ConcuBread

Presentación del trabajo realizado

Alumno: *Rozanec, Matías (97404)*

Profesor: *Deymonaz, Pablo*

7559 - Técnicas de programación concurrente I
Facultad de Ingeniería - Universidad de Buenos Aires

Diagrama de clases

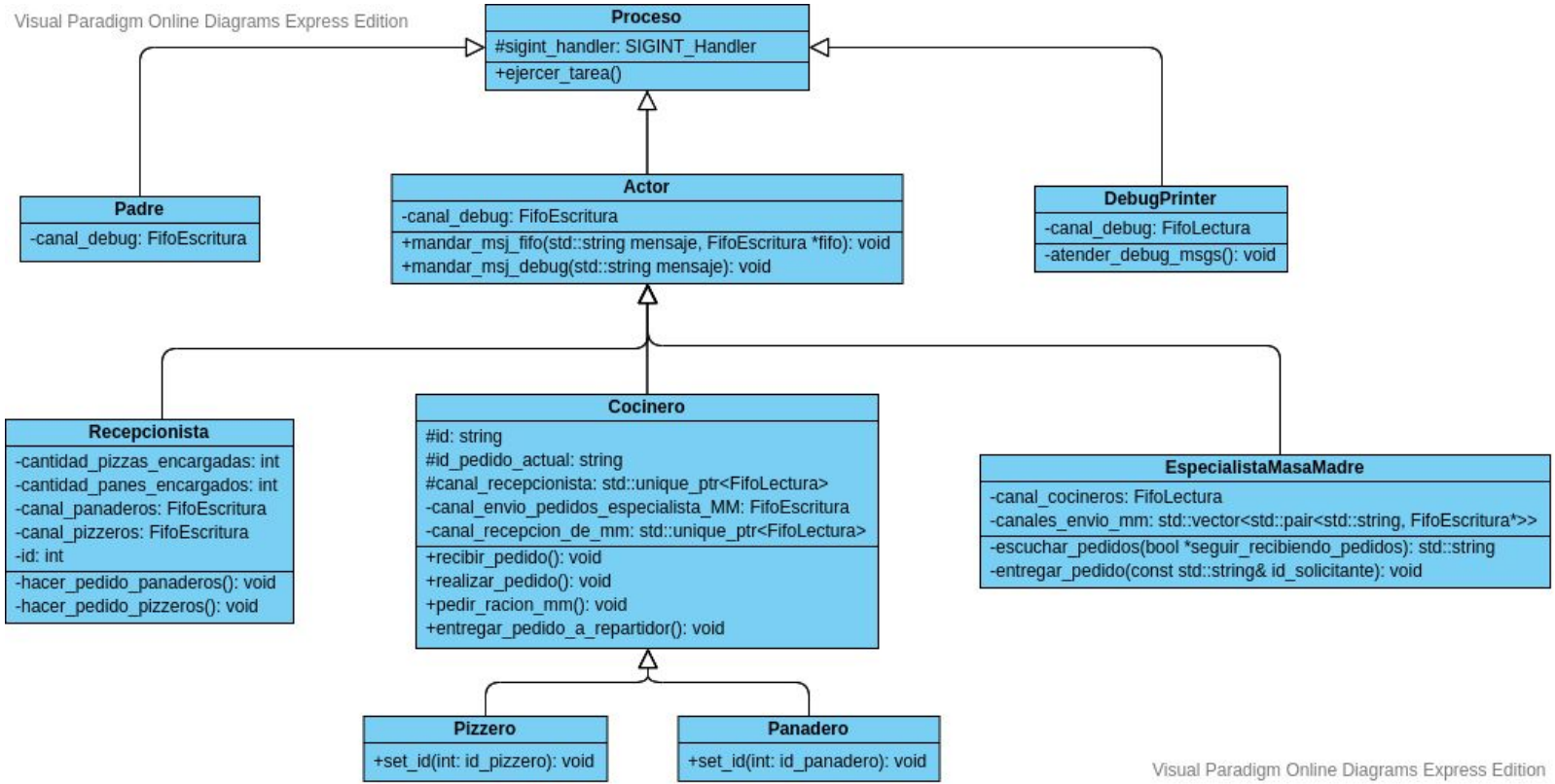


Diagrama de clases

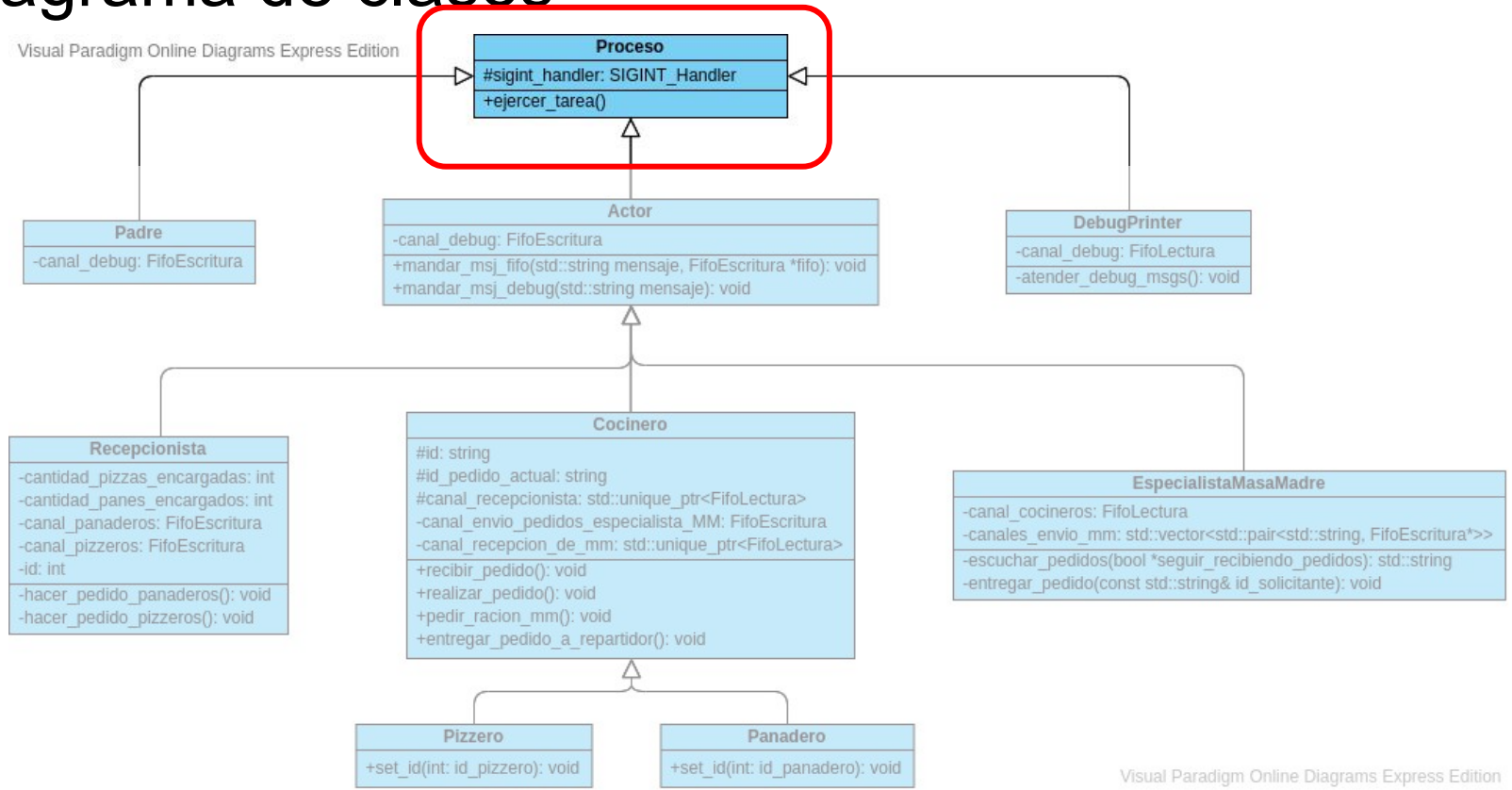
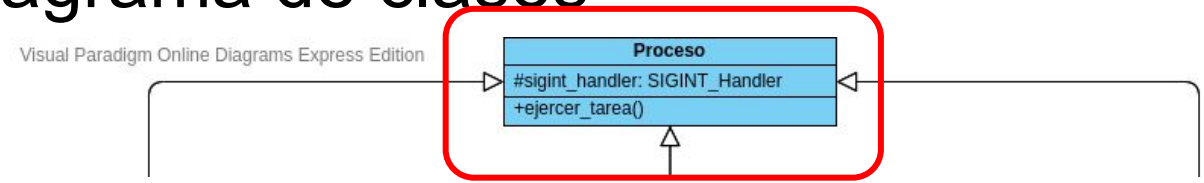


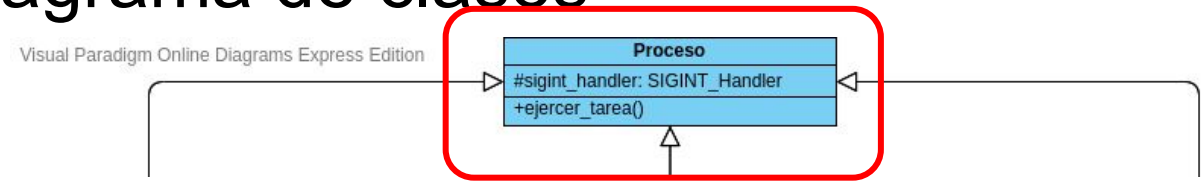
Diagrama de clases



Impacto en el main	
Antes	Después

`switch(crear_procesos()){`
`  case Especialista: {...},`
`  case Panadero: {...},`
`  ...`
`}`
	`proceso = crear_procesos();`
	`proceso.ejercer_tarea();`

Diagrama de clases

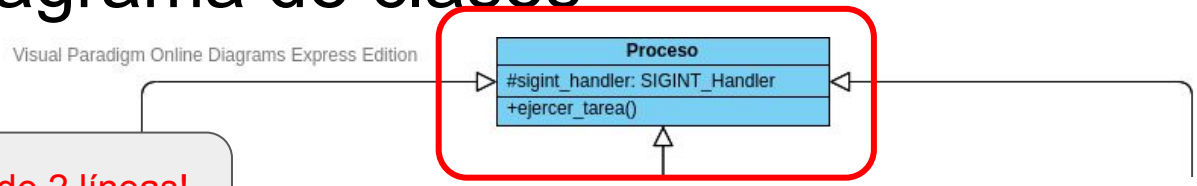


Impacto en el main	
Antes	Después

`switch(crear_procesos()){`
`  case Especialista: {...},`
`  case Panadero: {...},`
`  ...`
`}`
	`proceso = crear_procesos();`
	`proceso.ejercer_tarea();`

Main de 2 líneas!
(casi)

Diagrama de clases



Main de 2 líneas!
(casi)

```
int main(int argc, char* argv[]) {
    int cant_panaderos, cant_pizzeros, cant_recepcionistas;
    leer_config_file(&cant_panaderos, &cant_pizzeros, &cant_recepcionistas);

    std::cout << "Bienvenido a ConcuBread!" << std::endl;
    std::cout << "Simulando..." << std::endl;

    const std::unique_ptr<Proceso> &proceso_generado = ProcessManager::crear_procesos(cant_panaderos, cant_pizzeros,
                                                                                       cant_recepcionistas,
                                                                                       print_debug_msgs(argc, argv));

    proceso_generado->ejercer_tarea();
    return 0;
}
```

Config y saludos

Diagrama de clases

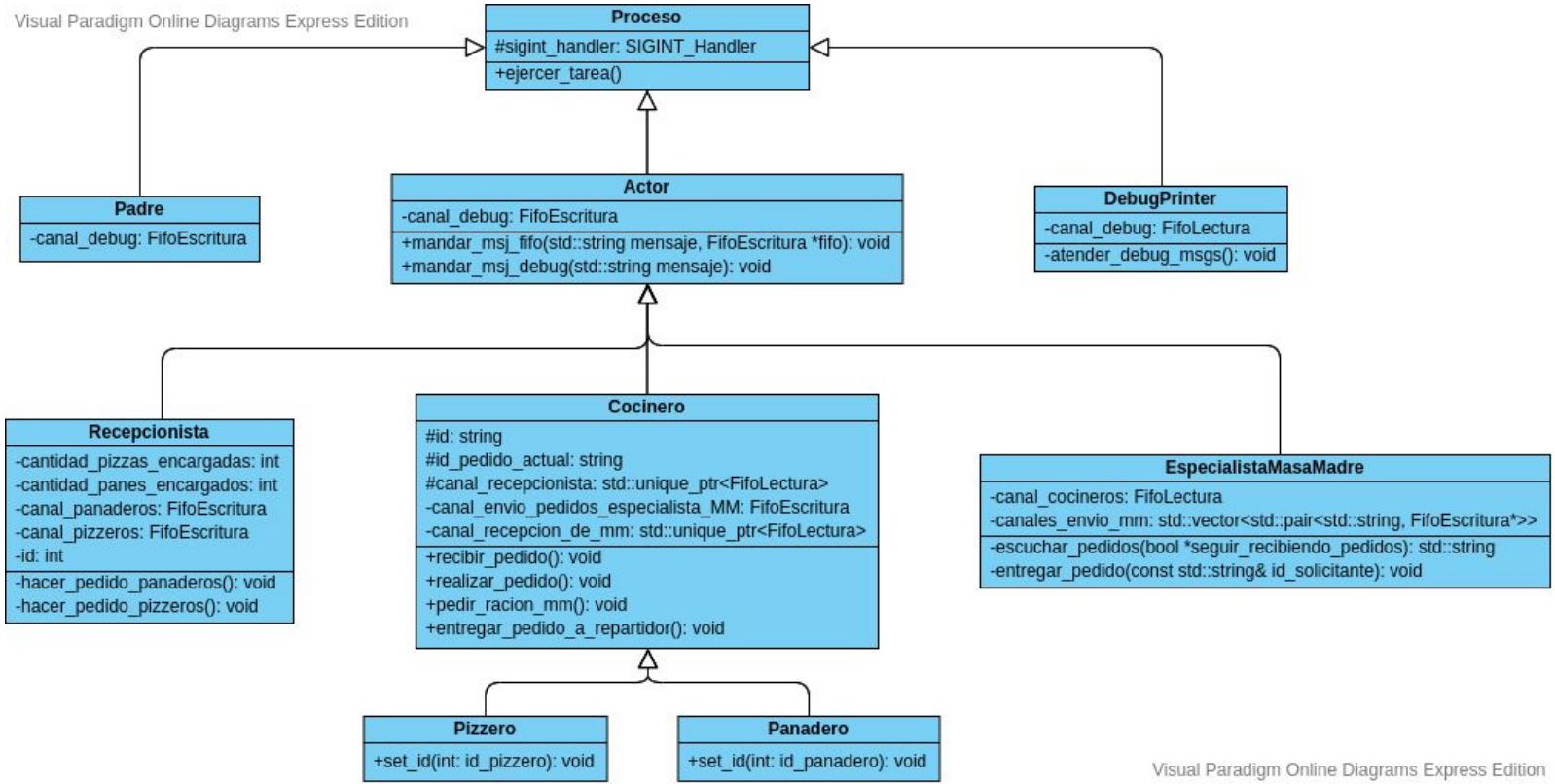


Diagrama de clases

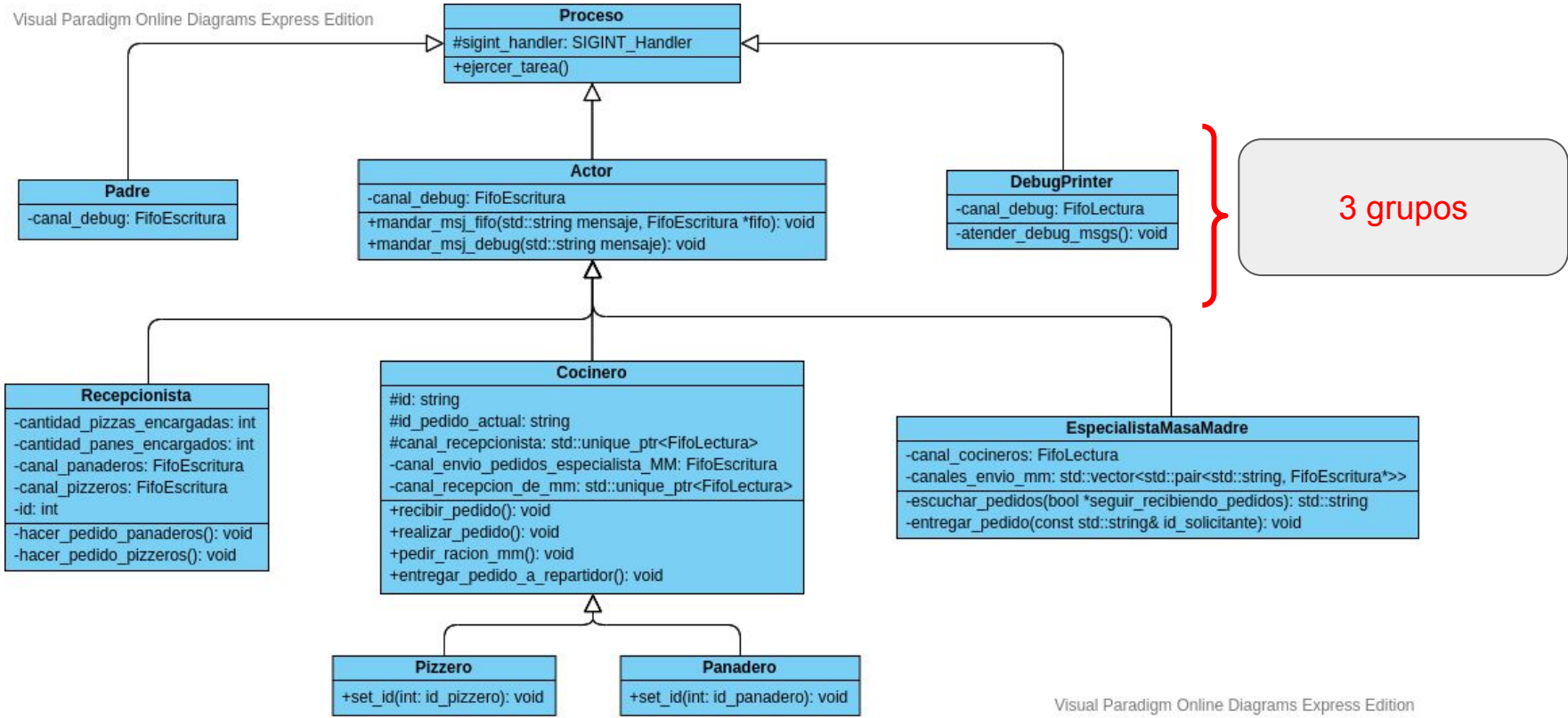
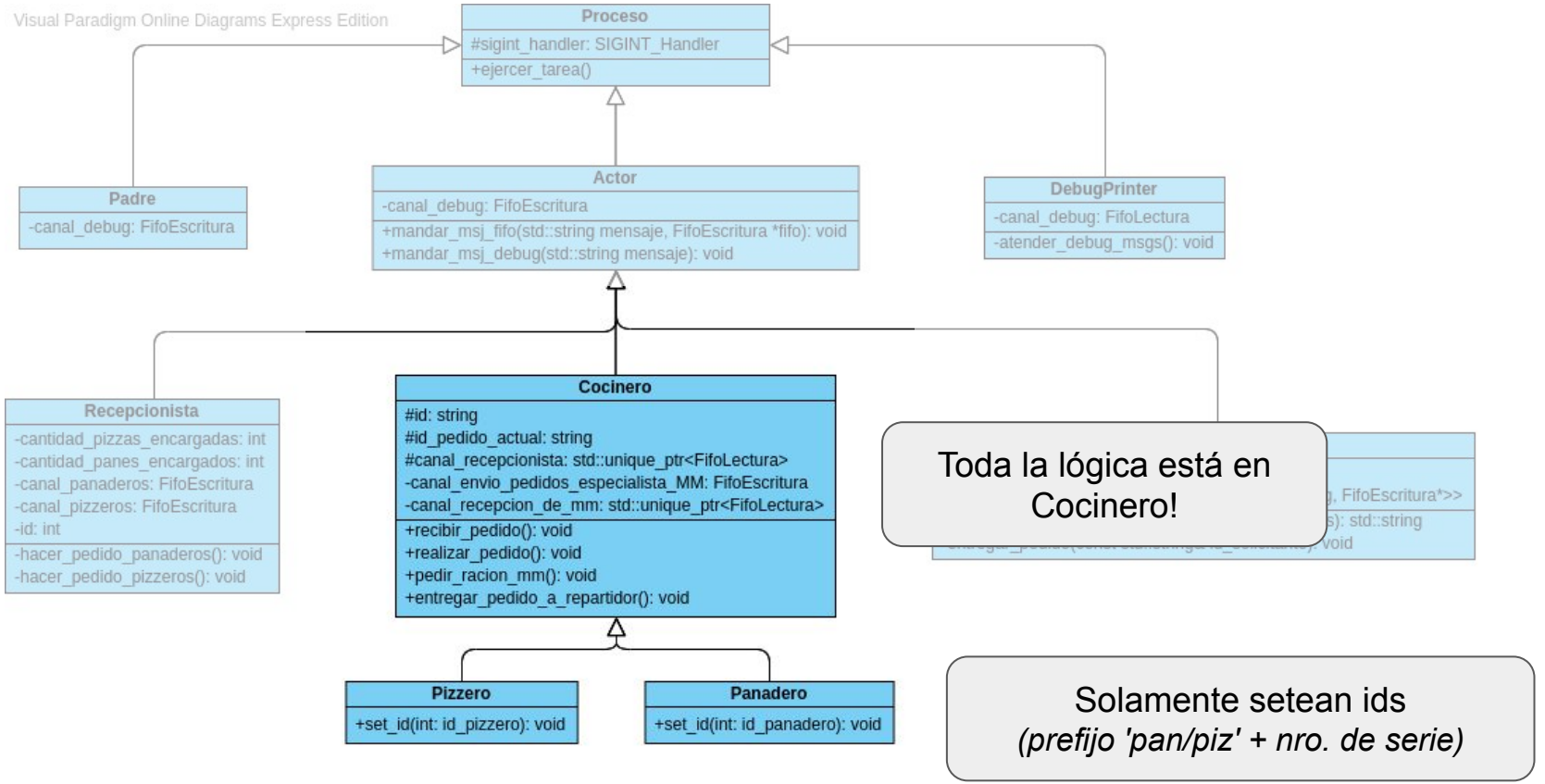
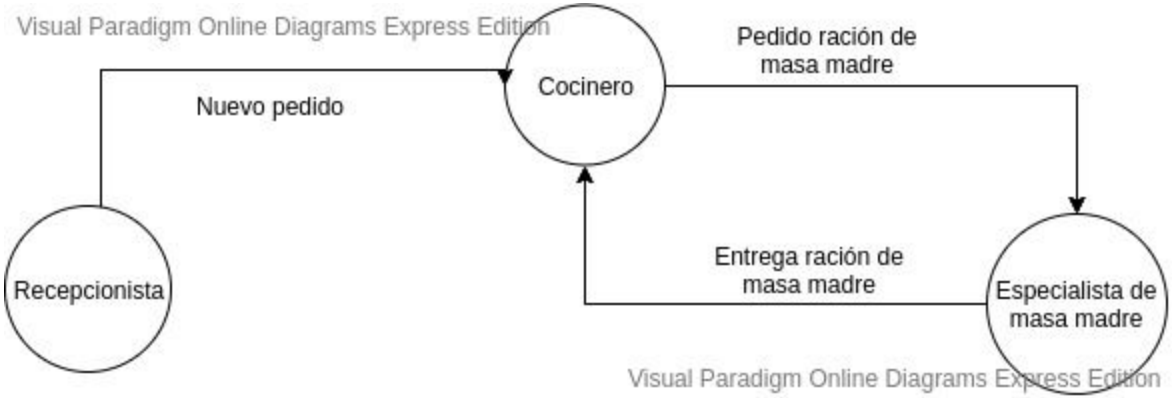


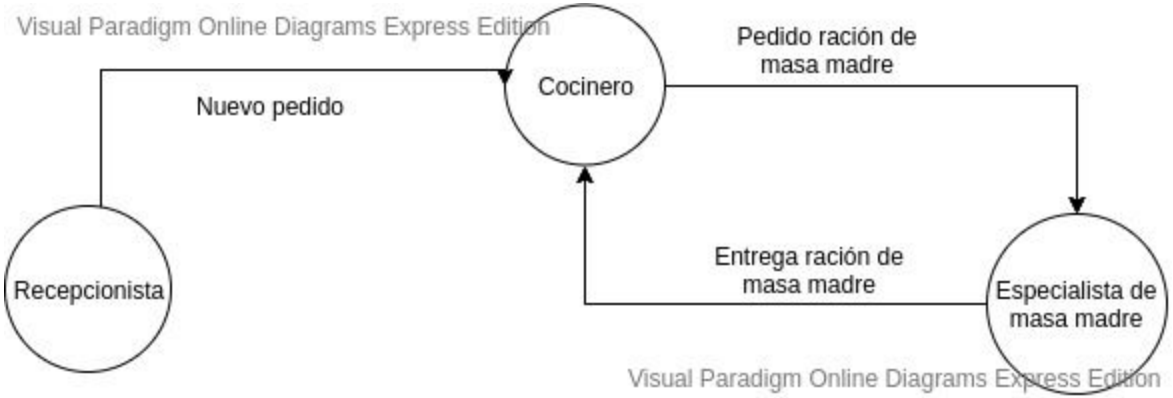
Diagrama de clases



Comunicaciones



Comunicaciones



NOTA Se excluye del diagrama la comunicación de todos los procesos hacia el Debug printer.

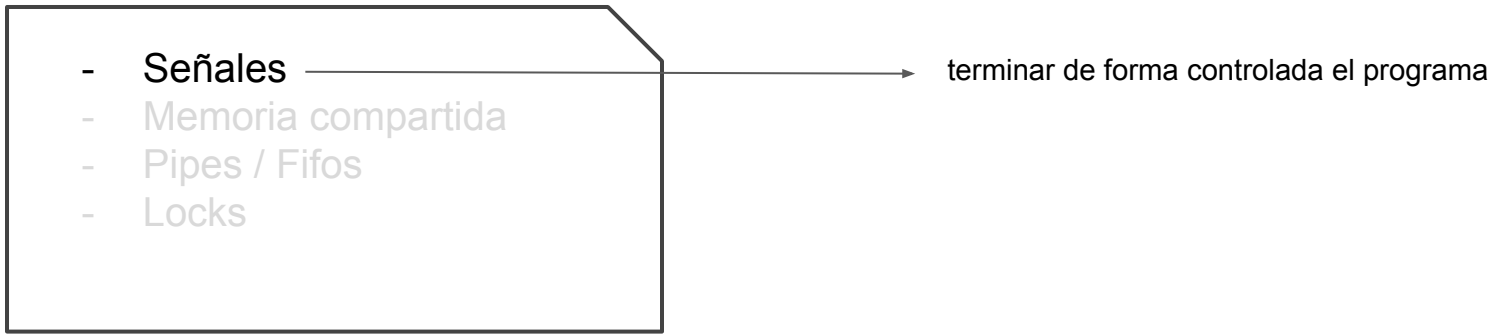
Comunicaciones

¿Qué opciones hay?

- Señales
- Memoria compartida
- Pipes / Fifos
- Locks

Comunicaciones

¿Qué opciones hay?



Comunicaciones

¿Qué opciones hay?

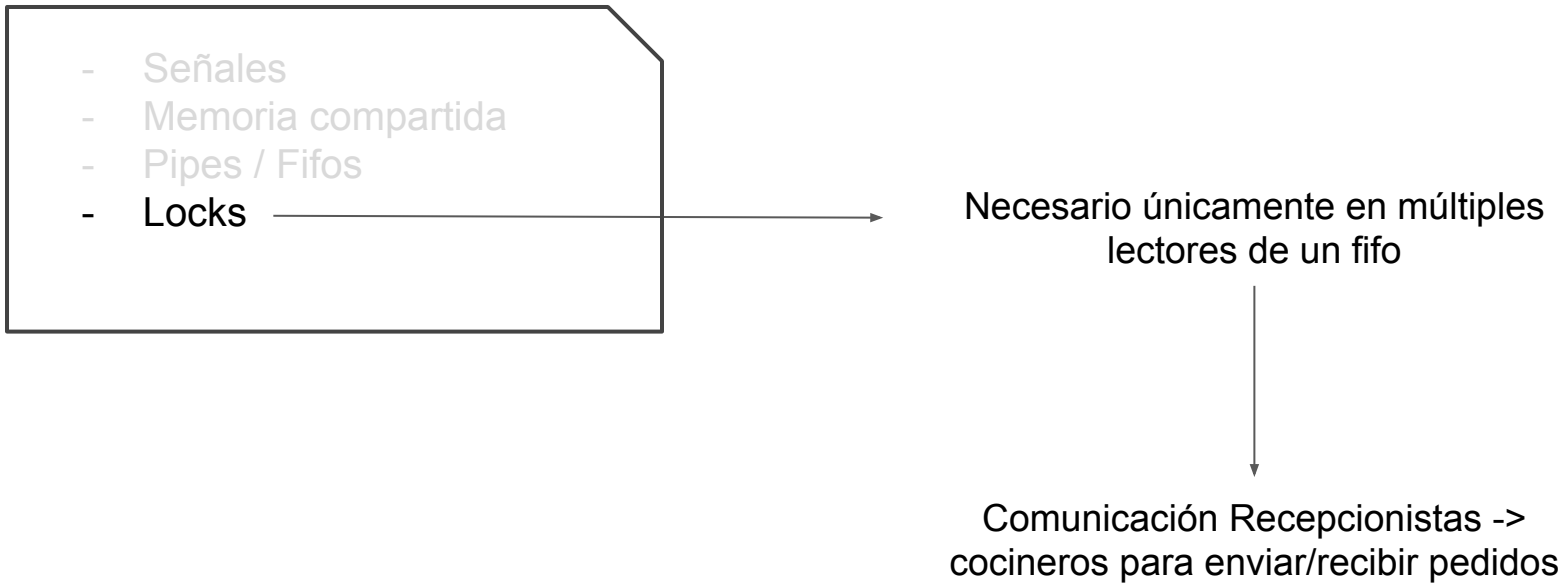
- Señales
- ~~Memoria compartida~~
- Pipes / Fifos
- Locks

~~Pipes~~ vs fifos

✔️ Flexibilidad adicional que posibilita comunicación 1:N en los casos en que esto es requerido.

Comunicaciones

¿Qué opciones hay?



Comunicaciones

Protocolos: Recepcionista

- Solamente envía el id del pedido
- El pedido queda conformado por el id del pedido, de longitud fija

Id pedido [7 chars]				
"Piz" "Pan"	+	Id recepcionista [2 dígitos]	+	Nro. de serie de la pizza o pan del recepcionista [2 dígitos]

Comunicaciones

Protocolos: Recepcionista

Id pedido [7 chars]

"Piz" / "Pan"	+	Id recepcionista [2 dígitos]	+	Nro. de serie de la pizza o pan del recepcionista [2 dígitos]

Ejemplo con 2 recepcionistas con 2 panes y 2 pizzas cada uno

```
-   Piz0000     Recepcionista 0, primer      pizza que encarga.
-   Piz0001     Recepcionista 0, segunda     pizza que encarga.
-   Pan0000     Recepcionista 0, primer      pan que encarga.
-   Pan0001     Recepcionista 0, segundo     pan que encarga.
-   Piz0100     Recepcionista 1, primer      pizza que encarga.
-   Piz0101     Recepcionista 1, segunda     pizza que encarga.
-   Pan0100     Recepcionista 1, primer      pan que encarga.
-   Pan0101     Recepcionista 1, segundo     pan que encarga.
```

Comunicaciones

Protocolos: Cocineros

- Piden ración masa madre
- El pedido queda conformado por el id del que pide
    - Suficiente para que el Especialista sepa a quién enviarle la ración

En este paso se crea el fifo correspondiente a cada cocinero para poder recibir la ración solicitada.

Comunicaciones

Protocolos: Cocineros

- Piden ración masa madre
- El pedido queda conformado por el id del que pide
    - Suficiente para que el Especialista sepa a quién enviarle la ración

En este paso se crea el fifo correspondiente a cada cocinero para poder recibir la ración solicitada.

Open de fifo es bloqueante. ¿Es un problema esto?

Comunicaciones

Protocolos: Cocineros

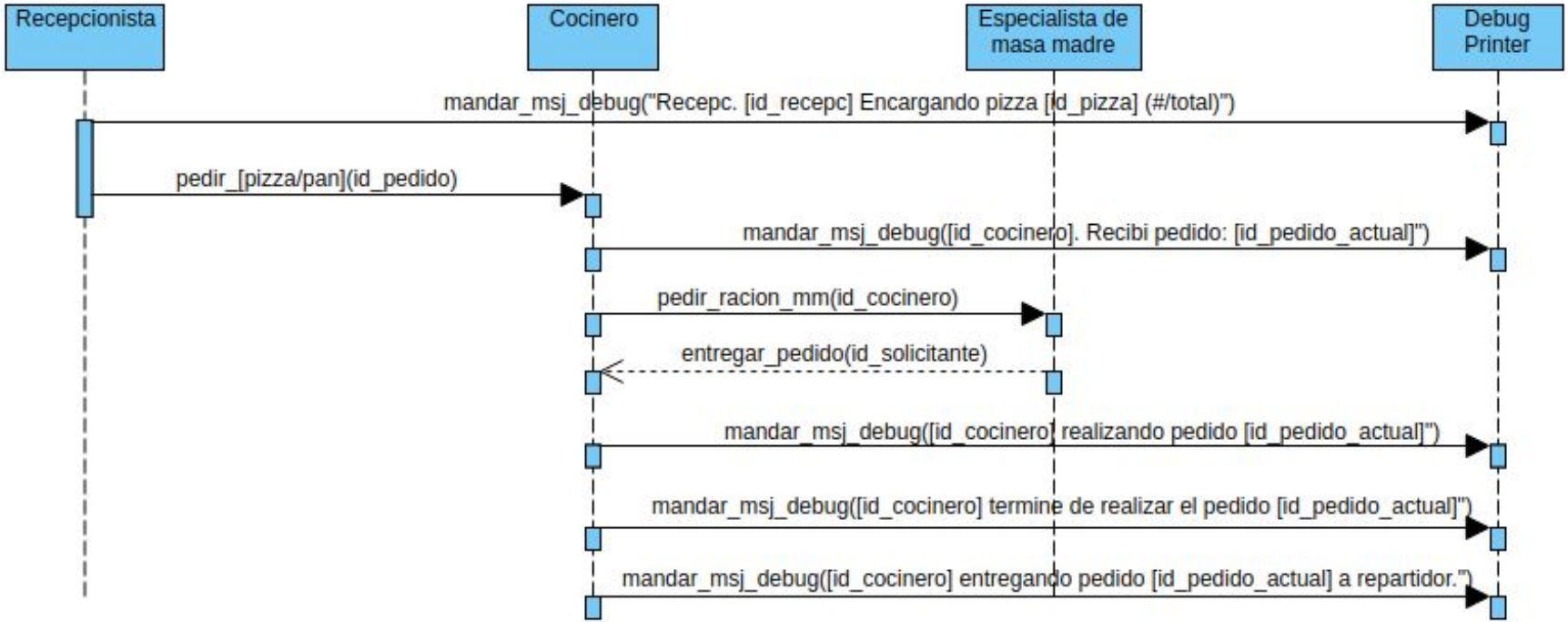
Open de fifo es bloqueante. ¿Es un problema esto?

NO

- Porque se maneja al momento del pedido.
  - El cocinero no puede seguir hasta que no reciba la masa, por lo que está bloqueado de todas formas.
- El Especialista crea el fifo al recibir el pedido.

Comunicaciones

Diagrama



Código

Cocinero

```
void Cocinero::ejercer_tarea() {
   bool seguimos_recibiendo = true;
   while (sigint_handler.getGracefulQuit() == 0 and seguimos_recibiendo) {
       recibir_pedido(&seguimos_recibiendo);
       if (seguimos_recibiendo) {
           realizar_pedido();
           entregar_pedido_a_repartidor();
       }
   }
   mandar_msj_debug(id + " termino con todos sus deberes.");
}
```

Código

Cocinero

```
void Cocinero::ejercer_tarea() {
    bool seguimos_recibiendo = true;
    while (sigint_handler.getGracefulQuit() == 0 and seguimos_recibiendo) {
        recibir_pedido(&seguimos_recibiendo);
        if (seguimos_recibiendo) {
            realizar_pedido();
            entregar_pedido_a_repartidor();
        }
    }
    mandar_msj_debug(id + " termino con toc
}
```

```
pedir_racion_mm();
esperar_envio_mm();

mandar_msj_debug(id + " realizando pedido " + id_pedido_actual);

sleep(dis(gen));

mandar_msj_debug(id + " termine de realizar el pedido " + id_pedido_actual);
```

Realizar pedido

Tiempo aleatorio de preparación de pedido

Entrega

Un simple mensaje a Debug para loguear que se terminó con el pedido

¿Preguntas?


Gracias