



Manual Tecnico

- Requerimientos

Programado en:

- GFORTRAN
- Visual Studio Code
- Sistema operativo de 64 bits
- JSON-PARSER
- GRAPHVIZ
- CMAKE

Archivo Opciones Vista					
Procesos Rendimiento Historial de aplicaciones Inicio Usuarios Detalles Servicios					
Nombre		Estado	1% CPU	32% Memoria	1% Disco
Aplicaciones (3)					
> Administrador de tareas			0.1%	27.1 MB	0.1 MB/s
> Brave Browser (21)			0.5%	1,255.2 MB	0.1 MB/s
> Visual Studio Code (15)			0%	577.3 MB	0 MB/s

Visual Studio Code (15)	0%	582.2 MB	0 MB/s	0 Mbps	0%	GPU 0 - 3D	Muy baja	Muy baja
C/C++ Extension for Visual Stu...	0%	8.8 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
fpm	0%	2.0 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
Host de ventana de consola	0%	0.9 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
Menu	0%	0.5 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
Menu.f90 - t4 - Visual Studio C...	0%	41.6 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
Procesador de comandos de ...	0%	0.9 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
Visual Studio Code	0%	102.3 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
Visual Studio Code	0%	26.3 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
Visual Studio Code	0%	118.2 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
Visual Studio Code	0%	17.2 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
Visual Studio Code	0%	201.1 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
Visual Studio Code	0%	24.0 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
Visual Studio Code	0%	5.6 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
Visual Studio Code	0%	3.8 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja
Windows PowerShell	0%	29.0 MB	0 MB/s	0 Mbps	0%		Muy baja	Muy baja

- Código Principal

```
1  program Menu
2      use cliente
3      use cola_recepcion
4      use clientes_atendidos
5      use List_of_list_m
6      use impresora
7      use ventanillas
8      !-----
9      use json_module
10     !-----
11
12     implicit none
13     !-----LISTAS-----
14     type(simple_linked_list) :: lista_v
15     type(queue) :: list
16     type(impresora_list) :: impresora_gr
17     type(impresora_list) :: impresora_p
18     type(linked_list) :: clientes_atendidos_list, lista_finalP
19     type(List_of_list) :: lista_de_listas
20
21     integer :: id_value
22     integer :: img_pf
23     integer :: img_pg
24     integer :: numItems
25     integer :: ventanillas
26     character(len=1) :: opcion
27     type(client), pointer :: temp
28     character(:), allocatable :: name_p
29     integer :: cant
30     character(:), allocatable :: tipo, cliente
31     integer :: contador_pasos = 0
32     !-----
33     !-----
34     type(json_file) :: json ! Se declara una variable del tipo json_file
35     type(json_value), pointer :: listPointer, personPointer, attributePointer
36     type(json_core) :: jsonc
37     integer :: id_n, steps_n, big_n, small_n
38     logical :: found_n
39     character(:), allocatable :: cliente_nl
40     integer :: count_n
41
42     integer :: i, size, j ! Se declaran variables enteras
43     logical :: found
44     logical :: is_found
45     integer :: id, smallImg, bigImgs
46     character(:), allocatable :: name_pp
47     character(len=100) :: cliente_reporte
48     character(len=100) :: file
49     character(:), allocatable :: file_name
50     integer :: i_espera
51
52     character(len=50, dimension(40)) :: names = [ &
53         "Juan ", "Maria ", "Jose ", "Ana ", "Carlos ", &
54         "Laura ", "Pedro ", "Isabel ", "Miguel ", "Carmen ", &
55         "Antonio ", "Elena ", "Francisco", "Rosa ", "Luis ", &
56         "Teresa ", "Javier ", "Sofia ", "Manuel ", "Raquel ", &
57         "David ", "Beatriz ", "Diego ", "Silvia ", "Jorge ", &
58         "Natalia ", "Roberto ", "Patricia ", "Fernando ", "Eva ", &
59         "Pablo ", "Cristina ", "Alvaro ", "Victoria ", "Ruben ", &
60         "Lorena ", "Alberto ", "Nerea ", "Joaquin ", "Marta " ]
61     integer :: id_r, steps_r, big_r, small_R
62     logical :: found_r
63     integer :: unit
64     integer :: count_r
65     count_r=1
66     count_n=1
67     i_espera = 0
68     j=0
```

```

68 !-----
69 print *, "-----"
70 print *, "FASE 1 EDD - 202200007"
71 print *, "-----"
72 print *, "MENU PRINCIPAL"
73 print *, "1. CARGA DE CLIENTES"
74 print *, "2. CANTIDAD DE VENTANILLAS"
75 print *, "3. EJECUTAR PASO"
76 print *, "4. ESTADO EN MEMORIA DE LAS ESTRUCTURAS DE DATOS"
77 print *, "5. REPORTES"
78 print *, "6. ACERCA DE -INFO CREADOR"
79 print *, "7. Salir"
80 print *, "-----"
81
82 ! inicia el loop de mostrar menu
83 do
84     print *, "Ingrese una opcion: "
85     read(*,*) opcion
86
87     select case(opcion)
88     case('1')
89         print *, "CARGANDO CLIENTES:"
90         read(*,*) file
91         print *, file
92         call json%initialize()
93         call json%load(filename=file)
94         call json%print()
94         call json%print()
95
96         call json%info(' ', n_children=size)
97
98         call json%get_core(jsonc)
99         call json%get(' ', listPointer, found)
100
101         do i = 1, size
102             call jsonc%get_child(listPointer, i, personPointer, found = found)
103             call jsonc%get_child(personPointer, 'nombre', attributePointer, found = found)
104             if (found) then
105                 call jsonc%get(attributePointer, name_pp)
106             end if
107             call jsonc%get_child(personPointer, 'id', attributePointer, found = found)
108             if (found) then
109                 call jsonc%get(attributePointer, id)
110             end if
111
112             call jsonc%get_child(personPointer, 'img_g', attributePointer, found = found)
113             if (found) then
114                 call jsonc%get(attributePointer, bigImgs)

```

```

114         call jsonc%get(attributePointer, bigImgs)
115         end if
116
117         call jsonc%get_child(personPointer, 'img_p', attributePointer, found = found)
118         if (found) then
119             call jsonc%get(attributePointer, smallImg)
120         end if
121         call list%enqueue(id, smallImg, bigImgs, name_pp)
122         !append(self, id, name, bigImgs, smallImgs)
123         call clientes_atendidos_list%append(id, name_pp, bigImgs, smallImg)
124     end do
125
126     call json%destroy()
127     call list%print()
128
129     !call list%dequeue(temp)
130     call list%print()
131
132     case('2')
133         print *, "¿Cuántas ventanillas quiere crear?"
134         read(*, *) ventanillas
135         print *, "-----"
136     case('3')

```

```

136     print *
137     print *, "-----"
138     contador_pasos= contador_pasos +1
139     print *, "EJECUTANDO PASO...", contador_pasos
140     img_pf = 3
141     img_pg = 2
142     print *, "-----"
143     ! AGREGAR CLIENTES RANDOM -----
144
145     j = j+ 1
146     name_p = names(j)
147     id_value= contador_pasos
148     call list%enqueue(id_value, img_pf, img_pg, name_p)
149
150     call clientes_atendidos_list%append(id_value,name_p, img_pg,img_pf)
151     print *, "-----"
152     print *, "CLIENTE A ENTRAR A cola:", name_p
153     print *, "-----"
154     print *, "-----COLA-----"
155     call list%print()
156     !-----
157
158     call list%top_info(id_value, img_pf, img_pg)
159     call list%get_top_name(name_p)
160     print *, "-----"
161     print *, "ID DEL CLIENTE A ENTRAR A LA VENTANILLA:", id_value
162     print *, "-----"

```

```

162     print *, "-----"
163     ! Ver las ventanas
164     call lista_v%get_count(numItems)
165     if (numItems < ventanillas) then
166         ! AÑADIR A VENTANILLAS
167         call lista_v%append(name_p, id_value, img_pf, img_pg)
168         call list%dequeue(temp)
169
170         call lista_v%get_count(numItems)
171
172     else
173         print *, "-----ventanillas llenas-----"
174     endif
175
176     !IMPRESIONES, TIEMPO DE ESPERA EN VENTANILLA
177     call lista_v%update_img_pg()
178
179     !PASA A IMPRIMIR
180     call lista_v%remove_and_enqueue_to_impresora(impresora_gr, impresora_p)
181     print *, "-----PASO TERMINADO-----"
182     print *, "-----"
183     print *, "VENTANAS"
184     call lista_v%print()
185     call lista_v%get_count(numItems)
186     call lista_v%graficar_vent('ventanillas.dot')
187     print *, "-----"
188     print *, "COLA:"

```

```

188     print *, "COLA:"
189     call list%print()
190     print *, "-----"
191     print *, "-----IMPRESORAS-----"
192     call impresora_gr%impresora_grande(lista_de_listas,i_espera)
193     !subroutine impresora_mini(self, lista_of_listas, insert_index, lista_final, lista_total)
194     call impresora_p%impresora_mini(lista_de_listas,i_espera, lista_finalP, clientes_atendidos_list)
195     print *, "-----impresora grande -----"
196     call impresora_gr%print()
197     print *, "-----impresora peq -----"
198     call impresora_p%print()
199     i_espera=i_espera +1
200     print *, "-----LISTA DE ESPERA(lista de listas)-----"
201     call lista_de_listas%printList()
202
203
204     case('4')
205     print *, "ESTADO DE LAS ESTRUCTURAS DE DATOS:"
206     print *, "-----"
207     print *, "VENTANAS"
208     call lista_v%print()
209     call lista_v%get_count(numItems)
210     call lista_v%graficar_vent('ventanillas.dot')
211     print *, "-----"
212     print *, "COLA recepcion:"
213     call list%print()
214     call list%graficar('cola_recepcion.dot')

```

```

215     print *, "-----"
216     print *, "-----IMPRESORA-----"
217     print *, "-----impresora grande -----"
218     call impresora_gr%print()
219     call impresora_gr%graficar('impresora_grande.dot')
220     print *, "-----impresora peq -----"
221     call impresora_p%print()
222     call impresora_p%graficar('impresora_sm.dot')
223     print *, "-----"
224     print *, "-----LISTA DE ESPERA(lista de listas)-----"
225     call lista_de_listas%printList()
226     !call lista_de_listas%graficar_final('lista_espera.dot')
227     !call graficar_final(lista_de_listas, 'lista_espera.dot')
228     case('5')
229     print *, "REPORTES DE CLIENTES:"
230     print *, "-----CLIENTES ATENDIDOS -----"
231     call lista_final%print()
232     call lista_final%graficar('clientes.dot')
233     print *, "--Ingrese el nombre del cliente que desea generar el reporte-"
234     read(*,*) cliente_reporte
235     print *, "NOMBRE", trim(cliente_reporte)
236     file_name="cliente_reporte.dot"
237     call lista_final%searchname(trim(cliente_reporte), id_r, steps_r, big_r, small_r, found_r)
238     print *, "--1"
239     if (found_r) then
240         print *, "--2"
241         open(unit, file='cliente_reporte.dot', status='replace')

```

```

241         open(unit, file= cliente_reporte.dot , status= replace )
242         write(unit, *) 'digraph Impresora{'
243         write(unit, *) '    node [fillcolor="lightyellow", style="filled", shape="box"];'
244         write(unit, *) '    "Node", count_r, " [label="REPORTE CLIENTE nombre:', cliente_reporte, char(10), &
245         ' pasos:', steps_r, char(10), &
246         ' cantidad img pequeñas', small_r, char(10), &
247         'Cantidad de imagen grandes', big_r,'];'
248     else
249         print *, "Cliente no encontrado."
250     end if
251     print *, "--3"
252
253     write(unit, *) '}'
254     close(unit)
255
256     ! Generar el archivo PNG utilizando Graphviz
257     print *, "--4"
258     call system('dot -Tpng ' // trim('cliente_reporte.dot') // ' -o ' // trim(adjustl('cliente_reporte.dot')) // '.png')
259     print *, "--5"
260     print *, "-----"
261     print *, "-----REPORTE DE CLIENTE CON MAS PASOS-----"
262     print *, "-----"
263     !name, id, steps, bigImgs, smallImgs, found)
264     call lista_final%get_highest_steps_info(cliente_n1,id_n, steps_n, big_n, small_n, found_n)
265     print *, cliente_n1
266     !call lista_final%get_highest_steps_info(cliente_n,id_n, steps_n, big_n, small_n, found_n)
267     print *, "--1"

```

```

267     print *, "--1"
268     if (found_n) then
269         print *, "--2"
270         open(unit, file='cliente_reporte_pasos.dot', status='replace')
271         write(unit, *) 'digraph Impresora{'
272         write(unit, *) '    node [fillcolor="pink", style="filled", shape="box"];'
273         write(unit, *) '    "Node", count_n, " [label="MAS PASOS", cliente_n1, char(10), &
274         ' pasos:', steps_n, char(10), &
275         ' cantidad img pequeñas', small_n, char(10), &
276         'Cantidad de imagen grandes', big_n,"];'
277     else
278         print *, "Cliente no encontrado."
279     end if
280     print *, "--3"
281
282     write(unit, *) '}'
283     close(unit)
284
285     ! Generar el archivo PNG utilizando Graphviz
286     print *, "--4"
287     call system('dot -Tpng ' // trim('cliente_reporte_pasos.dot') // &
288     ' -o ' // trim(adjustl('cliente_reporte_pasos.dot')) // '.png')
289
290     print *, "--5"
291
292     case('6')
293         print *, "-----"
294         print *, "NATALIA MARIEL CALDERON ECHEVERRIA "
295         print *, "202200007"
296
297         print *, "-----"
298     case('7')
299         ! salir
300         exit
301     case default
302         print *, "ERROR - OPCION INVALIDA"
303     end select
304 end do
305
306 end program Menu

```

Descripción:

Al inicio de este programa se encuentran declaradas cada una de las variables a utilizar, así también como las instancias de las listas que se planean usar al momento de ejecución. Es decir, en esta parte, se declaran variables para las estructuras de datos utilizadas en el programa, como listas enlazadas y colas, se declaran variables para controlar el flujo del programa, como opciones de menú, contadores y banderas de estado y por último, también se declaran las variables utilizadas para manejar archivos JSON, como la carga y manipulación de datos estructurados.

Se muestra el menú principal del programa con opciones para cargar clientes, gestionar ventanillas, ejecutar pasos, generar reportes y salir del programa, este menú principal es la manera en la cual el cliente ejecuta la simulación y logra generar los datos deseados. El programa está envuelto en un bucle principal que solicita al usuario que ingrese una opción del menú y ejecuta la acción correspondiente según la opción seleccionada.

Casos del Menú:

Carga de Clientes (Opción 1)

Permite cargar clientes desde un archivo JSON, parsear los datos y agregarlos a una cola de recepción y a una lista de clientes atendidos.

Cantidad de Ventanillas (Opción 2)

Permite al usuario especificar la cantidad de ventanillas para la atención de clientes.

Ejecutar Paso (Opción 3)

Simula el paso del tiempo y la atención de clientes en ventanillas, impresión de documentos y gestión de colas, a su vez también muestra en consola quienes entran y quienes salen de cada una de las distintas estaciones

Estado de las Estructuras de Datos (Opción 4)

Muestra el estado actual de las estructuras de datos, como ventanillas, colas y dispositivos de impresión.

Reportes (Opción 5)

Permite generar reportes de clientes atendidos, incluyendo información sobre clientes específicos y clientes con más pasos.

Acerca de - Información del Creador (Opción 6)

Muestra información sobre el creador del programa.

Salir del Programa (Opción 7)

Termina la ejecución del programa.

Punto de vista del usuario

```
-----  
FASE 1 EDD - 202200007  
-----
```

```
MENU PRINCIPAL
```

```
1. CARGA DE CLIENTES
```

```
2. CANTIDAD DE VENTANILLAS
```

```
3. EJECUTAR PASO
```

```
4. ESTADO EN MEMORIA DE LAS ESTRUCTURAS DE DATOS
```

```
5. REPORTES
```

```
6. ACERCA DE -INFO CREADOR
```

```
7. Salir  
-----
```

```
Ingrese una opcion:
```

Resultado:

- 1

```
1
NUEVO CLIENTE EN COLA ----> CLIENTE: NATALIA
Se ha insertado correctamente el cliente con id: 1
NUEVO CLIENTE EN COLA ----> CLIENTE: VIKIRA
Se ha insertado correctamente el cliente con id: 2
NUEVO CLIENTE EN COLA ----> CLIENTE: TACHI
Se ha insertado correctamente el cliente con id: 3
NUEVO CLIENTE EN COLA ----> CLIENTE: MARIO
Se ha insertado correctamente el cliente con id: 4
NUEVO CLIENTE EN COLA ----> CLIENTE: LAURA
Se ha insertado correctamente el cliente con id: 5
NUEVO CLIENTE EN COLA ----> CLIENTE: USUARIOX
Se ha insertado correctamente el cliente con id: 6
NUEVO CLIENTE EN COLA ----> CLIENTE: USUARIOY
Se ha insertado correctamente el cliente con id: 7
NUEVO CLIENTE EN COLA ----> CLIENTE: USUARIOZZ
Se ha insertado correctamente el cliente con id: 8
ELEMENTOS EN LA COLA DE CLIENTES:
ID:      1 , img_g:      3 , img_p:      2 , name:NATALIA
ID:      2 , img_g:      3 , img_p:      1 , name:VIKIRA
ID:      3 , img_g:      2 , img_p:      4 , name:TACHI
ID:      4 , img_g:      3 , img_p:      2 , name:MARIO
ID:      5 , img_g:      1 , img_p:      3 , name:LAURA
ID:      6 , img_g:      2 , img_p:      4 , name:USUARIOX
ID:      7 , img_g:      4 , img_p:      2 , name:USUARIOY
ID:      8 , img_g:      2 , img_p:      1 , name:USUARIOZZ
```

-2

```

Ingrese una opcion:
2
  Cuantas ventanillas quiere crear?
3
-----
```

-3

```

Ingrese una opcion:
3
-----
EJECUTANDO PASO...      1
-----
NUEVO CLIENTE EN COLA ----> CLIENTE: Juan
Se ha insertado correctamente el cliente con id: 1
-----
CLIENTE A ENTRAR A cola:Juan
-----
-----COLA-----
ELEMENTOS EN LA COLA DE CLIENTES:
ID:      1 , img_g:      3 , img_p:      2 , name:NATALIA
ID:      2 , img_g:      3 , img_p:      1 , name:VIKIRA
ID:      3 , img_g:      2 , img_p:      4 , name:TACHI
ID:      4 , img_g:      3 , img_p:      2 , name:MARIO
ID:      5 , img_g:      1 , img_p:      3 , name:LAURA
ID:      6 , img_g:      2 , img_p:      4 , name:USUARIOX
ID:      7 , img_g:      4 , img_p:      2 , name:USUARIOY
ID:      8 , img_g:      2 , img_p:      1 , name:USUARIOZZ
ID:      1 , img_g:      2 , img_p:      3 , name:Juan
-----
ID DEL CLIENTE A ENTRAR A LA VENTANILLA: 1
-----
Nº de nodos en la lista: 1
-----
Se ha insertado correctamente el valor: NATALIA
-----
```


EL CLIENTE HA SIDO ATENDIDO Y ELIMINADO DE LA COLA--> SE ENCUENTRA EN VENTANILLA
-----PASO TERMINADO-----

VENTANAS

Value: NATALIA ID: 1 Img_p: 3 Img_g: 1
la representacion grafica de VENTANILLAS: ventanillas.dot.png

COLA:

ELEMENTOS EN LA COLA DE CLIENTES:

ID:	2 ,	img_g:	3 ,	img_p:	1 ,	name:VIKIRA
ID:	3 ,	img_g:	2 ,	img_p:	4 ,	name:TACHI
ID:	4 ,	img_g:	3 ,	img_p:	2 ,	name:MARIO
ID:	5 ,	img_g:	1 ,	img_p:	3 ,	name:LAURA
ID:	6 ,	img_g:	2 ,	img_p:	4 ,	name:USUARIOX
ID:	7 ,	img_g:	4 ,	img_p:	2 ,	name:USUARIOY
ID:	8 ,	img_g:	2 ,	img_p:	1 ,	name:USUARIOZZ
ID:	1 ,	img_g:	2 ,	img_p:	3 ,	name:Juan

-----IMPRESORAS-----

La cola de impresi|n est|í vac|ja.
La cola de impresi|n est|í vac|ja.

-----impresora grande -----

La lista est|í vac|ja.

-----impresora peq -----

La lista est|í vac|ja.

-----LISTA DE ESPERA(lista de listas)-----

-4

Ingrese una opcion:

4

ESTADO DE LAS ESTRUCTURAS DE DATOS:

VENTANAS

Value: MARIO ID: 4 Img_p: 2 Img_g: 0
Value: LAURA ID: 5 Img_p: 1 Img_g: 1
la representacion grafica de VENTANILLAS: ventanillas.dot.png

COLA repencion:

ELEMENTOS EN LA COLA DE CLIENTES:

ID:	6 ,	img_g:	2 ,	img_p:	4 ,	name:USUARIOX
ID:	7 ,	img_g:	4 ,	img_p:	2 ,	name:USUARIOY
ID:	8 ,	img_g:	2 ,	img_p:	1 ,	name:USUARIOZZ
ID:	1 ,	img_g:	2 ,	img_p:	3 ,	name:Juan
ID:	2 ,	img_g:	2 ,	img_p:	3 ,	name:Maria
ID:	3 ,	img_g:	2 ,	img_p:	3 ,	name:Jose
ID:	4 ,	img_g:	2 ,	img_p:	3 ,	name:Ana
ID:	5 ,	img_g:	2 ,	img_p:	3 ,	name:Carlos
ID:	6 ,	img_g:	2 ,	img_p:	3 ,	name:Laura
ID:	7 ,	img_g:	2 ,	img_p:	3 ,	name:Pedro
ID:	8 ,	img_g:	2 ,	img_p:	3 ,	name:Isabel

la representacion grafica de las COLA DE RECEPCION: cola_recepcion.dot.png

-----IMPRESORA-----

-----impresora grande -----

Tipo: impresora grande, Cliente: TACHI, Cant: 3
la representacion grafica de IMPRESORA: impresora_grande.dot.png

-----impresora peq -----

Tipo: impresora inicial, Cliente: TACHI, Cant: 1
la representacion grafica de IMPRESORA: impresora_sm.dot.png

-----LISTA DE ESPERA(lista de listas)-----

Cliente:TACHI

----- imagen-----

----- imagen-----

-5

```
Ingrese una opcion:
5
REPORTES DE CLIENTES:
-----CLIENTES ATENDIDOS -----
ELEMENTOS EN LA LISTA DE CLIENTES:
ID:          2 , nombre:VIKIRA, imagenes grandes:      3 , imagenes pequenas:      1 , Total :          4 , pasos:          6
la representacion grafica de las lista de clientes atendidos fue creada: clientes.dot.png
--Ingrese el nombre del cliente que desea generar el reporte-
NATALIA
NOMBRENATALIA
--1
--2
--3
--4
--5
-----
-----REPORTE DE CLIENTE CON MAS PASOS-----
-----
--1
--2
--3
--4
--5
```

- 6 y 7

```
Ingrese una opcion:
6
-----
NATALIA MARIEL CALDERON ECHEVERRIA
202200007
-----
Ingrese una opcion:
7
```