

May 24, 20 1:20

Cocinero.cpp

Page 1/2

```

1  //
2  // Created by rozanecm on 5/12/20.
3  //
4
5  #include <iostream>
6  #include <random>
7  #include "Cocinero.h"
8
9  Cocinero::Cocinero() {
10     this->fl.l_type = F_RDLCK;
11     this->fl.l_whence = SEEK_SET;
12     this->fl.l_start = 0;
13     this->fl.l_len = 0;
14
15     canal_envio_pedidos_especialista_MM.abrir();
16 }
17
18 Cocinero::~Cocinero() {
19     canal_envio_pedidos_especialista_MM.cerrar();
20     canal_recepcion_de_mm->cerrar();
21     // eliminacion en void EspecialistaMasaMadre::cerrar_canales_particulares();
22 }
23
24 void Cocinero::ejercer_tarea() {
25     bool seguimos_recibiendo = true;
26     while (sigint_handler.getGracefulQuit() == 0 and seguimos_recibiendo) {
27         recibir_pedido(&seguimos_recibiendo);
28         if (seguimos_recibiendo) {
29             realizar_pedido();
30             entregar_pedido_a_repartidor();
31         }
32     }
33     mandar_msj_debug(id + " termino con todos sus deberes.");
34 }
35
36 void Cocinero::recibir_pedido(bool *seguir_recibiendo_pedidos) {
37     ssize_t bytesLeidos = 0;
38     this->id_pedido_actual = "";
39
40     fcntl(canal_recepcionista->get_fd(), F_SETLK, &(this->fl));
41
42     bytesLeidos += canal_recepcionista->leer(static_cast<void *>
43                                             (buffer_recepcionista),
44                                             FIFO_PEDIDOS_BUFFSIZE);
45
46     fcntl(canal_recepcionista->get_fd(), F_SETLK, &(this->fl));
47
48     if (bytesLeidos == 0) {
49         *seguir_recibiendo_pedidos = false;
50     } else {
51         this->id_pedido_actual = buffer_recepcionista;
52         mandar_msj_debug(id + " Recibi pedido: " +
53                         this->id_pedido_actual);
54     }
55 }
56
57 void Cocinero::realizar_pedido() {
58     /* Inicializacion random */
59     std::random_device rd;
60     std::mt19937 gen(rd());
61     std::uniform_int_distribution<> dis(1, 6);
62
63     pedir_racion_mm();
64     esperar_envio_mm();
65     mandar_msj_debug(id + " realizando pedido " + id_pedido_actual);
66     sleep(dis(gen));
67     mandar_msj_debug(id + " termine de realizar el pedido " +
68                     id_pedido_actual);
69 }
70
71 void Cocinero::pedir_racion_mm() {
72     mandar_msj_debug(id + " pidiendo racion de masa madre...");
73     mandar_msj_fifo(id, &this->canal_envio_pedidos_especialista_MM);

```

May 24, 20 1:20

Cocinero.cpp

Page 2/2

```

74 }
75
76 void Cocinero::esperar_envio_mm() {
77     abrir_canal_recepcion_de_mm();
78     std::string mensaje;
79     canal_recepcion_de_mm->leer(static_cast<void *>(buffer_recepcion_mm),
80                                LENGTH_MSJ_ENVIO_MM);
81     mensaje = buffer_recepcion_mm;
82     mensaje.resize(LENGTH_MSJ_ENVIO_MM);
83     if (mensaje == MENSAJE_PARA_ENVIAR_MM_A_COCINEROS) {
84         mandar_msj_debug(id + " recibi racion de masa madre! "
85                         "Continuemos con el pedido...");
86     } else {
87         mandar_msj_debug("ALERTA! " + id + " Por alguna razon no recibi"
88                         " racion masa madre y puedo seguir igual...");
89         std::cout << "msj recibido en lugar de mm: " << mensaje << std::endl;
90         mandar_msj_debug("msj recibido en lu gar de mm: " +
91                         std::to_string(int(buffer_recepcion_mm[1])));
92     }
93 }
94
95 void Cocinero::entregar_pedido_a_repartidor() {
96     mandar_msj_debug(id + " entregando pedido " + id_pedido_actual +
97                     " a repartidor.");
98 }
99
100 void Cocinero::abrir_canal_recepcion_de_mm() {
101     if (not canal_recepcion_de_mm_fue_abierto) {
102         canal_recepcion_de_mm = std::make_unique<FifoLectura>(id);
103         canal_recepcion_de_mm->abrir();
104         canal_recepcion_de_mm_fue_abierto = true;
105     }
106 }

```