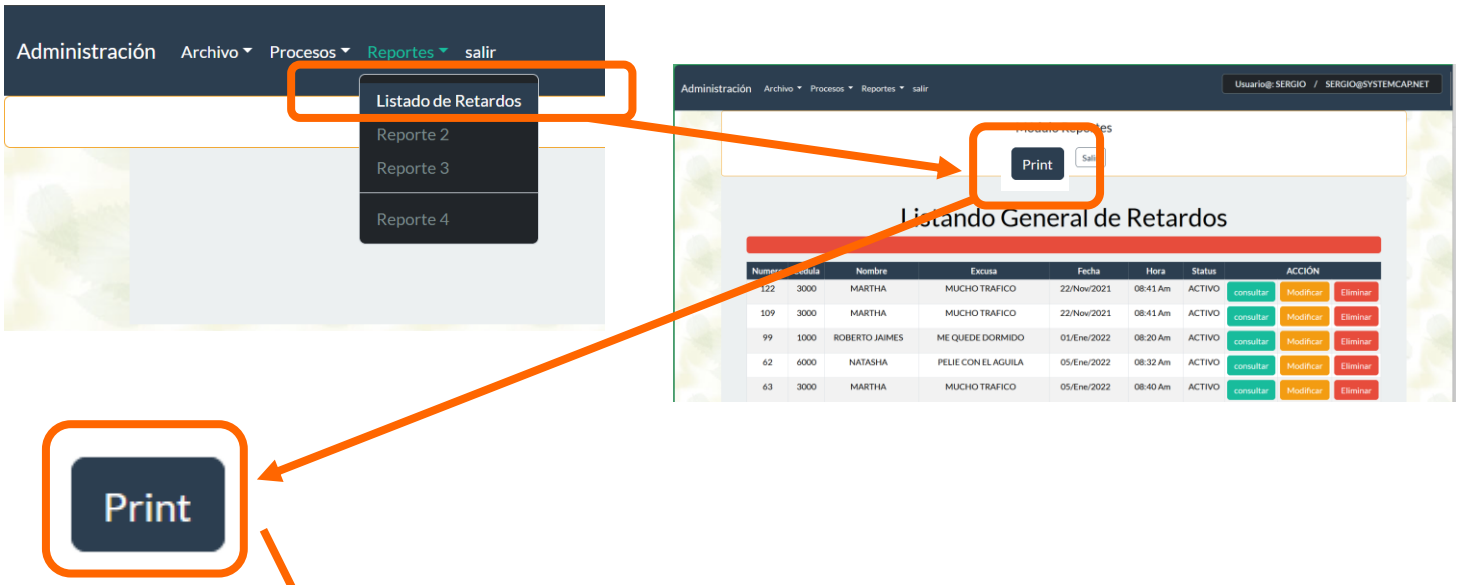
# Reporte01
# Listado general de retardos







## 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 |

**Fuente: Systemcap.Net** **Miercoles, 13/May/2026 - 05:56 Pm** **Página:6 de 6**

# 1.- Modelo

## 1.1.- Programa ---→ modelo20_tdretardos.py

```python
#Programa: modelo20_tdretardos.py
#Objetivo: Funciones para trabajar con la TDRETARDOS
#Autor: Sergio Capacho / Fecha: Martes 14/Mayo/2026 - 05:51 Pm

#1.- IMPORTAR DE LIBRERIAS
#.....................................................
#.....................................................................................
from flask import  Blueprint, render_template, flash, redirect, url_for, request, Flask
from flask import  session
from psycopg2 import connect


#2.- IMPORTAR EL MODELO Y LA CONEXIÓN CON LA BD
#.....................................................
#.....................................................................................

#2.1.- Importación Base de Datos
from modelos.modelo00_bd import cadenaConexion


#3.- OPERACIONES SQL CON TDRETARDOS
#.....................................................
#.....................................................................................

#3.1.- Registrar un retardo nuevo
def add_retardo(xfkcedemple, xfkcodexcu, xfecha, xhora):...


#3.2.- Obtener todos los registros de la Vista retardados00 (***)
def get_all_vretardos():
    #PASO 1 ---> Establecer la conexión con la BD
    conexion = cadenaConexion()
    cursor = conexion.cursor()

    #PASO 2 ---> Construir la consulta SQL
    sql ="select * from vretardos00;"

    #PASO 3 ---> Ejecutar la consulta SQL
    cursor.execute(sql,  )              #Ejecutar Insert/Update/Delete
    codigo_conexion = conexion.commit() #Confirmar
    resultado = cursor.fetchall()       #fetchone() ---> Un solo registro / fetchall()  ---> Varios

    #PASO 4 ---> Cerrar la conexión con la BD
    cursor.close()
    conexion.close()

    #PASO 5 ---> Retornar resultado
    return resultado


```

**Este código ya fue analizado en clase**

```python

#3.3.- Determinar el total de retardos (Get ---> Obtener)
def get_total_retardo():
    #PASO 1 ---> Establecer la conexión con la base de datos
    conexion = cadenaConexion()
    cursor = conexion.cursor()

    #PASO 2 ---> Construir la consulta SQL
    sql ="SELECT COUNT(pknr) FROM tdretardos;"

    #PASO 3 ---> Ejecutar la consulta SQL
    cursor.execute(sql,  )              #Ejecutar Insert/Update/Delete
    codigo_conexion = conexion.commit() #Confirmar
    resultado = cursor.fetchone()       #fetchall() ---> Retorna varios registros

    #PASO 4 ---> Cerrar la conexión con la BD
    cursor.close()
    conexion.close()

    #PASO 5 ---> Retornar resultado
    return resultado


#3.4.- Optener (---> Get) la fecha del servidor en formato
#                   Lunes...(Dia_semana), dd/Mes_texto/Año
def get_DS_DD_MesTexto_Year_server():
    #PASO 1 ---> Establecer la conexión con la base de datos
    conexion = cadenaConexion()
    cursor = conexion.cursor()

    #PASO 2 ---> Construir la consulta SQL
    sql ="SELECT ffecha_server_texto();"

    #PASO 3 ---> Ejecutar la consulta SQL
    cursor.execute(sql,  )              #Ejecutar Insert/Update/Delete
    codigo_conexion = conexion.commit() #Confirmar
    resultado = cursor.fetchone()       #fetchall() ---> Retorna varios registros

    #PASO 4 ---> Cerrar la conexión con la BD
    cursor.close()
    conexion.close()

    #PASO 5 ---> Retornar resultado
    return resultado


```

```python

#3.5.- Optener (---> Get) la hora del servidor en formato HH:MM AmPm
def get_hora_server_AmPm():
    #PASO 1 ---> Establecer la conexión con la base de datos
    conexion = cadenaConexion()
    cursor = conexion.cursor()

    #PASO 2 ---> Construir la consulta SQL
    sql ="SELECT fhora_server_ampm();"

    #PASO 3 ---> Ejecutar la consulta SQL
    cursor.execute(sql,  )              #Ejecutar Insert/Update/Delete
    codigo_conexion = conexion.commit() #Confirmar
    resultado = cursor.fetchone()       #fetchall() ---> Retorna varios registros

    #PASO 4 ---> Cerrar la conexión con la BD
    cursor.close()
    conexion.close()

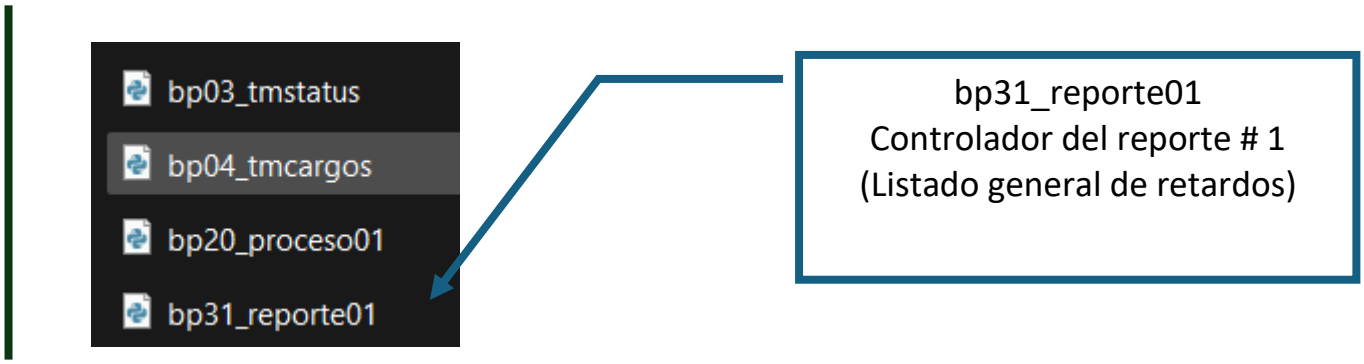
    #PASO 5 ---> Retornar resultado
    return resultado

#.....................................................
#.....................................................................................


```

# 2.- Controlador

## Blueprint



## 2.1.- Programa ---→ modelo20_tdretardos.py

```python
#Programa: bp31_reporte01.py
#Objetivo: Controlador para el reporte #1 ---> Listado general de retardos en PDF
#Autor: Sergio Capacho / Fecha: Jueves 14/Mayo/2026 - 06:28 Pm


#1.- IMPORTAR LAS LIBRERIAS DE TRABAJO
#.....................................................
#.....................................................................................
from flask import  Blueprint, render_template, redirect, url_for, request,  Flask
from flask import  session
from psycopg2 import connect


#2.- IMPORTAR LOS MODELOS Y LA CONEXIÓN
#.....................................................
#.....................................................................................
#2.1.- Importar la BD y su conexión
from modelos.modelo00_bd import cadenaConexion #Trae la conexión con la BD

#2.2.- Importar desde el modelo de la TDRETARDADOS las funciones para
#       para traer el listado de todos los retardos desde una vista
from modelos.modelo20_tdretardos import (
    get_all_vretardos,
    get_total_retardo,
    get_DS_DD_MesTexto_Year_server, #Retorna Lunes..., DD/Mes_texto/AAAA
    get_hora_server_AmPm #Retorna la hora en formato HH:MM AmPm
)


#3.- CREAR DEL OBJETO reporte01
#.....................................................
#.....................................................................................
obj_reporte01 = Blueprint('reporte01', __name__)


#4.- REGISTRAR LAS RUTAS DEL OBJETO reporte01
#.....................................................
#.....................................................................................

#4.2.- RUTA ---> /reporte01/r01
@obj_reporte01.route('/r01')
def r01():
    if ((session["xlogin"] == True) and ( session["xnivel"] == 4)):
        xmensaje = " "
        xdatos = get_all_vretardos()
        return render_template('reporte01/r01.html', ydatos = xdatos)
    else:
        return redirect('/login/entrar')


```

```python

#4.4.- RUTA ---> /reporte01/r01_pdf
@obj_reporte01.route('/r01_pdf')
def r01_pdf():
    if ((session["xlogin"] == True) and ( session["xnivel"] == 4)):
        #URL ---> https://pypi.org/project/fpdf/
        #(jardin) c:/jardin>pip install fpdf

        #URL ---> https://pypi.org/project/fpdf2/  (Versión Fpdf2)
        #FUENTE
        #https://pdforge.com/blog/generate-custom-pdfs-in-python-using-pyfpdf
        #https://py-pdf.github.io/fpdf2/Tutorial-es.html

        from fpdf import FPDF
        import os

        # 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

       '''
           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


       # Title
       #pdf.cell(w = 0, h = 15, txt = 'Listado de Retardos', border = 1, ln=1, align = 'C', fill = 0)


```

```
92
93      p = 0
94      while (p < tpaginas):
95          # Encabezado
96          p = p + 1
97          pdf.set_font('Arial', 'B', 25)  # Arial bold 25
98          pdf.image('static/img/Logo_03.png', x = 10, y = 7, w = 15, h = 15)
99          pdf.image('static/img/Logo_02.png', x = 180, y = 7, w = 20, h = 15)
100         pdf.multi_cell(w=0, h= 10, txt ='Listado de Retardos', border = 0, align = 'C', fill = 0)
101         pdf.ln(5)
102
103         # encabezado  pdf.cell(x=70, y= 20, txt ='Listado de Retardos')
104         # Border --> 0(nada) / 1(Si hay) / T (top) / B (Bottom) / L (Left) / R (Right)
105         pdf.set_font('Arial', 'B', 16)
106         pdf.cell(w = 15, h = 10, txt = '#', border = 1,  align = 'C', fill = 0)
107         pdf.cell(w = 25, h = 10, txt = 'Cédula', border = 1, align = 'C', fill = 0)
108         pdf.cell(w = 40, h = 10, txt = 'Nombre', border = 1, align = 'C', fill = 0)
109         pdf.cell(w = 50, h = 10, txt = 'Excusa', border = 1, align = 'C', fill = 0)
110         pdf.cell(w = 30, h = 10, txt = 'Fecha', border = 1, align = 'C', fill = 0)
111         pdf.multi_cell(w = 0, h = 10, txt = 'Hora', border = 1, align = 'C', fill = 0)
112
113         #Listado de los retardos
114         pdf.set_font('Arial', '', 10)
115         f = 1 #f ---> Numero de Filas impresas en la presente página
116         while( (f <= Numero_Filas_Imprimir) and (xr < xtotal) ):
117             pdf.cell(w = 15, h = 9, txt = str(datos[xr][0]), border = 1, align = 'C', fill = 0)
118             pdf.cell(w = 25, h = 9, txt = str(datos[xr][1]), border = 1, align = 'C', fill = 0)
119             pdf.cell(w = 40, h = 9, txt = str(datos[xr][2]), border = 1, align = 'C', fill = 0)
120             pdf.cell(w = 50, h = 9, txt = str(datos[xr][4]), border = 1, align = 'C', fill = 0)
121             pdf.cell(w = 30, h = 9, txt = str(datos[xr][5]), border = 1, align = 'C', fill = 0)
122             pdf.multi_cell(w = 0, h = 9, txt = str(datos[xr][6]), border = 1, align = 'C', fill = 0)
123             f = f + 1
124             xr = xr + 1    #xr ---> Contador de Registros Impresos de la vista o consulta
125
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
psycopg2     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
```

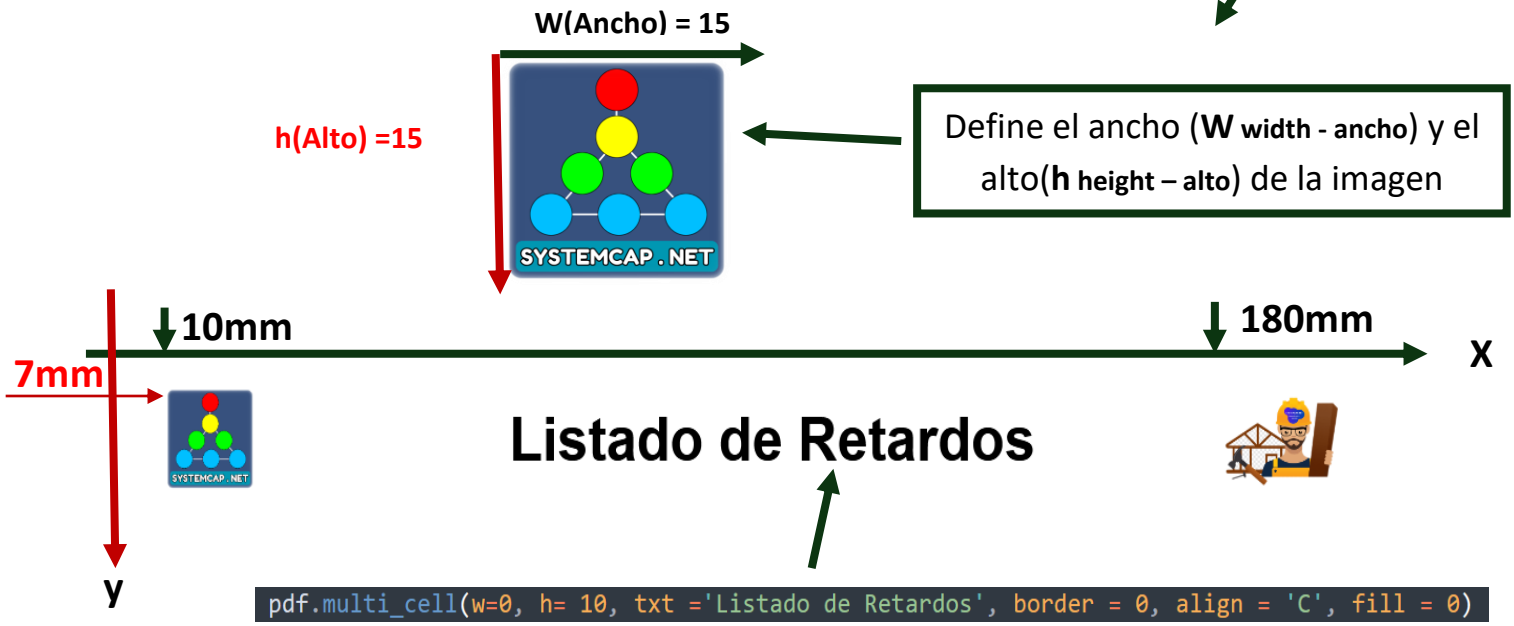
Se agrega una página para “escribir sobre ella”

```
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)
```




```
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)
```

Agrega 5 mm de separación

| # | ( |
|---|---|
| 122 | |

# 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 resumes 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

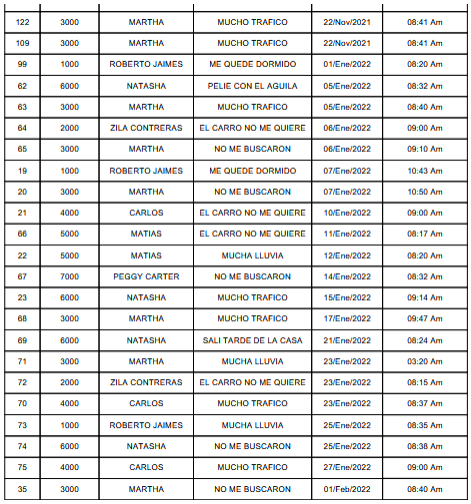
# 3. Celda con fondo de color
pdf.set_fill_color(200, 220, 255)
pdf.cell(80, 10, txt="Celda con color", border=1, ln=1, fill=True)

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 | 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 |
| 65 | 3000 | MARTHA | NO ME BUSCARON | 06/Ene/2022 | 09:10 Am |
| 19 | 1000 | ROBERTO JAIMES | ME QUEDE DORMIDO | 07/Ene/2022 | 10:43 Am |
| 20 | 3000 | MARTHA | NO ME BUSCARON | 07/Ene/2022 | 10:50 Am |
| 21 | 4000 | CARLOS | EL CARRO NO ME QUIERE | 10/Ene/2022 | 09:00 Am |
| 66 | 5000 | MATIAS | EL CARRO NO ME QUIERE | 11/Ene/2022 | 08:17 Am |
| 22 | 5000 | MATIAS | MUCHA LLUVIA | 12/Ene/2022 | 08:20 Am |
| 67 | 7000 | PEGGY CARTER | NO ME BUSCARON | 14/Ene/2022 | 08:32 Am |
| 23 | 6000 | NATASHA | MUCHO TRAFICO | 15/Ene/2022 | 09:14 Am |
| 68 | 3000 | MARTHA | MUCHO TRAFICO | 17/Ene/2022 | 09:47 Am |
| 69 | 6000 | NATASHA | SALI TARDE DE LA CASA | 21/Ene/2022 | 08:24 Am |
| 71 | 3000 | MARTHA | MUCHA LLUVIA | 23/Ene/2022 | 03:20 Am |
| 72 | 2000 | ZILA CONTRERAS | EL CARRO NO ME QUIERE | 23/Ene/2022 | 08:15 Am |
| 70 | 4000 | CARLOS | MUCHO TRAFICO | 23/Ene/2022 | 08:37 Am |
| 73 | 1000 | ROBERTO JAIMES | MUCHA LLUVIA | 25/Ene/2022 | 08:35 Am |
| 74 | 6000 | NATASHA | NO ME BUSCARON | 25/Ene/2022 | 08:38 Am |
| 75 | 4000 | CARLOS | MUCHO TRAFICO | 27/Ene/2022 | 09:00 Am |
| 35 | 3000 | MARTHA | NO ME BUSCARON | 01/Feb/2022 | 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 píe 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 
```

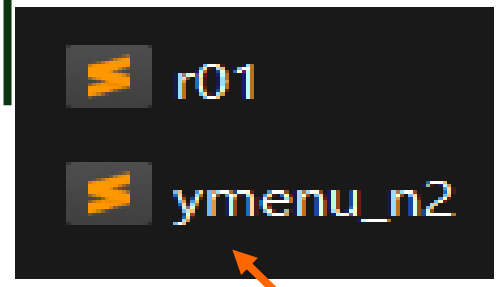
# 3.- Vista

r01.html




**Templates**

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



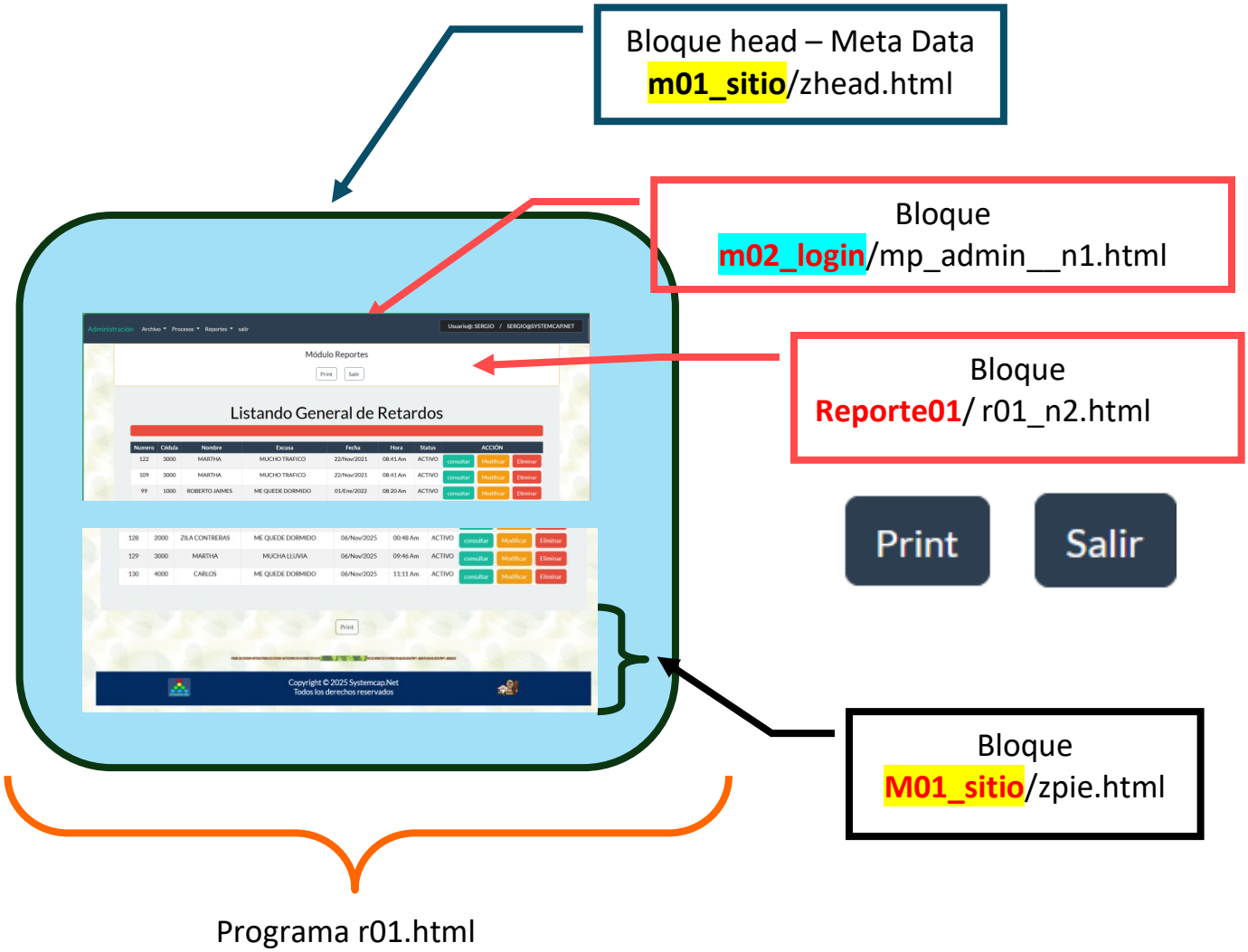

Botón para generar el reporte en formato PDF

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

```
<!-- Programa: r01.html -->
<!-- Autor: Sergio Capacho / Jueves 14/Mayo/2026 09]:23 Pm -->
<!-- Objetivo: Reporte #1 Para generar un listyado general de los retardos -->
<!DOCTYPE html>
<html lang="es">
    <!-- Inicio del Bloque Head -->
    {% include 'm01_sitio/zhead.html' %}
    <!-- Inicio del Bloque Head -->

     <!-- Inicio del Body -->
     <body >
        <!--- Inicio del Bloque Menu -->
        {% include 'm02_login/mp_admin_zmenu_n1.html' %}
        <!--- Inicio del Bloque Menu -->

        <!--- Inicio del Bloque Menu -->
        {% include 'reporte01/ymenu_n2.html' %}
        <!--- Inicio del Bloque Menu -->

        <!--Inicio del Bloque Centro /  Container Principal  -->
        <div class="container text-center">
            <div class="mb-0 p-5 bg-light">

                <h1 style="color:black">Listando General de Retardos</h1>

                {% if ymensaje != " " %}
                    <div class="card-footer text-center">
                        <div class="alert alert-danger" role="alert">
                            <strong> {{ ymensaje }} </strong>
                        </div>
                    </div>
                {% endif %}


                <table class="table table-striped table-bordered bg-white table-sm">
                    <thead>
                        <tr class="table-primary">
                            <th scope="col">
                                Numero
                            </th>
                            <th scope="col">
                                Cédula
                            </th>
                            <th scope="col">Nombre</th>
                            <th scope="col">Excusa</th>
                            <th scope="col">Fecha</th>
                            <th scope="col">Hora</th>
                            <th scope="col">Status</th>
                            <th scope="col" colspan="3">
                                ACCIÓN
                            </th>
                        </tr>
                    </thead>
                    <tbody>
                        {% for item in ydatos %}
```

```
                    <tbody>
                        {% for item in ydatos %}
                            <tr>
                                <td align="text-center">    {{ item[0] }}   </td>
                                <td align="center">         {{ item[1] }}   </td>
                                <td align="center">         {{ item[2] }}         </td>
                                <td align="center">         {{ item[4] }}         </td>
                                <td align="center">         {{ item[5] }}         </td>
                                <td align="center">         {{ item[6] }}         </td>
                                <td>
                                    {% if item[7] == 0 %}
                                        <font color="#FF0000">
                                            <b>{{ item[8] }}</b>
                                        </font>
                                    {%else%}
                                        {{ item[8] }}
                                    {%endif%}
                                </td>


                                <td width ="75px">
                                    <!--  -->
                                    <form action="#" method="POST">
                                        <input type="text" id="txt_cod" name="txt_cod"
                                                value = "{ { item[0] } }" hidden/>

                                        <button type="submit" value="Submit"
                                                class="btn btn-success"
                                                title="Click para realizar la consulta...">
                                                consultar
                                        </button>
                                    </form>
                                </td>


                                <td width ="75px">
                                    <!--  -->
                                    <form action="#" method="POST">
                                        <input type="text" id="txt_cod" name="txt_cod"
                                                value = "{ { item[0]  } }" hidden/>

                                        <button type="submit" value="Submit"
                                                class="btn btn-warning"
                                                title="Click para realizar la Modificación...">
                                                Modificar
                                        </button>
                                    </form>
                                </td>

```

```
99
100                                 <td width ="75px">
101
102                                     <a href="#"
103                                         class="btn btn-danger btn-delete"
104                                         title="Click para eliminar este registro...">
105                                         Eliminar
106                                     </a>
107                                 </td>
108                             </tr>
109                         {% endfor %}
110
111                     </tbody>
112                 </table>
113             </div>
114             <br/>
115             <div class="card-body text-center">
116                          
117                     <a href="/reporte01/r01_pdf"  class="btn btn-outline-primary"
118                             title="Listado de todos los cargos">
119                         Print
120                     </a>
121                          
122                     <a href="/login/admin"  class="btn btn-outline-primary"
123                             title="Salir al Menú principal del Administrador">
124                         Salir
125                     </a>
126             </div>
127             <br/>
128         </div>
129         <!--Fin del Bloque Centro / Cuerpo Container Principal -->
130         <BR/>
131
132         <!--Inicio del Bloque Pie de página  -->
133         {% include 'm01_sitio/zpie.html' %}
134         <!--Fin del Bloque Pie de Página -->
135         <BR/>
136     </body>
137     <!-- Fin del Body -->
138
139 </html>
```

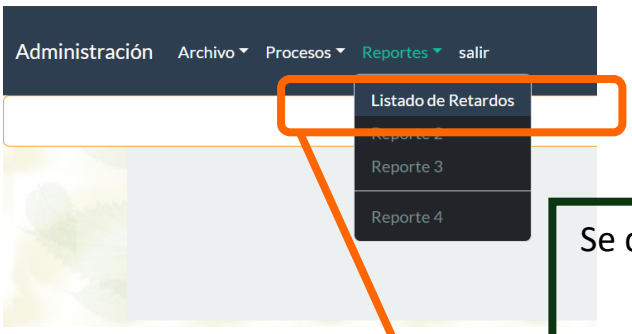
## c.- Programa --→ ymenu_n2.html

```
<!-- Programa: ymenu_n2.html -->
<!-- Autor: Sergio Capacho / Jueves 14/Mayo/2026 09:35 Pm -->
<!-- Objetivo: Menú Nivel 2 del Reporte #1 Para generar un listado general de los retardos -->
<div class="container text-center">
    <div class="card border-warning mb-0" >
        <h4 class="card-title mt-3">Módulo Reportes</h4>
        <div class="card-body">
                 
            <a href="/reporte01/r01_pdf"  class="btn btn-outline-primary"
                    title="Listado de todos los cargos">
                Print
            </a>
                 
            <a href="/login/admin"  class="btn btn-outline-primary"
                    title="Salir al Menú principal del Administrador">
                Salir
            </a>
        </div>
    </div>

</div>
<!-- Fin le menú nievel #2 . -->
```

# 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_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. D
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
```