

# Centro Universitario de Ciencias Exactas e Ingenierías

Departamento de Ciencias Computacionales

Sistemas Operativos



Profesora: Becerra Velázquez Violeta del Rocío

Alumno: Monjaraz Briseño Luis Fernando

Código: 218520958

Carrera: Ingeniería en Computación

Sección: D04

Actividad 14 (Programa 7 Paginación simple)

Fecha: 26/11/2023

**Índice**

<b>Índice</b>	<b>2</b>
<b>Tabla de imágenes</b>	<b>3</b>
<b>Datos personales</b>	<b>4</b>
<b>Datos de la materia</b>	<b>4</b>
<b>Número de actividad</b>	<b>4</b>
<b>Objetivo de la actividad</b>	<b>4</b>
<b>Notas acerca del lenguaje</b>	<b>4</b>
<b>Conclusión</b>	<b>20</b>
Lomeli Jaime:	20
Monjaraz Luis:	20

**Tabla de imágenes**

Ilustración 1 setupUi	6
Ilustración 2 retranslateUi	7
Ilustración 3 init	8
Ilustración 4 start_stop_func	8
Ilustración 5 pedirProcesos	9
Ilustración 6 terminadoProceso	10
Ilustración 7 inicio	10
Ilustración 8 paginación	11
Ilustración 9 suspendido	12
Ilustración 10 regresar	13
Ilustración 11 ejecución	14
Ilustración 12 init 2	14
Ilustración 13 keyPressEvent	15
Ilustración 14 setupUi 2	16
Ilustración 15 retranslateUi 2	17
Ilustración 16 Ejemplo 1	18
Ilustración 17 Ejemplo 2	18
Ilustración 18 Ejemplo 3.	19
Ilustración 19 Ejemplo 4.	19

**Datos personales**

Nombres: Lomeli Navarro Jaime Adalberto y Monjaraz Briseño Luis Fernando.

Códigos: 221978094 y 218520958.

Correos: [jaime.lomeli7809@alumnos.udg.mx](mailto:jaime.lomeli7809@alumnos.udg.mx) y [luis.monjaraz5209@alumnos.udg.mx](mailto:luis.monjaraz5209@alumnos.udg.mx)

**Datos de la materia**

Materia: Sistemas Operativos

Sección: D04

Horario: Martes, Jueves, Sábado. 11:00 a 12:55

NRC: 204880

Clave: IL366

**Número de actividad**

Programa 7 Paginación simple

**Objetivo de la actividad**

El objetivo de esta actividad es crear un algoritmo de paginación simple, este mismo utiliza toda la estructura y funcionamiento de un algoritmo de planificación Round Robin. Siendo así que también sigue para su procesamiento el “Diagrama de 5 Estados”, sin embargo, en este caso los procesos no se saben cuántos serán, ya que para entrar a listos estos deben de tener marcos libres, así que realmente no se saben cuántos caben, además de eso cada proceso cuenta con un tamaño de entre 6 a 26. Esta actividad es nos ayudara a comprender el funcionamiento de una paginación simple, esto de una manera visual y práctica, pues al momento de ejecutar el programa se muestra la fragmentación que existe.

En resumen, esta actividad es sumamente útil para comprender la teoría de la paginación simple y entender el concepto de fragmentación.

**Notas acerca del lenguaje**

Lenguaje usado: Python.

Motivo: Principalmente escogimos este lenguaje por que decidimos trabajar en equipo para esta actividad, y al ser el lenguaje que mejor sabemos usar los dos se llegó a la conclusión de utilizarlo. También es porque nuestros códigos ya no era sostenible actualizarlos, por lo que lo mejor era iniciarlo de cero, cosa que nos ayudó mucho, pues Python cuenta con muchas herramientas que facilitan mucho la programación de la simulación de los procesos, además, aprovechando que en Análisis de algoritmos hemos estado trabajando con la librería “PyQT” para interfaces graficas decidimos emplearla para nuestra actividad, cosa que consideramos que fue todo un éxito.

Estructuras: Utilizamos “clases” para la resolución del problema, una llamada “Ui\_widget” y “MainWindow”. En estas clases la crearon varias funciones. La clase Ui\_widget es la que se encarga principalmente del funcionamiento del programa, esto a nivel interno y

visual, mientras que la clase “MainWindows” se encarga más que nada de las pulsaciones de las teclas y otros detalles. Además de eso se utilizó otro archivo Python solamente para el BCP, en este si utilizo la clase “Ui\_Form”.

El código es de  $1585 + 96 = 1681$  líneas de código.

Funciones:

- setupUi: Esta es principalmente para el diseño de la interfaz con PyQt.
- retranslateUi: Esta es principalmente para el diseño, sin embargo, este se ira actualizando conforme avance el programa.
- \_\_init\_\_: Esta función es para inicializar algunas variables.
- start\_stop\_func: Inicializamos tiempos de bloqueo.
- pedirProcesos: Esta función es para solicitar la cantidad de procesos y asignarle sus respectivos datos de forma aleatoria.
- terminadoProceso: Esta función es la que se encarga de imprimir y manejar los procesos terminados.
- Inicio: Inicializa y controla contadores.
- Paginación: Esta función controla todo el funcionamiento de la paginación simple.
- Suspendido: Esta función se encarga de la suspensión, sin embargo, esto es para la próxima tarea, debido a que necesita correcciones, las cuales nos mencionó en clase.
- Regresar: Esta función se encarga de sacar a los procesos de suspendidos.
- Ejecución: Esta función se encarga de los procesos en ejecución.
- \_\_init\_\_: Inicializa variables de la segunda clase.
- keyPressEvent: Esta función se encarga de “recibir” las teclas que son pulsadas.
- setupUi: Esta función es para el diseño de la interfaz del BCP.
- retranslateUi: Esta función ajusta e imprime los valores del BCP.

Para el Quantum se creó una variable global llamada Quantum y a esta se le creo una copia, esto

Funciones:

```

25 def setupUi(self, widget):
26     widget.setObjectName("widget")
27     widget.resize(1921, 1019)
28     widget.setStyleSheet("QWidget#widget{\n"
29         "background-color: rgb(255, 239, 213);}")
30     self.label = QtWidgets.QLabel(widget)
31     self.label.setGeometry(QtCore.QRect(660, 10, 291, 40))
32     self.label.setStyleSheet("font: 20pt \"Algerian\";\n"
33         "\"background-color: rgb(255, 255, 255);")
34     self.label.setObjectName("Label")
35     self.label_2 = QtWidgets.QLabel(widget)
36     self.label_2.setGeometry(QtCore.QRect(10, 30, 221, 31))
37     self.label_2.setStyleSheet("font: 14pt \"Times New Roman\";")
38     self.label_2.setObjectName("Label_2")
39     self.proces = QtWidgets.QLineEdit(widget)
40     self.proces.setGeometry(QtCore.QRect(240, 30, 71, 31))
41     self.proces.setStyleSheet("font: 14pt \"Times New Roman\";")
42     self.proces.setText("")
43     self.proces.setObjectName("proces")
44     self.iniciar = QtWidgets.QPushButton(widget)
45     self.iniciar.setGeometry(QtCore.QRect(320, 70, 141, 31))
46     self.iniciar.setStyleSheet("font: 14pt \"Times New Roman\";")
47     self.iniciar.setObjectName("iniciar")
48
49     self.iniciar.clicked.connect(self.pedirProcesos)
50     self.iniciar.clicked.connect(self.inicio)
51     self.iniciar.clicked.connect(self.start_stop_func)
52
53     self.label_3 = QtWidgets.QLabel(widget)
54     self.label_3.setGeometry(QtCore.QRect(90, 160, 177, 31))
55     self.label_3.setStyleSheet("font: 14pt \"Times New Roman\";")
56     self.label_3.setObjectName("Label_3")
57     self.Nuevos = QtWidgets.QLabel(widget)
58     self.Nuevos.setGeometry(QtCore.QRect(280, 160, 131, 31))
59     self.Nuevos.setStyleSheet("font: 14pt \"Times New Roman\";\n"
60         "\"background-color: rgb(255, 255, 255);")
61     self.Nuevos.setText("")
62     self.Nuevos.setObjectName("Nuevos")
63     self.label_4 = QtWidgets.QLabel(widget)
64     self.label_4.setGeometry(QtCore.QRect(160, 460, 137, 31))
65     self.label_4.setStyleSheet("font: 14pt \"Times New Roman\";\n"
66         "\"background-color: rgb(255, 255, 255);")
67     self.label_4.setObjectName("Label_4")
68     self.tablaListos = QtWidgets.QTableWidget(widget)
69     self.tablaListos.setGeometry(QtCore.QRect(20, 500, 421, 231))
70     self.tablaListos.setStyleSheet("font: 10pt \"Times New Roman\";")
71     self.tablaListos.setObjectName("tablaListos")
72     self.tablaListos.setColumnCount(3)
73     self.tablaListos.setRowCount(0)
74     item = QtWidgets.QTableWidgetItem()
75     self.tablaListos.setHorizontalHeaderItem(0, item)
76     item = QtWidgets.QTableWidgetItem()
77     self.tablaListos.setHorizontalHeaderItem(1, item)
78     item = QtWidgets.QTableWidgetItem()

```

Ilustración 1 setupUi

```

520 def retranslateUi(self, widget):
521     _translate = QtCore.QCoreApplication.translate
522     widget.setWindowTitle(_translate("widget", "Form"))
523     self.label.setText(_translate("widget", "PS Suspendidos"))
524     self.label_2.setText(_translate("widget", "Número de procesos:"))
525     self.iniciar.setText(_translate("widget", "Iniciar"))
526     self.label_3.setText(_translate("widget", "Procesos nuevos:"))
527     self.label_4.setText(_translate("widget", "Cola de listos"))
528     item = self.tablaListos.horizontalHeaderItem(0)
529     item.setText(_translate("widget", "ID"))
530     item = self.tablaListos.horizontalHeaderItem(1)
531     item.setText(_translate("widget", "TME"))
532     item = self.tablaListos.horizontalHeaderItem(2)
533     item.setText(_translate("widget", "TT"))
534     self.label_5.setText(_translate("widget", "Proceso en ejecución"))
535     self.label_6.setText(_translate("widget", "Número de programa (ID):"))
536     self.label_7.setText(_translate("widget", "Operación a realizar:"))
537     self.label_8.setText(_translate("widget", "Tiempo maximo estimado:"))
538     self.label_9.setText(_translate("widget", "Tiempo transcurrido:"))
539     self.label_10.setText(_translate("widget", "Tiempo restante:"))
540     self.label_11.setText(_translate("widget", "Procesos Bloqueados"))
541     item = self.bloqueados.horizontalHeaderItem(0)
542     item.setText(_translate("widget", "ID"))
543     item = self.bloqueados.horizontalHeaderItem(1)
544     item.setText(_translate("widget", "TT"))
545     self.label_12.setText(_translate("widget", "Procesos terminados"))
546     item = self.Terminados.horizontalHeaderItem(0)
547     item.setText(_translate("widget", "ID"))
548     item = self.Terminados.horizontalHeaderItem(1)
549     item.setText(_translate("widget", "Operación"))
550     item = self.Terminados.horizontalHeaderItem(2)
551     item.setText(_translate("widget", "Resultado"))
552     self.label_13.setText(_translate("widget", "Reloj:"))
553     self.label_14.setText(_translate("widget", "Quantum:"))
554     self.label_15.setText(_translate("widget", "Quantum:"))
555     self.label_16.setText(_translate("widget", "Valor del Quantum:"))
556     self.label_17.setText(_translate("widget", "Proceso a entrar"))
557     self.label_18.setText(_translate("widget", "ID:"))
558     self.label_19.setText(_translate("widget", "Tamaño:"))
559     item = self.tableWidget.verticalHeaderItem(0)
560     item.setText(_translate("widget", "0"))
561     item = self.tableWidget.verticalHeaderItem(1)
562     item.setText(_translate("widget", "1"))
563     item = self.tableWidget.verticalHeaderItem(2)
564     item.setText(_translate("widget", "2"))
565     item = self.tableWidget.verticalHeaderItem(3)
566     item.setText(_translate("widget", "3"))
567     item = self.tableWidget.verticalHeaderItem(4)
568     item.setText(_translate("widget", "4"))
569     item = self.tableWidget.verticalHeaderItem(5)
570     item.setText(_translate("widget", "5"))
571     item = self.tableWidget.verticalHeaderItem(6)

```

Ilustración 2 retranslateUi

## Paginación simple

```
667
668     def __init__(self):
669         self.step = 0
670
671         self.timer = QTimer(self)
672         self.timer.timeout.connect(self.ejecucion)
673
674
```

*Ilustración 3 init*

```
674
675     def start_stop_func(self):
676         if not self.timer.isActive():
677             self.timer.start(800)
678         else:
679             self.timer.stop()
680
```

*Ilustración 4 start\_stop\_func*



```

716
717 def pedirProcesos(self):
718     try:
719         proses = int(self.proces.text())
720         self.quantum = int(self.quant.text())
721         _translate = QtCore.QCoreApplication.translate
722         self.proces.setText(_translate("Dialog", ""))
723         self.quant.setText(_translate("Dialog", ""))
724
725     except ValueError:
726         print("El valor ingresado no es un número, vuelva a ingresar")
727
728     contador = 0
729
730     tme = 0
731     oper = 0
732
733     _translate = QtCore.QCoreApplication.translate
734     self.valorQuant.setText(_translate("Dialog", str(self.quantum)))
735
736     while contador < proses:
737         lista = []
738
739         lista.append(self.ids)
740
741         tme = random.randint(6, 16)
742         lista.append(tme)
743
744         lista.append(0)
745
746         oper = random.randint(1, 6)
747         if oper == 1:
748             num1 = random.randint(0, 1000)
749             num2 = random.randint(0, 1000)
750             operacion = str(num1) + "+" + str(num2)
751             resultado = num1 + num2
752         elif oper == 2:
753             num1 = random.randint(0, 1000)
754             num2 = random.randint(0, 1000)
755             operacion = str(num1) + "-" + str(num2)
756             resultado = num1 - num2
757         elif oper == 3:
758             num1 = random.randint(-1000, 1000)

```

Ilustración 5 pedirProcesos

```
839
840 def terminadoProceso(self):
841     if self.lisd == [] and len(self.listaAux) > 0:
842         self.quantum2 = 1
843         self.lisd = self.listaAux.pop(0)
844     try:
845         if self.lisd[7] == 0:
846             self.lisd[7] = 1
847             self.lisd[6] = self.contadorGlobal
848     except IndexError:
849         pass
850
851     try:
852         if self.lisd[2] != 0:
853             self.cont = self.lisd[2] + 1
854         else:
855             self.cont = 1
856     except IndexError:
857         pass
858     self.terminador = 1
859
860     if len(self.listaLotes) == self.contPros:
861         self.bcp=QtWidgets.QDialog()
862         self.ui=Ui_Form()
863         self.ui.setupUi(self.bcp)
864         self.bcp.show()
865         cuenta = 0
866
867         self.ui.BCP.setRowCount(len(self.terminados))
868         for i in self.terminados:
869             self.ui.BCP.setItem(cuenta, 0, QTableWidgetItem(str(i[0])))
870             self.ui.BCP.setItem(cuenta, 1, QTableWidgetItem(str(i[1])))
871             self.ui.BCP.setItem(cuenta, 2, QTableWidgetItem(str(i[2])))
872             self.ui.BCP.setItem(cuenta, 3, QTableWidgetItem(str(i[3])))
873             self.ui.BCP.setItem(cuenta, 4, QTableWidgetItem(str(i[4])))
874             self.ui.BCP.setItem(cuenta, 5, QTableWidgetItem(str(i[5])))
875             self.ui.BCP.setItem(cuenta, 6, QTableWidgetItem(str(i[6])))
876             self.ui.BCP.setItem(cuenta, 7, QTableWidgetItem(str(i[7])))
```

Ilustración 6 terminadoProceso

```
883
884 def inicio(self):
885     try:
886         if self.lisd[2] != 0:
887             self.cont = self.lisd[2]
888         else:
889             self.cont = 1
890     except IndexError:
891         pass
892     self.terminador = 1
893
```

Ilustración 7 inicio

```

893
894     def paginacion(self):
895         try:
896             for x in range(len(self.memor)):
897                 if self.memor[x] == self.lisd[0]:
898                     lisAyuda = []
899                     lisAyuda.append(str(self.memor1[x]) + "/5")
900                     lisAyuda.append(self.memor[x])
901                     lisAyuda.append("Ejecucion")
902                     self.memor2[x] = lisAyuda
903         except IndexError:
904             pass
905         try:
906             for x in range(len(self.memor)):
907                 for i in self.listaAux:
908                     if self.memor[x] == i[0]:
909                         lisAyuda = []
910                         lisAyuda.append(str(self.memor1[x]) + "/5")
911                         lisAyuda.append(self.memor[x])
912                         lisAyuda.append("Listo")
913                         self.memor2[x] = lisAyuda
914         except IndexError:
915             pass
916         try:
917             for x in range(len(self.memor)):
918                 for i in self.prosBloq:
919                     if self.memor[x] == i[0]:
920                         lisAyuda = []
921                         lisAyuda.append(str(self.memor1[x]) + "/5")
922                         lisAyuda.append(self.memor[x])

```

Ilustración 8 paginación

```
957
958     def suspendido(self):
959         try:
960             listaP = self.prosBloq.pop(0)
961             self.suma2 = self.suma2 - listaP[14]
962             self.suspen.append(listaP)
963             for x in range(len(self.memor)):
964                 if self.memor[x] == listaP[0]:
965                     self.memor[x] = 0
966                     self.memor1[x] = 0
967         try:
968             while True:
969                 nuevalis = self.otraLista.pop(0)
970                 print(nuevalis)
971                 if nuevalis[14] <= 40 - self.suma2:
972                     memoria3 = nuevalis[14]
973                     memoria4 = nuevalis[13]
974                     memoria5 = int(memoria4/5)
975                     memoria4 = memoria4%5
976                     for x in range(len(self.memor)):
977                         if memoria3 == 0:
978                             break
979                         if self.memor[x] == 0:
980                             self.memor[x] = nuevalis[0]
981                             if memoria5 > 0:
982                                 self.memor1[x] = 5
983                                 memoria5 -= 1
984                             else:
985                                 self.memor1[x] = memoria4
986                                 memoria3 -= 1
987
988                     self.suma2 = self.suma2 + nuevalis[14]
989                     nuevalis[5] = self.contadorGlobal
990                     self.listaAux.append(nuevalis)
991                 else:
992                     self.otraLista = [nuevalis] + self.otraLista
993                     break
994             except IndexError:
995                 pass
996         except IndexError:
997             pass
998
```

Ilustración 9 suspendido

```

1008
1009 def regresar(self):
1010     try:
1011         lista = self.suspen.pop(0)
1012         print(lista)
1013
1014         tamano2 = int(lista[13]/5)
1015         tamano3 = tamano2
1016         tamano4 = lista[13]%5
1017         print(tamano2)
1018         self.suma = self.suma2
1019         self.suma = self.suma + tamano2
1020         if tamano4 != 0:
1021             self.suma = self.suma + 1
1022             tamano2 += 1
1023         if self.suma <= 40:
1024             self.suma2 = self.suma
1025             self.prosBloq.append(lista)
1026             for x in range(len(self.memor)):
1027                 if tamano2 == 0:
1028                     break
1029                 if self.memor[x] == 0:
1030                     self.memor[x] = lista[0]
1031                     if tamano3 > 0:
1032                         self.memor1[x] = 5
1033                         tamano3 -= 1
1034                     else:
1035                         self.memor1[x] = tamano4
1036                         tamano2 -= 1
1037             else:
1038                 self.suspen = [lista] + self.suspen
1039         except IndexError:
1040             pass
1041
1042         archivo = open("Procesos en Disco.txt", "w", encoding='utf8')
1043         try:
1044             for c in self.suspen:
1045                 archivo.write("ID: ")
1046                 archivo.write(str(c[0]))
1047                 archivo.write("\n")
1048             archivo.close()
1049         except:
1050             pass
1051

```

Ilustración 10 regresar

```

1052     def ejecucion(self):
1053         _translate = QtCore.QCoreApplication.translate
1054         self.step += 1
1055         if self.bandera == 3:
1056             self.bandera = 0
1057             self.start_stop_func()
1058         elif self.bandera == 6:
1059             self.bandera = 0
1060             self.suspendido()
1061         elif self.bandera == 7:
1062             self.bandera = 0
1063             self.regresar()
1064
1065         try:
1066             listoentrar = self.otraLista[0]
1067             self.Nuevos_2.setText(_translate("widget", str(listoentrar[0])))
1068             self.Nuevos_3.setText(_translate("widget", str(listoentrar[13])))
1069             self.Nuevos_4.setText(_translate("widget", str(listoentrar[14])))
1070         except IndexError:
1071             self.Nuevos_2.setText(_translate("widget", str(0)))
1072             self.Nuevos_3.setText(_translate("widget", str(0)))
1073             self.Nuevos_4.setText(_translate("widget", str(0)))
1074
1075         try:
1076             listoentrar = self.suspen[0]
1077             medida = len(self.suspen)
1078             self.disco1.setText(_translate("widget", str(medida)))
1079             self.disco2.setText(_translate("widget", str(listoentrar[0])))
1080             self.disco3.setText(_translate("widget", str(listoentrar[13])))
1081             self.disco4.setText(_translate("widget", str(listoentrar[14])))
1082         except IndexError:
1083             self.disco1.setText(_translate("widget", str(0)))
1084             self.disco2.setText(_translate("widget", str(0)))
1085             self.disco3.setText(_translate("widget", str(0)))
1086             self.disco4.setText(_translate("widget", str(0)))
1087
1088         for p in self.prosBloq:
1089             if p[15] == 8:
1090                 p.pop(8)
1091                 self.listaAux.append(p)
1092                 for n in self.prosBloq:
1093                     if n[0] == p[0]:
1094                         self.prosBloq.remove(n)
1095
1096         trabajos = len(self.otraLista)
1097         self.Nuevos.setText(_translate("widget", str(trabajos)))
1098

```

Ilustración 11 ejecución

```

class MainWindow(QMainWindow, Ui_widget):
    def __init__(self, parent=None):
        super(MainWindow, self).__init__(parent)
        self.setupUi(self)

```

Ilustración 12 init 2

```
def keyPressEvent(self, event):  
    if event.key() == Qt.Key_I:  
        self.bandera = 1  
        print("Usted a presionado I")  
    elif event.key() == Qt.Key_E:  
        self.bandera = 2  
        print("Usted a presionado E")  
    elif event.key() == Qt.Key_P:  
        self.bandera = 3  
        print("Usted a presionado P")  
    elif event.key() == Qt.Key_T:  
        self.bandera = 3  
        print("Usted a presionado T")  
    elif event.key() == Qt.Key_N:  
        self.bandera = 4  
        print("Usted a presionado N")  
    elif event.key() == Qt.Key_B:  
        self.bandera = 5  
        print("Usted a presionado B")  
    elif event.key() == Qt.Key_C:  
        self.start_stop_func()  
        print("Usted a presionado C")  
    elif event.key() == Qt.Key_S:  
        self.bandera = 6  
        print("Usted a presionado S")  
    elif event.key() == Qt.Key_R:  
        self.bandera = 7  
        print("Usted a presionado R")
```

*Ilustración 13 keyPressEvent*

```

15     def setupUi(self, Form):
16         Form.setObjectName("Form")
17         Form.resize(1535, 596)
18         Form.setStyleSheet("background-color: rgb(255, 239, 213);")
19         self.verticalLayout = QtWidgets.QVBoxLayout(Form)
20         self.verticalLayout.setObjectName("verticalLayout")
21         self.label = QtWidgets.QLabel(Form)
22         self.label.setStyleSheet("font: 20pt \"Algerian\";")
23         self.label.setObjectName("label")
24         self.verticalLayout.addWidget(self.label)
25         self.BCP = QtWidgets.QTableWidget(Form)
26         self.BCP.setStyleSheet("font: 10pt \"Times New Roman\";")
27         self.BCP.setObjectName("BCP")
28         self.BCP.setColumnCount(12)
29         self.BCP.setRowCount(0)
30         item = QtWidgets.QTableWidgetItem()
31         self.BCP.setHorizontalHeaderItem(0, item)
32         item = QtWidgets.QTableWidgetItem()
33         self.BCP.setHorizontalHeaderItem(1, item)
34         item = QtWidgets.QTableWidgetItem()
35         self.BCP.setHorizontalHeaderItem(2, item)
36         item = QtWidgets.QTableWidgetItem()
37         self.BCP.setHorizontalHeaderItem(3, item)
38         item = QtWidgets.QTableWidgetItem()
39         self.BCP.setHorizontalHeaderItem(4, item)
40         item = QtWidgets.QTableWidgetItem()
41         self.BCP.setHorizontalHeaderItem(5, item)
42         item = QtWidgets.QTableWidgetItem()
43         self.BCP.setHorizontalHeaderItem(6, item)
44         item = QtWidgets.QTableWidgetItem()
45         self.BCP.setHorizontalHeaderItem(7, item)
46         item = QtWidgets.QTableWidgetItem()
47         self.BCP.setHorizontalHeaderItem(8, item)
48         item = QtWidgets.QTableWidgetItem()
49         self.BCP.setHorizontalHeaderItem(9, item)
50         item = QtWidgets.QTableWidgetItem()
51         self.BCP.setHorizontalHeaderItem(10, item)
52         item = QtWidgets.QTableWidgetItem()
53         self.BCP.setHorizontalHeaderItem(11, item)
54         self.verticalLayout.addWidget(self.BCP)
55
56         self.retranslateUi(Form)
57         QtCore.QMetaObject.connectSlotsByName(Form)
58

```

Ilustración 14 setupUi 2



```
59 def retranslateUi(self, Form):
60     _translate = QtCore.QCoreApplication.translate
61     Form.setWindowTitle(_translate("Form", "Form"))
62     self.label.setText(_translate("Form", "Tabla de procesos"))
63     item = self.BCP.horizontalHeaderItem(0)
64     item.setText(_translate("Form", "ID"))
65     item = self.BCP.horizontalHeaderItem(1)
66     item.setText(_translate("Form", "Operación"))
67     item = self.BCP.horizontalHeaderItem(2)
68     item.setText(_translate("Form", "Resultado"))
69     item = self.BCP.horizontalHeaderItem(3)
70     item.setText(_translate("Form", "Estado"))
71     item = self.BCP.horizontalHeaderItem(4)
72     item.setText(_translate("Form", "TME"))
73     item = self.BCP.horizontalHeaderItem(5)
74     item.setText(_translate("Form", "Llegada"))
75     item = self.BCP.horizontalHeaderItem(6)
76     item.setText(_translate("Form", "Finalización"))
77     item = self.BCP.horizontalHeaderItem(7)
78     item.setText(_translate("Form", "Retorno"))
79     item = self.BCP.horizontalHeaderItem(8)
80     item.setText(_translate("Form", "Respuesta"))
81     item = self.BCP.horizontalHeaderItem(9)
82     item.setText(_translate("Form", "Servicio"))
83     item = self.BCP.horizontalHeaderItem(10)
84     item.setText(_translate("Form", "Espera"))
85     item = self.BCP.horizontalHeaderItem(11)
86     item.setText(_translate("Form", "Restante"))
87
```

Ilustración 15 retranslateUi 2

## Paginación simple

Como se debería de ver:

The interface is titled "Form" and contains several sections for process management:

- Número de procesos:** Input field.
- Reloj:** Input field.
- Quantum:** Input field and "Iniciar" button.
- Valor del Quantum:** Input field.
- Procesos nuevos:** Input field.
- Proceso a entrar:** Input fields for ID, Tamaño, and Marcos.
- Procesos en Disco:** Input field for Cantidad en disco.
- Cola de listas:** Table with columns ID, TME, and TT.
- Procesos Bloqueados:** Table with columns ID and TT.
- Proceso en ejecución:** Input fields for Número de programa (ID), Operación a realizar, Tiempo máximo estimado, Tiempo transcurrido, Tiempo restante, and Quantum.
- Procesos terminados:** Table with columns ID, Operación, and Resultado.
- PS SUSPENDIDOS:** Two large empty tables with columns Marco, Tamaño, ID, and Estado.

The Windows taskbar at the bottom shows the date 15/11/2023 and time 08:04 p.m.

Ilustración 16 Ejemplo 1

The interface is populated with data:

- Número de procesos:** 2
- Reloj:** 2
- Quantum:** 5
- Valor del Quantum:** 5
- Procesos nuevos:** 3
- Proceso a entrar:** ID: 12, Tamaño: 20, Marcos: 4
- Procesos en Disco:** Cantidad en disco: 0
- Cola de listas:**

ID	TME	TT
1	2	10
2	3	14
3	4	7
4	5	6
5	6	10
6	7	14
7	8	12
- Procesos Bloqueados:** Empty table.
- Proceso en ejecución:** ID: 1, Operación a realizar: 428\*291, Tiempo máximo estimado: 8, Tiempo transcurrido: 2, Tiempo restante: 6, Quantum: 2
- Procesos terminados:** Empty table.
- PS SUSPENDIDOS:**

Marco	Tamaño	ID	Estado
0	5/5	1	Ejecucion
1	2/5	1	Ejecucion
2	5/5	2	Listo
3	5/5	2	Listo
4	5/5	3	Listo
5	5/5	3	Listo
6	5/5	3	Listo
7	2/5	3	Listo
8	5/5	4	Listo
9	5/5	4	Listo
10	5/5	4	Listo
11	5/5	4	Listo
12	5/5	4	Listo
13	5/5	5	Listo
14	5/5	5	Listo
15	5/5	6	Listo
16	5/5	6	Listo
17	5/5	6	Listo
18	5/5	6	Listo
19	5/5	7	Listo
20	5/5	7	Listo
21	5/5	7	Listo
22	5/5	7	Listo
23	5/5	7	Listo
24	1/5	7	Listo
25	5/5	8	Listo
26	5/5	8	Listo
27	4/5	8	Listo
28	5/5	9	Listo
29	1/5	9	Listo
30	5/5	10	Listo
31	5/5	10	Listo
32	5/5	10	Listo
33	5/5	10	Listo
34	2/5	10	Listo
35	5/5	11	Listo
36	5/5	11	Listo
37	3/5	11	Listo
38	Libre	Libre	Libre
39	Libre	Libre	Libre
40	5/5	50	50
41	5/5	50	50
42	5/5	50	50
43	5/5	50	50

Ilustración 17 Ejemplo 2

Form

?

×

**TABLA DE PROCESOS**

	ID	Operación	Resultado	Estado	TME	Llegada	Finalización	Retorno	Respuesta	Servicio	Espera	Restante
1	12	933-210	Null	Nuevo	13	Null	Null	Null	Null	Null	Null	13
2	13	602-235	Null	Nuevo	9	Null	Null	Null	Null	Null	Null	9
3	14	118-135	Null	Nuevo	11	Null	Null	Null	Null	Null	Null	11
4	3	-443*-563	Null	Listo	14	0	Null	Null	Null	0	8	14
5	4	646MOD-71	Null	Listo	7	0	Null	Null	Null	0	8	7
6	5	62*-615	Null	Listo	6	0	Null	Null	Null	0	8	6
7	6	-531-/959	Null	Listo	10	0	Null	Null	Null	0	8	10
8	7	864MOD381	Null	Listo	14	0	Null	Null	Null	0	8	14
9	8	484MOD439	Null	Listo	12	0	Null	Null	Null	0	8	12
10	9	708-93	Null	Listo	12	0	Null	Null	Null	0	8	12
11	10	817-797	Null	Listo	10	0	Null	Null	Null	0	8	10
12	11	204*-413	Null	Listo	13	0	Null	Null	Null	0	8	13
13	1	428*291	Null	Listo	8	0	Null	Null	0	5	3	3
14	2	-562-/323	Null	Ejecución	10	0	Null	Null	5	3	5	7

Ilustración 18 Ejemplo 3.

Form

Número de procesos:  Reloj: 20

Quantum:  **Iniciar**

Valor del Quantum: 10

Procesos nuevos: 0

Proceso a entrar

ID: 0 Tamaño: 0 Marcos: 0

Procesos en Disco

Cantidad en disco 0

ID: 0 Tamaño: 0 Marcos: 0

Cola de listos

	ID	TME	TT
1	5	10	0
2	6	11	0
3	7	10	0
4	8	7	0
5	9	11	0
6	10	12	0
7	1	10	5

Procesos Bloqueados

	ID	TT
1	3	2

**PS SUSPENDIDOS**

Proceso en ejecución

Número de programa (ID): 4

Operación a realizar: 229MOD-661

Tiempo máximo estimado: 12

Tiempo transcurrido: 2

Tiempo restante: 10

Quantum: 2

Procesos terminados

	ID	Operación	Resultado
1	2	729-69*	798

Marcos	Tamaño	ID	Estado
0	5/5	1	Listo
1	4/5	1	Listo
2	Libre	Libre	Libre
3	Libre	Libre	Libre
4	Libre	Libre	Libre
5	5/5	3	Bloqueado
6	5/5	3	Bloqueado
7	5/5	3	Bloqueado
8	5/5	3	Bloqueado
9	3/5	3	Bloqueado
10	5/5	4	Ejecucion
11	5/5	4	Ejecucion
12	5/5	4	Ejecucion
13	1/5	4	Ejecucion
14	5/5	5	Listo
15	1/5	5	Listo
16	5/5	6	Listo
17	5/5	6	Listo
18	5/5	6	Listo
19	5/5	7	Listo
20	3/5	7	Listo
21	5/5	8	Listo
22	5/5	8	Listo
23	5/5	8	Listo
24	5/5	8	Listo

Marcos	Tamaño	ID	Estado
25	2/5	8	Listo
26	5/5	9	Listo
27	5/5	9	Listo
28	5/5	10	Listo
29	5/5	10	Listo
30	1/5	10	Listo
31	Libre	Libre	Libre
32	Libre	Libre	Libre
33	Libre	Libre	Libre
34	Libre	Libre	Libre
35	Libre	Libre	Libre
36	Libre	Libre	Libre
37	Libre	Libre	Libre
38	Libre	Libre	Libre
39	Libre	Libre	Libre
40	5/5	50	50
41	5/5	50	50
42	5/5	50	50
43	5/5	50	50

Buscar

ARS/MXN -0.20%

ESP

15/11/2023

Ilustración 19 Ejemplo 4.

Enlace de descarga (contenido):

<https://drive.google.com/drive/folders/1biPHuknNuTjFZLdFk0bWMacDbdEAHwiE?usp=sharing>

### ***Conclusión***

#### **Lomeli Jaime:**

Para finalizar la actividad, tuvimos complicaciones debido a la implementación gráfica, teníamos ya una idea de cómo aplicarlo, pero la base que tuvimos del uso de Pyqt5 en otra materia fue muy básica, entonces, sumado a la propia complejidad del problema, tuvimos que investigar funciones y estructuras visuales correctas para nuestra representación de la paginación simple.

Este problema en grandes rasgos es una propuesta bastante compleja, lleva su gran cantidad de tiempo en realizar, pero ya obtenido el resultado es bastante gratificante ver lo conseguido.

#### **Monjaraz Luis:**

En conclusión, esta actividad nos ayudó a reafirmar los conocimientos adquiridos sobre la paginación simple, en lo personal, esta actividad fue bastante compleja y tardada en realizar, pues tuvimos que iniciar de cero el código, aunque debo de decir que el trabajar en equipo si ayuda demasiado para la elaboración de este, pues en palabras simples “Si a uno no se le prende el foco al otro sí”. Debo de decir que la forma visual en la que se ve la fragmentación más la paginación me recuerda mucho al “administrador de tareas” que en teoría este es motivo de la práctica. En resumen, esta actividad considero que es bastante útil para complementar el tema de paginación simple.