

A continuación, se presenta un código que se utilizaría en caso de que, en la implementación de las colas, se desee usar el mecanismo push para enviar los mensajes

En el archivo server.py se pondría lo siguiente:

```
def envioMensaje(self):
    if (len(self.consumidoresConectados) != 0):
        valor = 0
        for cliente in self.consumidoresConectados:
            valor = valor + 1
            idCliente = cliente
            arreglo = self.consumidoresConectados[idCliente]
            conexionAplicacion = arreglo[0]
            direccionAplicacion = arreglo[1]
            nombre = arreglo[2]
            idCola = arreglo[3]
            estado = arreglo[4]
            try:
                if (self.colas[int(idCola)].getTamañoCola() != 0):
                    consumidores = self.colas[int(idCola)].getClientes()
                    estados = self.colas[int(idCola)].getEstadosClientes()
                    for i in range(0, len(consumidores)):
                        clienteArreglo = consumidores[i]
                        estadoArreglo = estados[i]
                        if (idCliente == clienteArreglo and estadoArreglo == False and
                            len(consumidores) == 1 and self.colas[int(idCola)].getTamañoCola() > 1):
                            contadorAux = 0
                            while contadorAux < self.colas[int(idCola)].getTamañoCola():
                                respuesta = ""
                                mensajeEnviar = self.colas[int(idCola)].cambiarIndiceEnvio()
                                respuesta = f"Repsuesta para: {direccionAplicacion} Tiene un nuevo
mensaje: {mensajeEnviar}\n"
                                conexionAplicacion.sendall(respuesta.encode(constants.ENCODING_FORMAT))
                                elif (idCliente == clienteArreglo and estadoArreglo == False and
                                    len(consumidores) > 1):
                                    respuesta = ""
                                    mensajeEnviar = self.colas[int(idCola)].cambiarIndiceEnvio()
                                    respuesta = f"Repsuesta para: {direccionAplicacion} Tiene un nuevo mensaje:
{mensajeEnviar}\n"
                                    conexionAplicacion.sendall(respuesta.encode(constants.ENCODING_FORMAT))
                                    self.colas[int(idCola)].cambiarEstadoCliente(i)
                                    if (i == len(consumidores)-1):
                                        self.colas[int(idCola)].actualizarTodosConsumidores()
                                        break
                                elif (idCliente == clienteArreglo and estadoArreglo == False):
                                    respuesta = ""
                                    mensajeEnviar = self.colas[int(idCola)].cambiarIndiceEnvio()
                                    respuesta = f"Repsuesta para: {direccionAplicacion} Tiene un nuevo mensaje:
{mensajeEnviar}\n"
                                    conexionAplicacion.sendall(respuesta.encode(constants.ENCODING_FORMAT))
                                    if (i == len(consumidores)-1):
                                        self.colas[int(idCola)].actualizarTodosConsumidores()
                                        break
                                else:
                                    respuesta = f"Repsuesta para: {direccionAplicacion} No hay mensajes nuevos\n"
                                    conexionAplicacion.sendall(respuesta.encode(constants.ENCODING_FORMAT))
            except:
                continue
```

El anterior código permite el envío de mensajes desde el MOM a los consumidores usando push, y enviando mensajes de este modo: el mensaje 1 para el consumidor1 conectado, el mensaje 2 para el consumidor 2 conectado ... el mensaje m al consumidor n conectado,

El siguiente código es para establecer una conexión entre un consumidor y el servidor:

```
elif (opcion == "CONECTAR-CONSUMIDOR"):
    print(f'{direccionAplicacion[0]} solicita: {opcion}')
    nombreCola = arreglo[1]
    idCola = arreglo[2]
    respuesta = ""
    try:
        nombreAux = self.colas[int(idCola)].getNombre()
        idAux = self.colas[int(idCola)].getId()
        if (str(idCola) == str(idAux) and str(nombreCola) == str(nombreAux)):
            respuesta = f'Respuesta para: {direccionAplicacion[0]} La conexión se establecio
correctamente, ahora puedes recibir mensajes\n'
            conexionAplicacion.sendall(respuesta.encode(constants.ENCODING_FORMAT))
            arregloConsumidor = [conexionAplicacion,direccionAplicacion,nombreAux,idCola,True]
            self.colas[int(idCola)].agregarCliente(self.contadorConsumidores)
            self.consumidoresConectados[self.contadorConsumidores] = arregloConsumidor
            self.contadorConsumidores = self.contadorConsumidores + 1
            threadEnvio = threading.Thread(target=self.envioMensaje)
            threadEnvio.start()
            threadEnvio.join()
    except:
        respuesta = f'Respuesta para: {direccionAplicacion[0]} No hay conexión, prueba nuevamente\n'
        conexionAplicacion.sendall(respuesta.encode(constants.ENCODING_FORMAT))
        print(f'Se envia respuesta a: {direccionAplicacion[0]} por la solicitud: {opcion}')
```

De igual forma en el repositorio se sube una carpeta que se llama colas-push y corresponde al proyecto (únicamente la implementación de las colas) en funcionamiento con método push, es importante aclarar que esta parte del proyecto solo funciona de forma local, es decir, la dirección IP del servidor es local.