** ()
* void **adicionar\_arquivo** (const char \*arquivo)
* void **criarSnapshot** (**Fila** \*arquivosMarcados)
* void **listarSnapshots** (int exibirConteudo)
* void **mostrarSnapshot** (int identificador)
* void **mudarSnapshot** (int identificador)
* void **mudarAtual** ()
* int **main** (int argc, char \*argv[])

### Variáveis

* **Snapshot** \* **inicioSnapshot** = NULL
* **Snapshot** \* **fimSnapshot** = NULL
* int **proximoIdentificador** = 1

### Definições dos tipos

#### typedef struct Node Node

### Funções

#### void adicionar\_arquivo (const char \* *arquivo*)

84 {

85 char arquivoConteudo[256];

86 snprintf(arquivoConteudo, sizeof(arquivoConteudo), ".versionador/conteudo/%s", arquivo);

87

88 FILE\* arquivoOriginal = fopen(arquivo, "r");

89 if (arquivoOriginal == NULL) {

90 printf("Erro ao abrir o arquivo %s.\n", arquivo);

91 return;

92 }

93

94 FILE\* arquivoConteudoNovo = fopen(arquivoConteudo, "w");

95 if (arquivoConteudoNovo == NULL) {

96 printf("Erro ao criar o arquivo %s.\n", arquivoConteudo);

97 return;

98 }

99

100 char linha[256];

101 while (fgets(linha, sizeof(linha), arquivoOriginal) != NULL) {

102 fprintf(arquivoConteudoNovo, "%s", linha);

103 }

104

105 fclose(arquivoOriginal);

106 fclose(arquivoConteudoNovo);

107

108 char arquivoVersao[256];

109 snprintf(arquivoVersao, sizeof(arquivoVersao), ".versionador/versoes/%s", arquivo);

110

111 FILE\* arquivoVersaoNovo = fopen(arquivoVersao, "w");

112 if (arquivoVersaoNovo == NULL) {

113 printf("Erro ao criar o arquivo %s.\n", arquivoVersao);

114 return;

115 }

116

117 fprintf(arquivoVersaoNovo, "%d\n", proximoIdentificador);

118 fprintf(arquivoVersaoNovo, "%s\n", arquivo);

119

120 fclose(arquivoVersaoNovo);

121 }

#### Fila \* criarFila ()

26 {

27 Fila\* fila = (Fila\*)malloc(sizeof(Fila));

28 fila->inicio = NULL;

29 fila->fim = NULL;

30 return fila;

31 }

#### void criarSnapshot (Fila \* *arquivosMarcados*)

123 {

124 if (arquivosMarcados->inicio == NULL) {

125 printf("Nenhum arquivo marcado para snapshot.\n");

126 return;

127 }

128

129 char arquivo[256];

130 snprintf(arquivo, sizeof(arquivo), ".versionador/versoes/%d", proximoIdentificador);

131

132 FILE\* arquivoSnapshot = fopen(arquivo, "w");

133 if (arquivoSnapshot == NULL) {

134 printf("Erro ao criar o arquivo de snapshot.\n");

135 return;

136 }

137

138 fprintf(arquivoSnapshot, "%d\n", proximoIdentificador);

139

140 Node\* arquivoAtual = arquivosMarcados->inicio;

141 while (arquivoAtual != NULL) {

142 fprintf(arquivoSnapshot, "%s\n", arquivoAtual->arquivo);

143 arquivoAtual = arquivoAtual->prox;

144 }

145

146 fclose(arquivoSnapshot);

147

148 printf("Snapshot %d criado com sucesso.\n", proximoIdentificador);

149

150 Snapshot\* novoSnapshot = (Snapshot\*)malloc(sizeof(Snapshot));

151 novoSnapshot->identificador = proximoIdentificador;

152 novoSnapshot->arquivos = arquivosMarcados;

153 novoSnapshot->prox = NULL;

154

155 if (inicioSnapshot == NULL) {

156 inicioSnapshot = novoSnapshot;

157 fimSnapshot = novoSnapshot;

158 } else {

159 fimSnapshot->prox = novoSnapshot;

160 fimSnapshot = novoSnapshot;

161 }

162

163 proximoIdentificador++;

164 }

#### char \* desenfileirar (Fila \* *fila*)

47 {

48 if (fila->inicio == NULL) {

49 return NULL;

50 }

51

52 Node\* noRemovido = fila->inicio;

53 fila->inicio = fila->inicio->prox;

54 char\* arquivo = strdup(noRemovido->arquivo);

55 free(noRemovido);

56 return arquivo;

57 }

#### void enfileirar (Fila \* *fila*, const char \* *arquivo*)

33 {

34 Node\* novoNo = (Node\*)malloc(sizeof(Node));

35 strcpy(novoNo->arquivo, arquivo);

36 novoNo->prox = NULL;

37

38 if (fila->inicio == NULL) {

39 fila->inicio = novoNo;

40 fila->fim = novoNo;

41 } else {

42 fila->fim->prox = novoNo;

43 fila->fim = novoNo;

44 }

45 }

#### void iniciar ()

59 {

60 struct stat st;

61

62 // Verificar se o diretório .versionador já existe

63 if (stat(".versionador", &st) == 0 && S\_ISDIR(st.st\_mode)) {

64 return; // O diretório já existe, não é necessário criar novamente

65 }

66

67 if (mkdir(".versionador") != 0) {

68 printf("Erro ao criar o diretório .versionador.\n");

69 exit(1);

70 }

71

72 if (mkdir(".versionador/versoes") != 0) {

73 printf("Erro ao criar o diretório de versões.\n");

74 exit(1);

75 }

76

77 if(mkdir(".versionador/conteudo") != 0) {

78 printf("Erro ao criar o diretório de conteúdo.\n");

79 exit(1);

80 }

81 }

#### void listarSnapshots (int *exibirConteudo*)

166 {

167 Snapshot\* snapshotAtual = inicioSnapshot;

168

169 while (snapshotAtual != NULL) {

170 printf("Snapshot %d:\n", snapshotAtual->identificador);

171

172 if (exibirConteudo) {

173 Fila\* arquivos = snapshotAtual->arquivos;

174 Node\* arquivoAtual = arquivos->inicio;

175

176 while (arquivoAtual != NULL) {

177 char arquivo[256];

178 snprintf(arquivo, sizeof(arquivo), ".versionador/versoes/%s", arquivoAtual->arquivo);

179

180 FILE\* arquivoSnapshot = fopen(arquivo, "r");

181 if (arquivoSnapshot == NULL) {

182 printf("Erro ao abrir o arquivo %s.\n", arquivoAtual->arquivo);

183 arquivoAtual = arquivoAtual->prox;

184 continue;

185 }

186

187 printf("Conteúdo do arquivo %s:\n", arquivoAtual->arquivo);

188

189 char linha[256];

190 while (fgets(linha, sizeof(linha), arquivoSnapshot) != NULL) {

191 printf("%s", linha);

192 }

193

194 printf("\n");

195

196 fclose(arquivoSnapshot);

197 arquivoAtual = arquivoAtual->prox;

198 }

199 }

200

201 snapshotAtual = snapshotAtual->prox;

202 }

203 }

#### int main (int *argc*, char \* *argv*[])

343 {

344 if (argc < 2) {

345 printf("Uso: %s <comando> [argumentos]\n", argv[0]);

346 return 1;

347 }

348

349 if (strcmp(argv[1], "iniciar") == 0) {

350 iniciar();

351 } else if (strcmp(argv[1], "adiciona") == 0) {

352 if (argc < 3) {

353 printf("Uso: %s adiciona <arquivo> [arquivo2] ...\n", argv[0]);

354 return 1;

355 }

356

357 Fila\* arquivosMarcados = criarFila();

358

359 for (int i = 2; i < argc; i++) {

360 enfileirar(arquivosMarcados, argv[i]);

361 }

362

363 criarSnapshot(arquivosMarcados);

364 } else if (strcmp(argv[1], "registra") == 0) {

365 if (argc < 3) {

366 printf("Uso: %s registra <mensagem>\n", argv[0]);

367 return 1;

368 }

369

370 Fila\* arquivosMarcados = criarFila();

371

372 for (int i = 2; i < argc; i++) {

373 enfileirar(arquivosMarcados, argv[i]);

374 }

375

376 criarSnapshot(arquivosMarcados);

377 } else if (strcmp(argv[1], "log") == 0) {

378 if (argc > 2 && strcmp(argv[2], "--conteudo") == 0) {

379 listarSnapshots(1);

380 } else {

381 listarSnapshots(0);

382 }

383 } else if (strcmp(argv[1], "mostrar") == 0) {

384 if (argc < 3) {

385 printf("Uso: %s mostrar <identificador>\n", argv[0]);

386 return 1;

387 }

388

389 int identificador = atoi(argv[2]);

390 mostrarSnapshot(identificador);

391 } else if (strcmp(argv[1], "mudar") == 0) {

392 if (argc < 3) {

393 printf("Uso: %s mudar <identificador>\n", argv[0]);

394 return 1;

395 }

396

397 int identificador = atoi(argv[2]);

398 mudarSnapshot(identificador);

399 } else if (strcmp(argv[1], "mudar") == 0) {

400 mudarAtual();

401 } else {

402 printf("Comando %s não encontrado.\n", argv[1]);

403 return 1;

404 }

405 return 0;

406 }

#### void mostrarSnapshot (int *identificador*)

205 {

206 Snapshot\* snapshotAtual = inicioSnapshot;

207

208 while (snapshotAtual != NULL) {

209 if (snapshotAtual->identificador == identificador) {

210 break;

211 }

212

213 snapshotAtual = snapshotAtual->prox;

214 }

215

216 if (snapshotAtual == NULL) {

217 printf("Snapshot %d não encontrado.\n", identificador);

218 return;

219 }

220

221 printf("Snapshot %d:\n", snapshotAtual->identificador);

222

223 Fila\* arquivos = snapshotAtual->arquivos;

224 Node\* arquivoAtual = arquivos->inicio;

225

226 while (arquivoAtual != NULL) {

227 char arquivo[256];

228 snprintf(arquivo, sizeof(arquivo), ".versionador/versoes/%s", arquivoAtual->arquivo);

229

230 FILE\* arquivoSnapshot = fopen(arquivo, "r");

231 if (arquivoSnapshot == NULL) {

232 printf("Erro ao abrir o arquivo %s.\n", arquivoAtual->arquivo);

233 arquivoAtual = arquivoAtual->prox;

234 continue;

235 }

236

237 printf("Conteúdo do arquivo %s:\n", arquivoAtual->arquivo);

238

239 char linha[256];

240 while (fgets(linha, sizeof(linha), arquivoSnapshot) != NULL) {

241 printf("%s", linha);

242 }

243

244 printf("\n");

245

246 fclose(arquivoSnapshot);

247 arquivoAtual = arquivoAtual->prox;

248 }

249 }

#### void mudarAtual ()

297 {

298 Snapshot\* snapshotAtual = inicioSnapshot;

299

300 while (snapshotAtual != NULL) {

301 if (snapshotAtual->identificador == proximoIdentificador - 1) {

302 break;

303 }

304

305 snapshotAtual = snapshotAtual->prox;

306 }

307

308 if (snapshotAtual == NULL) {

309 printf("Snapshot %d não encontrado.\n", proximoIdentificador - 1);

310 return;

311 }

312

313 printf("Snapshot %d:\n", snapshotAtual->identificador);

314

315 Fila\* arquivos = snapshotAtual->arquivos;

316 Node\* arquivoAtual = arquivos->inicio;

317

318 while (arquivoAtual != NULL) {

319 char arquivo[256];

320 snprintf(arquivo, sizeof(arquivo), ".versionador/versoes/%s", arquivoAtual->arquivo);

321

322 FILE\* arquivoSnapshot = fopen(arquivo, "r");

323 if (arquivoSnapshot == NULL) {

324 printf("Erro ao abrir o arquivo %s.\n", arquivoAtual->arquivo);

325 arquivoAtual = arquivoAtual->prox;

326 continue;

327 }

328

329 printf("Conteúdo do arquivo %s:\n", arquivoAtual->arquivo);

330

331 char linha[256];

332 while (fgets(linha, sizeof(linha), arquivoSnapshot) != NULL) {

333 printf("%s", linha);

334 }

335

336 printf("\n");

337

338 fclose(arquivoSnapshot);

339 arquivoAtual = arquivoAtual->prox;

340 }

341 }

#### void mudarSnapshot (int *identificador*)

251 {

252 Snapshot\* snapshotAtual = inicioSnapshot;

253

254 while (snapshotAtual != NULL) {

255 if (snapshotAtual->identificador == identificador) {

256 break;

257 }

258

259 snapshotAtual = snapshotAtual->prox;

260 }

261

262 if (snapshotAtual == NULL) {

263 printf("Snapshot %d não encontrado.\n", identificador);

264 return;

265 }

266

267 printf("Snapshot %d:\n", snapshotAtual->identificador);

268

269 Fila\* arquivos = snapshotAtual->arquivos;

270 Node\* arquivoAtual = arquivos->inicio;

271

272 while (arquivoAtual != NULL) {

273 char arquivo[256];

274 snprintf(arquivo, sizeof(arquivo), ".versionador/versoes/%s", arquivoAtual->arquivo);

275

276 FILE\* arquivoSnapshot = fopen(arquivo, "r");

277 if (arquivoSnapshot == NULL) {

278 printf("Erro ao abrir o arquivo %s.\n", arquivoAtual->arquivo);

279 arquivoAtual = arquivoAtual->prox;

280 continue;

281 }

282

283 printf("Conteúdo do arquivo %s:\n", arquivoAtual->arquivo);

284

285 char linha[256];

286 while (fgets(linha, sizeof(linha), arquivoSnapshot) != NULL) {

287 printf("%s", linha);

288 }

289

290 printf("\n");

291

292 fclose(arquivoSnapshot);

293 arquivoAtual = arquivoAtual->prox;

294 }

295 }

### Variáveis

#### Snapshot\* fimSnapshot = NULL

#### Snapshot\* inicioSnapshot = NULL

#### int proximoIdentificador = 1