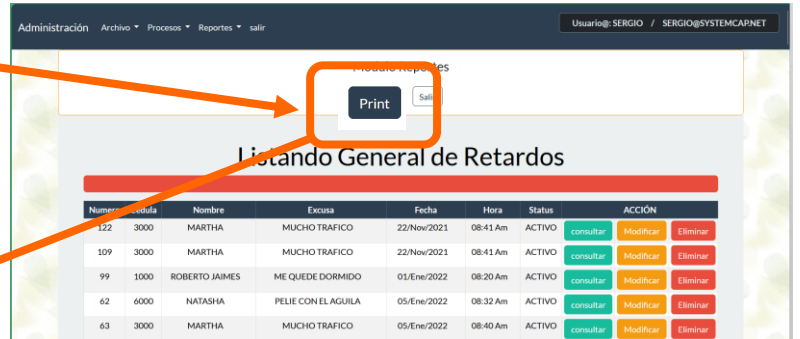
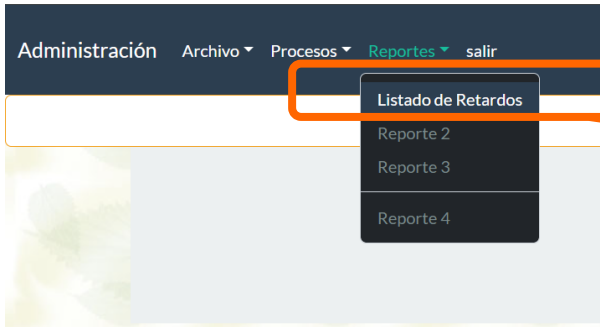


Reporte01

Listado general de retardos



Listado de Retardos



#	Cédula	Nombre	Excusa	Fecha	Hora
122	3000	MARTHA	MUCHO TRAFICO	22/Nov/2021	08:41 Am
109	3000	MARTHA	MUCHO TRAFICO	22/Nov/2021	08:41 Am
99	1000	ROBERTO JAIMES	ME QUEDE DORMIDO	01/Ene/2022	08:20 Am
62	6000	NATASHA	PELIE CON EL AGUILA	05/Ene/2022	08:32 Am
63	3000	MARTHA	MUCHO TRAFICO	05/Ene/2022	08:40 Am
64	2000	ZILA CONTRERAS	EL CARRO NO ME QUIERE	06/Ene/2022	09:00 Am
136	6000	NATASHA	ME QUEDE DORMIDO	12/May/2026	00:16 Am

1.- Modelo

1.1.- Programa ---→ modelo20_tdretardos.py

```
1 #Programa: modelo20_tdretardos.py
2 #Objetivo: Funciones para trabajar con la TDRETARDOS
3 #Autor: Sergio Capacho / Fecha: Martes 14/Mayo/2026 - 05:51 Pm
4
5 #1.- IMPORTAR DE LIBRERIAS
6 #.....
7 #.....
8 from flask import Blueprint, render_template, flash, redirect, url_for, request, Flask
9 from flask import session
10 from psycopg2 import connect
11
12
13 #2.- IMPORTAR EL MODELO Y LA CONEXIÓN CON LA BD
14 #.....
15 #.....
16
17 #2.1.- Importación Base de Datos
18 from modelos.modelo00_bd import cadenaConexion
19
20
21 #3.- OPERACIONES SQL CON TDRETARDOS
22 #.....
23 #.....
24
25 #3.1.- Registrar un retardo nuevo
26 def add_retardo(xfkcedemple, xfkcodexcu, xfecha, xhora):
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47 #3.2.- Obtener todos los registros de la Vista retardados00 (***)
48 def get_all_vretardos():
49     #PASO 1 ---> Establecer la conexión con la BD
50     conexion = cadenaConexion()
51     cursor = conexion.cursor()
52
53     #PASO 2 ---> Construir la consulta SQL
54     sql ="select * from vretardos00;"
55
56     #PASO 3 ---> Ejecutar la consulta SQL
57     cursor.execute(sql, ) #Ejecutar Insert/Update/Delete
58     codigo_conexion = conexion.commit() #Confirmar
59     resultado = cursor.fetchall() #fetchone() ---> Un solo registro / fetchall() ---> Varios
60
61     #PASO 4 ---> Cerrar la conexión con la BD
62     cursor.close()
63     conexion.close()
64
65     #PASO 5 ---> Retornar resultado
66     return resultado
67
68
```

Este código ya fue
analizado en clase

```

68
69 #3.3.- Determinar el total de retardos (Get ---> Obtener)
70 def get_total_retardo():
71     #PASO 1 ---> Establecer la conexión con la base de datos
72     conexion = cadenaConexion()
73     cursor = conexion.cursor()
74
75     #PASO 2 ---> Construir la consulta SQL
76     sql ="SELECT COUNT(pknr) FROM tdretardos;"
77
78     #PASO 3 ---> Ejecutar la consulta SQL
79     cursor.execute(sql, ) #Ejecutar Insert/Update/Delete
80     codigo_conexion = conexion.commit() #Confirmar
81     resultado = cursor.fetchall() #fetchall() ---> Retorna varios registros
82
83     #PASO 4 ---> Cerrar la conexión con la BD
84     cursor.close()
85     conexion.close()
86
87     #PASO 5 ---> Retornar resultado
88     return resultado
89
90
91 #3.4.- Obtener (---> Get) la fecha del servidor en formato
92 # Lunes...(Dia_semana), dd/Mes_texto/Año
93 def get_DS_DD_MesTexto_Year_server():
94     #PASO 1 ---> Establecer la conexión con la base de datos
95     conexion = cadenaConexion()
96     cursor = conexion.cursor()
97
98     #PASO 2 ---> Construir la consulta SQL
99     sql ="SELECT ffecha_server_texto();"
100
101     #PASO 3 ---> Ejecutar la consulta SQL
102     cursor.execute(sql, ) #Ejecutar Insert/Update/Delete
103     codigo_conexion = conexion.commit() #Confirmar
104     resultado = cursor.fetchall() #fetchall() ---> Retorna varios registros
105
106     #PASO 4 ---> Cerrar la conexión con la BD
107     cursor.close()
108     conexion.close()
109
110     #PASO 5 ---> Retornar resultado
111     return resultado
112
113

```

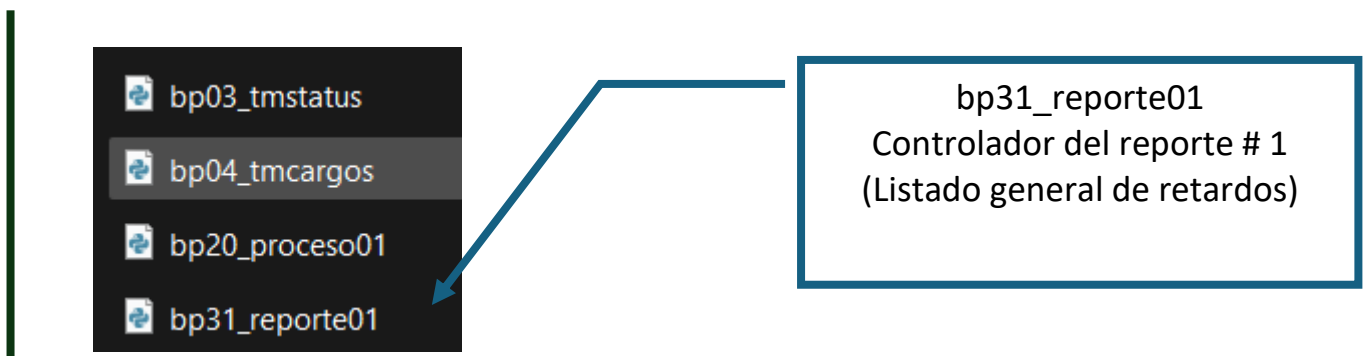
```

113
114 #3.5.- Optener (---> Get) la hora del servidor en formato HH:MM AmPm
115 def get_hora_server_AmPm():
116     #PASO 1 ---> Establecer la conexión con la base de datos
117     conexion = cadenaConexion()
118     cursor = conexion.cursor()
119
120     #PASO 2 ---> Construir la consulta SQL
121     sql ="SELECT fhora_server_ampm);"
122
123     #PASO 3 ---> Ejecutar la consulta SQL
124     cursor.execute(sql, ) #Ejecutar Insert/Update/Delete
125     codigo_conexion = conexion.commit() #Confirmar
126     resultado = cursor.fetchall() #fetchall() ---> Retorna varios registros
127
128     #PASO 4 ---> Cerrar la conexión con la BD
129     cursor.close()
130     conexion.close()
131
132     #PASO 5 ---> Retornar resultado
133     return resultado
134
135 #.....
136 #.....
137
138

```

2.- Controlador

Blueprint



2.1.- Programa ---→ modelo20_tdretardos.py

```
1 #Programa: bp31_reporte01.py
2 #Objetivo: Controlador para el reporte #1 ---> Listado general de retardos en PDF
3 #Autor: Sergio Capacho / Fecha: Jueves 14/Mayo/2026 - 06:28 Pm
4
5
6 #1.- IMPORTAR LAS LIBRERIAS DE TRABAJO
7 #.....
8 #.....
9 from flask import Blueprint, render_template, redirect, url_for, request, Flask
10 from flask import session
11 from psycopg2 import connect
12
13
14 #2.- IMPORTAR LOS MODELOS Y LA CONEXIÓN
15 #.....
16 #.....
17 #2.1.- Importar la BD y su conexión
18 from modelos.modelo00_bd import cadenaConexion #Trae la conexión con la BD
19
20 #2.2.- Importar desde el modelo de la TDRETARDADOS las funciones para
21 # para traer el listado de todos los retardos desde una vista
22 from modelos.modelo20_tdretardos import (
23     get_all_vretardos,
24     get_total_retardo,
25     get_DS_DD_MesTexto_Year_server, #Retorna Lunes..., DD/Mes_texto/AAAA
26     get_hora_server_AmPm #Retorna la hora en formato HH:MM AmPm
27 )
28
29
30 #3.- CREAR DEL OBJETO reporte01
31 #.....
32 #.....
33 obj_reporte01 = Blueprint('reporte01', __name__)
34
35
36 #4.- REGISTRAR LAS RUTAS DEL OBJETO reporte01
37 #.....
38 #.....
39
40 #4.2.- RUTA ---> /reporte01/r01
41 @obj_reporte01.route('/r01')
42 def r01():
43     if ((session["xlogin"] == True) and ( session["xnivel"] == 4)):
44         xmensaje = " "
45         xdatos = get_all_vretardos()
46         return render_template('reporte01/r01.html', ydatos = xdatos)
47     else:
48         return redirect('/login/entrar')
49
50
```

```

50
51 #4.4.- RUTA ---> /reporte01/r01_pdf
52 @obj_reporte01.route('/r01_pdf')
53 def r01_pdf():
54     if ((session["xlogin"] == True) and ( session["xnivel"] == 4)):
55         #URL ---> https://pypi.org/project/fpdf/
56         #(jardin) c:/jardin>pip install fpdf
57
58         #URL ---> https://pypi.org/project/fpdf2/ (Versión Fpdf2)
59         #FUENTE
60         #https://pdforge.com/blog/generate-custom-pdfs-in-python-using-pyfpdf
61         #https://py-pdf.github.io/fpdf2/Tutorial-es.html
62
63         from fpdf import FPDF
64         import os
65
66         # datos para usar
67         datos = get_all_vretardos()
68         xtotal = get_total_retardo()[0]
69         xfecha_server = get_DS_DD_MesTexto_Year_server()[0] #Fecha en Lunes..., DD/MesTexto/Año
70         xhora_server = get_hora_server_AmPm()[0] # HH:MM AmPm
71         xr = 0 #xr ---> Contador de Registros Impresos de la vista o consulta
72
73         Numero_Filas_Imprimir = 23
74
75         if (xtotal % Numero_Filas_Imprimir) == 0:
76             tpaginas = xtotal // Numero_Filas_Imprimir
77         else:
78             tpaginas = (xtotal // Numero_Filas_Imprimir) + 1
79
80         '''
81         Orientación: P : portrait (vertical) / L : landscape (horizontal)
82         unit: Unidad de medida mm
83         Formast: A4 : 210x297mm / Letter : 220 x 280 mm(Tipo de hoja)
84         '''
85         pdf = FPDF(orientation = 'P', unit = 'mm', format='Letter') #Configurar la página
86         pdf.add_page() #Agrega una página
87
88
89         # Title
90         #pdf.cell(w = 0, h = 15, txt = 'Listado de Retardos', border = 1, ln=1, align = 'C', fill = 0)
91
92

```

```

92
93
94     p = 0
95     while (p < tpaginas):
96         # Encabezado
97         p = p + 1
98         pdf.set_font('Arial', 'B', 25) # Arial bold 25
99         pdf.image('static/img/Logo_03.png', x = 10, y = 7, w = 15, h = 15)
100        pdf.image('static/img/Logo_02.png', x = 180, y = 7, w = 20, h = 15)
101        pdf.multi_cell(w=0, h= 10, txt = 'Listado de Retardos', border = 0, align = 'C', fill = 0)
102        pdf.ln(5)
103
104        # encabezado pdf.cell(x=70, y= 20, txt = 'Listado de Retardos')
105        # Border --> 0(nada) / 1(Si hay) / T (top) / B (Bottom) / L (Left) / R (Right)
106        pdf.set_font('Arial', 'B', 16)
107        pdf.cell(w = 15, h = 10, txt = '#', border = 1, align = 'C', fill = 0)
108        pdf.cell(w = 25, h = 10, txt = 'Cédula', border = 1, align = 'C', fill = 0)
109        pdf.cell(w = 40, h = 10, txt = 'Nombre', border = 1, align = 'C', fill = 0)
110        pdf.cell(w = 50, h = 10, txt = 'Excusa', border = 1, align = 'C', fill = 0)
111        pdf.cell(w = 30, h = 10, txt = 'Fecha', border = 1, align = 'C', fill = 0)
112        pdf.multi_cell(w = 0, h = 10, txt = 'Hora', border = 1, align = 'C', fill = 0)
113
114        #Listado de los retardos
115        pdf.set_font('Arial', '', 10)
116        f = 1 #f ---> Numero de Filas impresas en la presente página
117        while( (f <= Numero_Filas_Imprimir) and (xr < xtotal) ):
118            pdf.cell(w = 15, h = 9, txt = str(datos[xr][0]), border = 1, align = 'C', fill = 0)
119            pdf.cell(w = 25, h = 9, txt = str(datos[xr][1]), border = 1, align = 'C', fill = 0)
120            pdf.cell(w = 40, h = 9, txt = str(datos[xr][2]), border = 1, align = 'C', fill = 0)
121            pdf.cell(w = 50, h = 9, txt = str(datos[xr][4]), border = 1, align = 'C', fill = 0)
122            pdf.cell(w = 30, h = 9, txt = str(datos[xr][5]), border = 1, align = 'C', fill = 0)
123            pdf.multi_cell(w = 0, h = 9, txt = str(datos[xr][6]), border = 1, align = 'C', fill = 0)
124            f = f + 1
125            xr = xr + 1 #xr ---> Contador de Registros Impresos de la vista o consulta
126
127        pdf.ln(5)
128        pdf.set_font('Arial', 'B', 10)
129        pdf.cell(w = 70, h = 10, txt = 'Fuente: Systemcap.Net', border = 'T', align = 'L', fill = 0)
130        pdf.cell(w = 60, h = 10, txt = xfecha_server + ' - ' + xhora_server, border = 'T',
131                align = 'C', fill = 0)
132        pdf.multi_cell(w = 0, h = 10, txt = 'Página:' + str(p) + ' de ' + str(tpaginas),
133                border = 'T', align = 'R', fill = 0)
134
135        if (p < tpaginas):
136            pdf.add_page()
137
138
139        # 1. Obtiene la ruta de la carpeta de usuario (Home Directory): /home/usuario o C:\Users\usuario
140        ruta_usuario = os.path.expanduser('~')
141        #print(ruta_usuario)
142

```

```

138
139 # 1. Obtiene la ruta de la carpeta de usuario (Home Directory): /home/usuario o |:\Users\usuario
140 ruta_usuario = os.path.expanduser('~')
141 #print(ruta_usuario)
142
143 # 2. Construye la ruta completa al archivo en la carpeta 'Downloads'/'Descargas'
144 # funciona incluso si la interfaz está en español.
145 nombre_archivo = 'reporte.pdf'
146
147 #3.- Concatena Ruta_usuario + /Downloads + nombre_archivo
148 ruta_descargas = os.path.join(ruta_usuario, 'Downloads', nombre_archivo)
149 #print(f"El archivo se guardará en: {ruta_descargas}")
150
151 #4.- Ejecuta la salida con la ruta completa y el destino 'F' (File/Archivo)
152 pdf.output(ruta_descargas, 'F')
153
154 return redirect('/login/admin')
155 else:
156 return redirect('/login/entrar')
157
158
159 #.....
160
161

```

Elementos importantes

Instalar el paquete fpdf

```

(pjardin) C:\jardin>pip list
Package      Version
-----
blinker      1.9.0
click        8.3.0
colorama     0.4.6
Flask        3.1.2
itsdangerous 2.2.0
Jinja2       3.1.6
MarkupSafe   3.0.3
pip          25.2
psycogp2     2.9.11
Werkzeug     3.1.3

(pjardin) C:\jardin>pip install fpdf
Collecting fpdf
  Using cached fpdf-1.7.2-py2.py3-none-any.whl
Installing collected packages: fpdf
Successfully installed fpdf-1.7.2

```

El título del reporte

Crear una variable objeto de tipo FPDF

```
...  
Orientación: P : portrait (vertical) / L : landscape (horizontal)  
unit: Unidad de medida mm  
Formast: A4 : 210x297mm / Letter : 220 x 280 mm(Tipo de hoja)  
...  
pdf = FPDF(orientation = 'P', unit = 'mm', format='Letter') #Configurar la página  
pdf.add_page() #Agrega una página
```

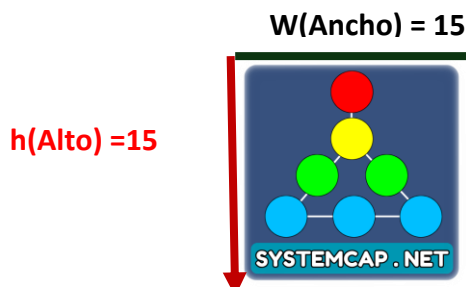
Se agrega una página para "escribir sobre ella"

```
pdf.set_font('Arial', 'B', 25) # Arial bold 25
```

Establece que el tipo de letra a usar en el reporte es: Arial, en Negrita (Bold) y tamaño 25 puntos

Insertar una imagen

```
pdf.image('static/img/Logo_03.png', x = 10, y = 7, w = 15, h = 15)  
pdf.image('static/img/Logo_02.png', x = 180, y = 7, w = 20, h = 15)
```



Define el ancho (W width - ancho) y el alto(h height - alto) de la imagen



```
pdf.multi_cell(w=0, h= 10, txt = 'Listado de Retardos', border = 0, align = 'C', fill = 0)
```



Listado de Retardos



```
pdf.multi_cell(w=0, h= 10, txt ='Listado de Retardos', border = 0, align = 'C', fill = 0)
```

pdf.multi_cell() ---→ Construye una celda con múltiples celdas.

w (width - ancho): Determina el ancho de la celda; Si pones **0**, la celda se extiende de izquierda a derecha de la página. Es la forma más fácil de decir "usa todo el espacio disponible".

h (height – alto): El alto de la celda.

txt: El texto a imprimir dentro de la celda.

border: Define si quieres un borde alrededor de la celda.

0(nada) / **1**(Si hay) / **T** (top) / **B** (Bottom) / **L** (Left) / **R** (Right)

align: Determina la alineación del texto dentro de la celda. '**L**': Izquierda. / '**C**': Centro. / '**R**': Derecha.

fill: Booleano (True **1**/False **0**). Si es verdadero, pinta el fondo de la celda (usando el color definido previamente con `set_fill_color()`).

Nota: Con `multi_cell`: Si el texto llega al borde definido por **W(Width – Ancho)**, FPDF hace un "Enter" automático y continúa en la línea de abajo. Además, después de ejecutar un `multi_cell`, el cursor se coloca automáticamente al principio de la siguiente línea (debajo de la celda). No necesitas agregar un `pdf.ln()` después, ya que `multi_cell` lo hace por ti por defecto. A menos, que UD desea agregar más espacios, como es nuestro caso



```
pdf.ln(5)
```

#	(
122	

Agrega 5 mm de separación

El encabezado del reporte

#	Cédula	Nombre	Excusa	Fecha	Hora
---	--------	--------	--------	-------	------



```
111 pdf.set_font('Arial', 'B', 16)
112 pdf.cell(w = 15, h = 10, txt = '#', border = 1, align = 'C', fill = 0)
113 pdf.cell(w = 25, h = 10, txt = 'Cédula', border = 1, align = 'C', fill = 0)
114 pdf.cell(w = 40, h = 10, txt = 'Nombre', border = 1, align = 'C', fill = 0)
115 pdf.cell(w = 50, h = 10, txt = 'Excusa', border = 1, align = 'C', fill = 0)
116 pdf.cell(w = 30, h = 10, txt = 'Fecha', border = 1, align = 'C', fill = 0)
117 pdf.multi_cell(w = 0, h = 10, txt = 'Hora', border = 1, align = 'C', fill = 0)
```

pdf.cell() ---→ Básicamente, crea un rectángulo (una celda) donde puedes colocar texto, bordes y colores de fondo. En nuestro caso una celda para el signo de “#”, otra celda para el texto “Cédula”, otra celda para la descripción de la “Excusa”, y así sucesivamente. En resúmenes se construye el encabezado del reporte.

La sintaxis básica es la siguiente:

`pdf.cell(w=0, h=0, txt="", border=0, ln=0, align="", fill=False, link="")`

w (width - ancho): El ancho de la celda. Si pones 0, la celda se extiende automáticamente hasta el margen derecho de la página.

h (height – alto): El alto de la celda.

txt: El texto a imprimir dentro de la celda.

border: Define si quieres un borde alrededor de la celda.

O combinaciones de letras: **0**(nada) / **L** (izquierda) / **T** (arriba) / **R** (derecha) / **B** (abajo).

ln (salto de línea): Indica dónde se posiciona el cursor después de llamar a la celda.

- 0: A la derecha (para poner otra celda al lado).
- 1: Al inicio de la siguiente línea.
- 2: Debajo de la celda actual.

align: Determina la alineación del texto dentro de la celda. **'L'**: Izquierda. / **'C'**: Centro. / **'R'**: Derecha.

fill: Booleano (True **1**/False **0**). Si es verdadero, pinta el fondo de la celda (usando el color definido previamente con `set_fill_color()`).

Programa Ejemplo:

```
from fpdf import FPDF
```

```
pdf = FPDF()
```

```
pdf.add_page()
```

```
pdf.set_font("Arial", size=12)
```

```
# 1. Una celda simple con borde (Título centrado)
```

```
pdf.cell(200, 10, txt="Mi Reporte de Ventas", ln=1, align='C', border=1)
```

```
# 2. Celdas una al lado de la otra (como una tabla)
```

```
pdf.cell(40, 10, txt="Nombre", border=1, ln=0)
```

```
pdf.cell(40, 10, txt="Puntaje", border=1, ln=1) # Aquí ln=1 para bajar de línea
```

```
# 3. Celda con fondo de color
```

```
pdf.set_fill_color(200, 220, 255)
```

```
pdf.cell(80, 10, txt="Celda con color", border=1, ln=1, fill=True)
```

```
pdf.output("ejemplo.pdf")
```

La página

Este bloque de programa imprime cada página del reporte, donde:

- Se controla cantidad de filas (**f**) que se imprimen por página
(Mientras variable **f** < variable **Numero_filas_Imprimir** (23)).
- Se controla la cantidad de registros impresos (**xr**) del total (**xtotal**) de registros a imprimir

122	3000	MARTHA	MUCHO TRAFICO	22Nov2021	08:41 Am
109	3000	MARTHA	MUCHO TRAFICO	22Nov2021	08:41 Am
99	1000	ROBERTO JAIMES	ME QUEDE DORMIDO	01Ene2022	08:20 Am
82	6000	NATASHA	PELJE CON EL AGUILA	05Ene2022	08:32 Am
63	3000	MARTHA	MUCHO TRAFICO	05Ene2022	08:40 Am
64	2000	ZILA CONTRERAS	EL CARRO NO ME QUIERE	06Ene2022	09:00 Am
65	3000	MARTHA	NO ME BUSCARON	06Ene2022	09:10 Am
19	1000	ROBERTO JAIMES	ME QUEDE DORMIDO	07Ene2022	10:43 Am
20	3000	MARTHA	NO ME BUSCARON	07Ene2022	10:50 Am
21	4000	CARLOS	EL CARRO NO ME QUIERE	10Ene2022	09:00 Am
66	5000	MATIAS	EL CARRO NO ME QUIERE	11Ene2022	08:17 Am
22	5000	MATIAS	MUCHA LLUVIA	12Ene2022	08:20 Am
67	7000	PEGGY CARTER	NO ME BUSCARON	14Ene2022	08:32 Am
23	6000	NATASHA	MUCHO TRAFICO	15Ene2022	09:14 Am
68	3000	MARTHA	MUCHO TRAFICO	17Ene2022	09:47 Am
69	6000	NATASHA	SALI TARDE DE LA CASA	21Ene2022	08:24 Am
71	3000	MARTHA	MUCHA LLUVIA	23Ene2022	03:20 Am
72	2000	ZILA CONTRERAS	EL CARRO NO ME QUIERE	23Ene2022	08:15 Am
70	4000	CARLOS	MUCHO TRAFICO	23Ene2022	08:37 Am
73	1000	ROBERTO JAIMES	MUCHA LLUVIA	25Ene2022	08:35 Am
74	6000	NATASHA	NO ME BUSCARON	25Ene2022	08:38 Am
75	4000	CARLOS	MUCHO TRAFICO	27Ene2022	09:00 Am
35	3000	MARTHA	NO ME BUSCARON	01Feb2022	08:40 Am

```
#Listado de los retardos
pdf.set_font('Arial', '', 10)
f = 1 #f ---> Numero de Filas impresas en la presente página
while( (f <= Numero_Filas_Imprimir) and (xr < xtotal) ):
    pdf.cell(w = 15, h = 9, txt = str(datos[xr][0]), border = 1, align = 'C', fill = 0)
    pdf.cell(w = 25, h = 9, txt = str(datos[xr][1]), border = 1, align = 'C', fill = 0)
    pdf.cell(w = 40, h = 9, txt = str(datos[xr][2]), border = 1, align = 'C', fill = 0)
    pdf.cell(w = 50, h = 9, txt = str(datos[xr][4]), border = 1, align = 'C', fill = 0)
    pdf.cell(w = 30, h = 9, txt = str(datos[xr][5]), border = 1, align = 'C', fill = 0)
    pdf.multi_cell(w = 0, h = 9, txt = str(datos[xr][6]), border = 1, align = 'C', fill = 0)
    f = f + 1
    xr = xr + 1 #xr ---> Contador de Registros Impresos de la vista o consulta
```

Muestra el contenido de la variable

datos = get_all_vretardos()

str(datos[xr][0]) -> Número del retardo

str(datos[xr][1]) -> Cédula

str(datos[xr][2]) -> Nombre

str(datos[xr][4]) -> Descripción de la excusa

str(datos[xr][5]) -> Hora del retardo

str(datos[xr][6]) -> Fecha del Retardo

xr -> Es el contador que permite recorrer el arreglo de los datos. NOTA: El arreglo inicia en la posición 0

El pie de página

```
pdf.ln(5)
pdf.set_font('Arial', 'B', 10)
pdf.cell(w = 70, h = 10, txt = 'Fuente: Systemcap.Net', border = 'T', align = 'L', fill = 0)
pdf.cell(w = 60, h = 10, txt = xfecha_server + ' - ' + xhora_server, border = 'T',
        align = 'C', fill = 0)
pdf.multi_cell(w = 0, h = 10, txt = 'Página:' + str(p) + ' de ' + str(tpaginas),
              border = 'T', align = 'R', fill = 0)
```

Este bloque de programa construye los mensajes:

- Fuente: Systemcap.Net
- Fecha y hora del servidor -> xfecha_server + ' - ' + xhora_server
- La numeración de la página: 'Página:' + str(p) + ' de ' + str(tpaginas)

Control de páginas a imprimir

Primero: se definen cuantas páginas se van a imprimir

Segundo: Construir el ciclo de validación de cantidad de páginas impresas
(Mientras $p < tpaginas$)


Código que construye cada página

Tercero: Preguntar si existe otra página para imprimir


```
# datos para usar
datos = get_all_vretardos()
xtotal = get_total_retardo()[0]
xfecha_server = get_DS_DD_MesTexto_Year_server()[0] #Fecha en Lunes..., DD/Mestexto/Año
xhora_server = get_hora_server_AmPm()[0]           # HH:MM AmPm
xr = 0 #xr ---> Contador de Registros Impresos de la vista o consulta

Numero_Filas_Imprimir = 23


if (xtotal % Numero_Filas_Imprimir) == 0:
    tpaginas = xtotal // Numero_Filas_Imprimir
else:
    tpaginas = (xtotal // Numero_Filas_Imprimir) + 1
```



```
p = 0
while (p < tpaginas):
    # Encabezado
    p = p + 1
    pdf.set_font('Arial', 'B', 25) # Arial bold 25
    pdf.image('static/img/Logo_03.png', x = 10, y = 7, w = 15, h = 15)
    pdf.image('static/img/Logo_02.png', x = 180, y = 7, w = 20, h = 15)
    pdf.multi_cell(w=0, h= 10, txt = 'Listado de Retardos', border = 0, align = 'C', fill = 0)
    pdf.ln(5)
```



```
if (p < tpaginas):
    pdf.add_page()
```



Generar el pdf en la carpeta Dowloads

```
133         if (p < tpaginas):
134             pdf.add_page()
135
136
137         # 1. Obtiene la ruta de la carpeta de usuario (Home Directory): /home/usuario o :\Users\usuario
138         ruta_usuario = os.path.expanduser('~')
139         #print(ruta_usuario)
140
141         # 2. Construye la ruta completa al archivo en la carpeta 'Downloads'/'Descargas'
142         #     funciona incluso si la interfaz está en español.
143         nombre_archivo = 'reporte.pdf'
144
145         #3.- Concatena Ruta_usuario + /Downloads + nombre_archivo
146         ruta_descargas = os.path.join(ruta_usuario, 'Downloads', nombre_archivo)
147         #print(f"El archivo se guardará en: {ruta_descargas}")
148
149         #4.- Ejecuta la salida con la ruta completa y el destino 'F' (File/Archivo)
150         pdf.output(ruta_descargas, 'F')
151
152         return redirect('/login/admin')
153     else:
154         return redirect('/login/entrar')
155
156
157     #.....
158
```

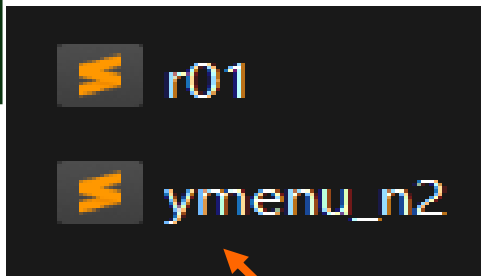
3.- Vista

r01.html



Templates

- M01_sitio
- M02_login
- M03_TMStatus
- M04_TMCargos
- proceso01
- Reporte01



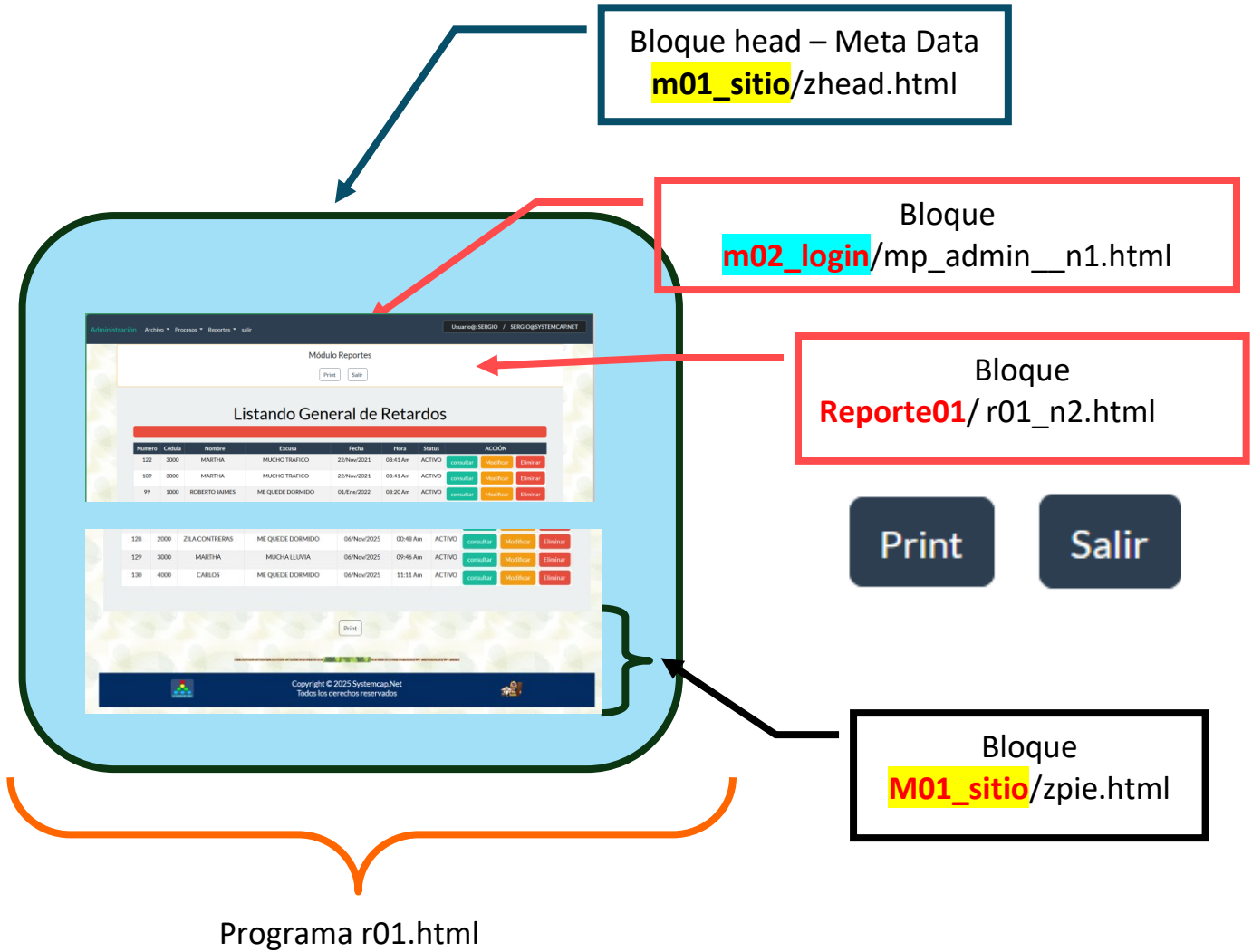
Botón para generar el reporte en formato PDF

Botón para salir del reporte

Se deben codificar los siguientes programas dentro de la carpeta reporte01

3.1.- r01.html

A.- Diseño de la página



B.- Programa --> r01.html

```
1 <!-- Programa: r01.html -->
2 <!-- Autor: Sergio Capacho / Jueves 14/Mayo/2026 09]:23 Pm -->
3 <!-- Objetivo: Reporte #1 Para generar un listyado general de los retardos -->
4 <!DOCTYPE html>
5 <html lang="es">
6 <!-- Inicio del Bloque Head -->
7 {% include 'm01_sitio/zhead.html' %}
8 <!-- Inicio del Bloque Head -->
9
10 <!-- Inicio del Body -->
11 <body >
12 <!-- Inicio del Bloque Menu -->
13 {% include 'm02_login/mp_admin_zmenu_n1.html' %}
14 <!-- Inicio del Bloque Menu -->
15
16 <!-- Inicio del Bloque Menu -->
17 {% include 'reporte01/ymenu_n2.html' %}
18 <!-- Inicio del Bloque Menu -->
19
20 <!--Inicio del Bloque Centro / Container Principal -->
21 <div class="container text-center">
22 <div class="mb-0 p-5 bg-light">
23
24 <h1 style="color:black">Listando General de Retardos</h1>
25
26 {% if ymensaje != " " %}
27 <div class="card-footer text-center">
28 <div class="alert alert-danger" role="alert">
29 <strong> {{ ymensaje }} </strong>
30 </div>
31 </div>
32 {% endif %}
33
34
35 <table class="table table-striped table-bordered bg-white table-sm">
36 <thead>
37 <tr class="table-primary">
38 <th scope="col">
39 Numero
40 </th>
41 <th scope="col">
42 Cédula
43 </th>
44 <th scope="col">Nombre</th>
45 <th scope="col">Excusa</th>
46 <th scope="col">Fecha</th>
47 <th scope="col">Hora</th>
48 <th scope="col">Status</th>
49 <th scope="col" colspan="3">
50 ACCIÓN
51 </th>
52 </tr>
53 </thead>
54 <tbody>
55 {% for item in ydatos %}
```

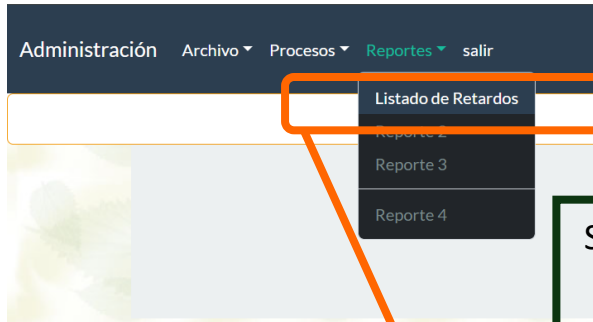
```

54 <tbody>
55   {% for item in ydatos %}
56     <tr>
57       <td align="text-center">      {{ item[0] }} </td>
58       <td align="center">          {{ item[1] }} </td>
59       <td align="center">          {{ item[2] }} &nbsp; &nbsp; &nbsp; </td>
60       <td align="center">          {{ item[4] }} &nbsp; &nbsp; &nbsp; </td>
61       <td align="center">          {{ item[5] }} &nbsp; &nbsp; &nbsp; </td>
62       <td align="center">          {{ item[6] }} &nbsp; &nbsp; &nbsp; </td>
63     <td>
64       {% if item[7] == 0 %}
65         <font color="#FF0000">
66           <b>{{ item[8] }}</b>
67         </font>
68       {%else%}
69         {{ item[8] }}
70       {%endif%}
71     </td>
72
73     <td width ="75px">
74       <!-- -->
75       <form action="#" method="POST">
76         <input type="text" id="txt_cod" name="txt_cod"
77           value = "{ { item[0] } }" hidden/>
78
79         <button type="submit" value="Submit"
80           class="btn btn-success"
81           title="Click para realizar la consulta...">
82           consultar
83         </button>
84       </form>
85     </td>
86
87     <td width ="75px">
88       <!-- -->
89       <form action="#" method="POST">
90         <input type="text" id="txt_cod" name="txt_cod"
91           value = "{ { item[0] } }" hidden/>
92
93         <button type="submit" value="Submit"
94           class="btn btn-warning"
95           title="Click para realizar la Modificación...">
96           Modificar
97         </button>
98       </form>
99     </td>

```


3.2.- Agregar la ruta /reporte01/r01

m02_login/mp_admin_zmenu_admin_n1.html



Se debe agregar la ruta “/reporte01/r01” dentro del código del programa
m02_login/mp_admin_zmenu_admin_n1.html

```
<li class="nav-item dropdown">
  <a class="nav-link dropdown-toggle" data-bs-toggle="dropdown" href="#" role="button"
  aria-haspopup="true" aria-expanded="false">Reportes</a>
  <div class="dropdown-menu">
    <a class="dropdown-item" href="/reportes/r01">
      Listado de Retardos
    </a>
    <a class="dropdown-item" href="#">Reporte 2</a>
    <a class="dropdown-item" href="#">Reporte 3</a>
    <div class="dropdown-divider"></div>
    <a class="dropdown-item" href="#">Reporte 4</a>
  </div>
</li>
```

3.3.- En el programa App.py

A.- Agregar el objeto reporte01 (**obj_reporte01**) desde el controlador bp31_reporte01

```
#02.06 ---> Importación el objeto PROCESO01
#           ( Controlador de pa el proceso: Registrar Retardos )
#           que está en la carpeta blueprints y el archivo bp20_proceso01
from blueprints.bp20_proceso01 import obj_proceso01
```

```
#02.07 ---> Importación el objeto reportes
#           ( Controlador para generar los reportes de los retardos )
#           que está en la carpeta blueprints y el archivo bp30_reportes
from blueprints.bp31_reporte01 import obj_reporte01
```

B.- Registrar la ruta /reporte01 con base al objeto reporte01 (**obj_reporte01**)

```
#05.08.- RUTA DEL MÓDULO PROCESO # 01 ---> /proceso01
mi_app.register_blueprint(obj_proceso01, url_prefix='/proceso01')
```

```
#05.08.- RUTA MÓDULO Reporte # 01 ---> /reportes
mi_app.register_blueprint(obj_reporte01, url_prefix='/reporte01')
```

4.- Ejecutar el proyecto

```
(pjardin) C:\jardin>py app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in production.
I server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 491-858-376
```