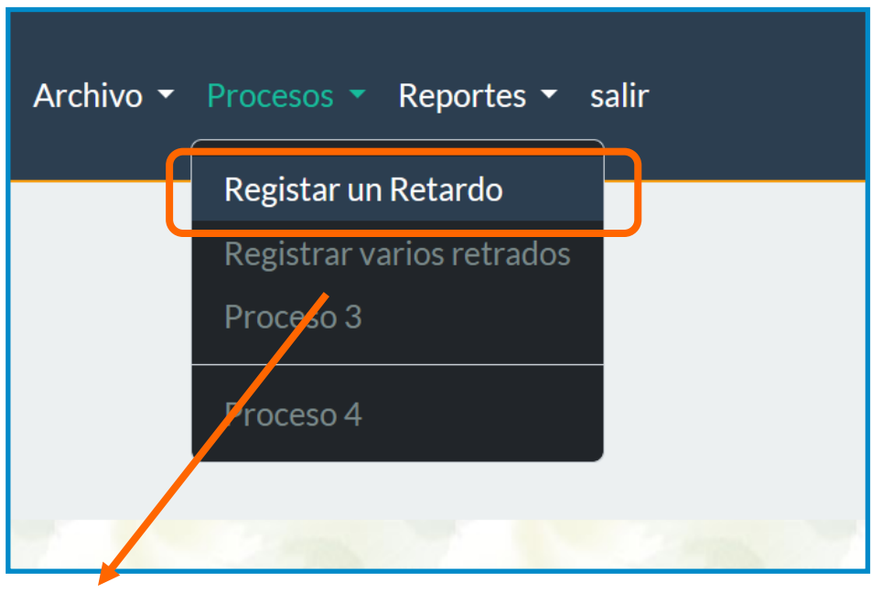
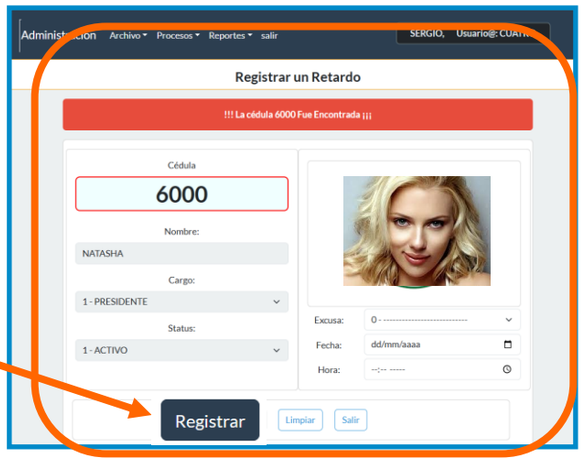
Registrar un retardo (Módulo 21)







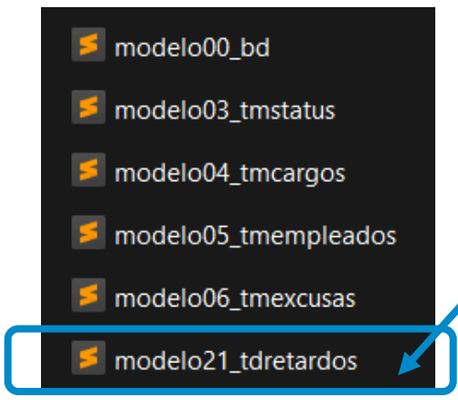


Pasos para construir el módulo 21

(Registrar un retardo)

Paso 1.- Construir los modelos

modelos



Se necesitan las siguientes funciones:
Modelo00_bd --→ cadenaConexion
modelo03_tmstatus --→ get_all_status
modelo04_tmcargos --→ get_all_vtmcargos
modelo05_tmempleados -→ get_empleado_by_cedemple
modelo06_tmexcusas --→ get_all_tmexcusas
modelo21_tdretardos --→ add_retardo

Funciones que se usan para las operaciones (CRUD) de cada modelo (Tabla) :
C (Agregar);
R (Leer/Consultar);
U (Actualizar);
D (borrar).

1.1.- Modelo00_bd -→ cadenaConexion()

```
#Programa: modelo00_DB.py
#Objetivo:  Declarar  y  Construir la cadena de conexión con la BD
#Autor: Sergio Capacho / Sábado 24/Oct/2026 11:00 Pm

#1.- IMPORTAR  LIBRERIAS
from   psycopg   import   connect

#2.- CREAR LA CADENA DE CONEXIÓN
Def   cadenaConexion():
        conexion = connect( host = "localhost",
                            port = 5432,
                            dbname = "bdretardados",
                            user = "postgres",
                            password = "1234567"    )
        return conexion
```

1.2.- Modelo03_tmstatus -→ get_all_status

```
#Programa: modelo03_tmstatus.py
#Objetivo: Manejar las funciones para las operaciones con la TMSTATUS
#Autor: Sergio Capacho / Sábado 24/Oct/2026  -  11:00 Pm

#1.- IMPORTAR   LIBRERIAS
from   psycopg   import    connect
#.....................................................

#2.- IMPORTAR EL MODELO DE LA BD CON SU LA CONEXIÓN
from   modelos.modelo00_bd   import   cadenaConexion
#.....................................................

#3.- OPERACIONES SQL DE LA TMSTATUS
#.....................................................

#3.1.- Obtener todos los registros de la TMSTATUS
def   get_all_status():
        #PASO 1 ---> Establecer la conexión con la base de datos
        conexion   =   cadenaConexion()
        cursor = conexion.cursor()

        #PASO 2 ---> Construir la consulta SQL
        sql = "SELECT * FROM  tmstatus ORDER BY pkcods::BIGINT;"
        try:
                #PASO 3 ---> Ejecutar la consulta SQL
                cursor.execute(sql,  )  #Ejecute
                conexion.commit()      #Confirma

                #PASO 4 ---> Recuperar resultados
                xresultados = cursor.fetchall()   #fetchone() ---> Un solo  /  #fetchall() ---> Varios
                #print("Datos recuperados de la TMStatus ---> " + str(xresultado))
                xfilas = cursor.rowcount          #cursor.rowcount ---> Total de filas afectadas
        except Exception as e:
                print( f"Este fue el Error: {e}")
                #PASO 4 ---> Recuperar resultados
                xresultados = 0
                xfilas = 0

        print("Total de Filas afectadas ---->" + str(xfilas))

        #PASO 5 ---> Cerrar la conexión con la base de datos
        cursor.close()
        conexion.close()

        #PASO 6 ---> Retornar resultados
        return xresultados
```

1.3.- modelo04_tmcargos --→ get_all_vtmcargos()

```
#Programa: modelo04_tmcargos.py
#Objetivo: Manejar las funciones para las operaciones con la TMCARGOS
#Autor: Sergio Capacho / Domingo 25/Oct/2026  -  11:00 Pm

#1.- IMPORTAR  LIBRERIAS
from   psycopg   import    connect

#2.- IMPORTAR EL MODELO Y LA CONEXIÓN CON LA BD
from   modelos.modelo00_bd   import   cadenaConexion

#3.- OPERACIONES SQL PARA LA TMCARGOS
#3.1.- Obtener todos los registros de la Vista TMCargos
def   get_all_vtmcargos():
        #PASO 1 ---> Establecer la conexión con la BD
        conexion = cadenaConexion()
        cursor = conexion.cursor()

        #PASO 2 ---> Construir la consulta SQL
        sql ="select * from vtmcargos order by pkcodcar::BIGINT;"

        try:
                #PASO 3 ---> Ejecutar la consulta SQL
                cursor.execute(sql,  )     #Ejecuta
                conexion.commit()         #Confirma

                #PASO 4 ---> Recuperar los resultados
                xresultados = cursor.fetchall()    #fetchone() ---> Un / fetchall() ---> Varios
                xfilas = cursor.rowcount          # ---> Total de filas afectadas en la operación
        except Exception as e:
                print( f"Este fue el Error: {e}")
                #PASO 4 ---> Recuperar los resultados
                xresultados = 0
                xfilas = 0

        #print("Total de Filas afectadas ----------> " + str(xfilas))
        #print("Datos recuperados de la TMCargos ---> " + str(xresultados))

        #PASO 5 ---> Cerrar la conexión con la base de datos
        cursor.close()
        conexion.close()

        #PASO 6 ---> Retornar resultados
        return xresultados
```

1.4.- modelo05_tmempleados -→ get_empleado_by_cedemple

```
#Programa: modelo05_tmempleados.py
#Objetivo: Funciones para manejar las operaciones CRUD en la TMEMPLEADOS
#Autor: Sergio Capacho / Fecha: Miércoles 28/Oct/2026 - 11:00 Pm

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


#2.- IMPORTAR  MODELOS
#2.1.- Importación de la conexión con la BD
from  modelos.modelo00_bd  import  cadenaConexion


#3.- OPERACIONES SQL CON TMEMPLEADOS
#3.1.- Obtener todos los registros de la TMEMPLEADOS

#3.2.- Agregar un Empleado Nuevo a la TMEMPLEADOS

#3.3.- Buscar (Obtener) los datos del empleado por medio del número de cédula
def  get_empleado_by_cedemple(xcedemple):
        #print("Número de cédula en el Modelo ---> " + str(xcedemple))
        #PASO 1 ---> Establecer la conexión con la BD
        conexion = cadenaConexion()
        cursor = conexion.cursor()

        #PASO 2 ---> Se construye la consulta SQL
        sql = "SELECT * FROM tmempleados  WHERE pkcedemple = %s; "
        datos = (xcedemple, )

        try:
                #PASO 3 ---> Ejecutar la consulta
                cursor.execute(sql, datos)  #Ejecutar
                conexion.commit()         #Confirmar

                #PASO 4 ---> Recuperar los datos
                xresultados = cursor.fetchone()   #fetchone() ---> Uno   /   fetchall() ---> Varios
                xfilas = cursor.rowcount         #Total de filas afectadas
        except Exception as e:
                printf(f"Este fue el Error: {e}")
                #PASO 4 ---> Recuperar los datos
                xresultados = 0
                xfilas = 0
```

```
192
193         #print("Total de Filas afectadas ----------------> " + str(xfilas))
194         #print("Datos recuperados de la TMCargos ---> " + str(xresultados))
195
196         #PASO 5 ---> Cerrar la conexión con la BD
197         cursor.close()
198         conexion.close()
199
200         #PASO 5 ---> Retornar resultado
201         return xresultados
202
203 #3.4.- Modificar (Set ---> Fijar) los datos deL Empleado
204
205 #3.5.- Eliminar (Set ---> Fijar) lógicamente un Empleado
206 #.....
207 #.....................................................
208
209
210
211
212
213
```

214 1.5.- modelo06_tmexcusas--→ 215 get_all_tmexcusas

```
216
217 #Programa: modelo06_tmexcusas.py
218 #Objetivo: Funciones para manejar las operaciones CRUD en la TMEXCUSAS
219 #Autor: Sergio Capacho / Fecha: Miércoles 28/Oct/2026 - 11:00 Pm
220
221 #1.- IMPORTAR  LIBRERIAS
222 from flask  import  Blueprint,  render_template,  flash redirect
223 from  flask  import  url_for,  request,  Flask,  session
224 from  psycopg  import  connect
225
226 #2.- IMPORTAR  MODELOS
227 #2.1.- Importación de la conexión con la BD
228 from  modelos.modelo00_bd  import  cadenaConexion
229
230 #3.- OPERACIONES SQL CON TMEXCUSAS
231 #3.1.- Obtener todos los registros de la TMEXCUSAS
232 Def  get_all_tmexcusas():
233         #PASO 1 ---> Establecer la conexión con la BD
234         conexion = cadenaConexion()
235         cursor = conexion.cursor()
236
237         #PASO 2 ---> Construir la consulta SQL
238         sql ="SELECT * FROM tmexcusas WHERE fkcods = 1;"
239
```

```
        try:
            #PASO 3 ---> Ejecutar la consulta SQL
            cursor.execute(sql,  )  #Ejecutar
            conexion.commit()      #Confirmar

            #PASO 4 ---> Recuperar resultados
            xresultados = cursor.fetchall()  #fetchone() ---> Un   /   fetchall() ---> Varios
            xfilas = cursor.rowcount        #---> Total de filas afectadas
        except Exception as e:
            printf(f"Este fue el Error: {e}")
            #PASO 4 ---> Recuperar los datos
            xresultados = 0
            xfilas = 0

        #print("Total de Filas afectadas ----------> " + str(xfilas))
        #print("Datos recuperados de la TMCargos ---> " + str(xresultados))

        #PASO 5 ---> Cerrar la conexión con la BD
        cursor.close()
        conexion.close()

        #PASO 6 ---> Retornar resultado
        return xresultados

#3.2.- Agregar una nueva excusa a la TMEXCUSAS

#3.3.- Buscar (Get ----> Obtener) los datos de la Excusa por medio del código

#3.4.- Modificar (Set ---> Fijar) los datos de la Excusa

#3.5.- Eliminar (Set ----> Fijar) lógicamente una Excusa
```

1.6.- modelo21_tdretardos --→ add_retardo

```
#Programa: modelo21_tdretardos.py
#Objetivo: Funciones para trabajar con la TDRETARDOS
#Autor: Sergio Capacho / Fecha: Miércoles 28/Oct/2026 - 11:00 Pm

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

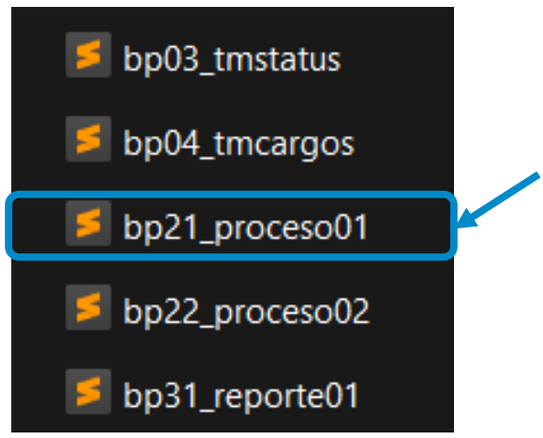
#2.- IMPORTAR  MODELOS
#2.1.- Importación de la conexión con la BD
from  modelos.modelo00_bd  import  cadenaConexion
```

```
290
291 #3.- OPERACIONES SQL CON TDRETARDOS
292
293 #3.1.- Registrar un retardo nuevo
294 def  add_retardo(xfkcedemple, xfkcodexcu, xfecha, xhora):
295         #PASO 1 ---> Establecer la conexión con la BD
296         conexion = cadenaConexion()
297         cursor = conexion.cursor()
298
299         #PASO 2 ---> Construir la consulta SQL
300         sql ="INSERT  INTO tdretardos  (fkcedemple,   fkcodexcu,  fecha,  hora)
301                                         VALUES (%s, %s, %s, %s);"
302         datos = (xfkcedemple, xfkcodexcu, xfecha, xhora)
303
304         try:
305                 #PASO 3 ---> Ejecutar la consulta SQL
306                 cursor.execute(sql, datos)  #Ejecutar
307                 conexion.commit()         #Confirmar
308
309                 #PASO 4 ---> Recuperar los resultados
310                 xfilas = cursor.rowcount # ---> Total de filas afectadas en la operación
311         except Exception as e:
312                 print( f"Este fue el Error: {e}")
313                 #PASO 4 ---> Recuperar los resultados
314                 xfilas = 0
315
316         #print("Total de Filas afectadas ----------> " + str(xfilas))
317
318         #PASO 5 ---> Cerrar la conexión con la base de datos
319         cursor.close()
320         conexion.close()
321
322         #PASO 6 ---> Retornar resultado
323         return xfilas
324
325
326 #3.2.- Obtener un listado de todos los retardos (Vista retardados00)
327
328
329 #3.3.- Obtener (Get)  a cantidad total de retardos
330
331
332 #3.4.- Obtener ( Get ) la fecha del servidor en formato
333 #       Lunes...(Dia_semana), dd/Mes_texto/Año
334
335
336 #3.5.- Obtener (Get) la hora del servidor en formato HH:MM AmPm
337 #.....
338 #.....
339
340
341
```

Paso 2.- Construir el controlador

(bp21_proceso01.py)

blueprints



El programa llamado "**bp21_proceso01.py**" es quien controla todas las operaciones para el proceso "**Registrar Un Retardo**". Además, maneja las siguientes rutas:

- /proceso01/paso01
  - **Cargar paso01.html** (Formulario para leer el número de cédula del empleado)
- /proceso01/paso02 - POST
  - Buscar datos (empleado, cargo, status y excusas)
  - **Cargar paso03.html** (Formulario para capturar el retardo del empleado – Código del retardo, fecha y hora)
- /proceso01/paso04 - POST
  - **Registrar los datos del** retardo del empleado
- /proceso01/limpiar
  - **Cargar paso01.html** (Formulario para leer el número de cédula del empleado)

2.1.- bp21_proceso01.py

```
#Programa: bp21_proceso01.py
#Objetivo: Controlador para el proceso #1 ----> Registrar un Retardo
#Autor: Sergio Capacho / Fecha: Miércoles 28/Oct/2026 - 11:00 Pm

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

```
#2.- IMPORTAR  MODELOS
#2.1.- Importar la BD y su conexión
From   modelos.modelo00_bd   import   cadenaConexion

#2.2.- Importar el modelo modelo03_tmstatus para Traer todos los Status
from   modelos.modelo03_tmstatus   import   get_all_status


#2.3.- Importar el modelo modelo04_tmcargos para Traer todos los Cargos
from   modelos.modelo04_tmcargos   import   get_all_vtmcargos


#2.4.- Importar el modelo modelo08_tmempleados para Traer  los datos de un empleado X
from   modelos.modelo05_tmempleados
       import   get_empleado_by_cedemple

#2.5.- Importar el modelo modelo11_tmexcusas para Traer todas las Excusas
from   modelos.modelo06_tmexcusas   import   get_all_tmexcusas


#2.6.- Importar el modelo modelo21_tdretardos para Registrar un retardo nuevo
from   modelos.modelo21_tdretardos   import   add_retardo


#3.- CREAR DEL OBJETO PORCESO01
obj_proceso01   =   Blueprint('proceso01', __name__)


#4.- REGISTRAR LAS RUTAS DEL OBJETO PROCESO01
#4.1.- RUTA Menú ---> /proceso01/paso01
@obj_proceso01.route('/paso01')
Def  paso01():
    if  (session["xlogin"]    and    session["xnivel"] in (2, 4) ):
        xmen = " "
        xempleado = None
        return render_template('m21_proceso01/paso01.html',
                               yempleado = xempleado ,
                               ymensaje = xmen)
    else:
        return redirect('/login/entrar')
```

#4.2.- RUTA Buscar ---> /proceso01/paso02

```
@obj_proceso01.route('/paso02', methods=['GET', 'POST'])
def   paso02():
    if (session["xlogin"]    and    session["xnivel"] in (2, 4) ):
        if request.method == 'POST':
            xmen = " "
            xcedemple = request.form['txt_cedemple']

            xtmempleado  =  get_empleado_by_cedemple(xcedemple)
            xtmcargos  =  get_all_vtmcargos()
            xtmstatus  =  get_all_status()
            xtmexcusas  =  get_all_tmexcusas()

            #print("Datos desde el controlador.....")
            #print("\nCédula en el controlador  ---> " + str(xcedemple))
            #print("\nXEmpleado ----------> " + str(xtmempleado))
            #print("\nXCargos ------------> " + str(xtmcargos))
            #print("\nXStatus ------------> " + str(xtmstatus))

            if xtmempleado :
                xmen = "!!! La cédula  " + str(xcedemple) + " Fue un Éxito ¡¡¡"
            else:
                xmen = "!!! La cédula  " + str(xcedemple) + "  No Existe ¡¡¡"

            return render_template('m21_proceso01/paso03.html',
                                   yempleado = xtmempleado,
                                   ymensaje = xmen,
                                   ytmstatus = xtmstatus,
                                   ytmcargos = xtmcargos,
                                   ytmexcusas = xtmexcusas   )
        else:
            return redirect('/login/entrar')
```

```
#4.3.- RUTA Registrar ---> /proceso01/paso04
@obj_proceso01.route('/paso04', methods=['GET', 'POST'])
def   paso04():
    if (session["xlogin"]    and     session["xnivel"] in (2, 4) ):
        xmen = " "
        if request.method == 'POST':
            xfkcedemple   =  request.form['txt_cedemple']
            xfkcodexcu   =  request.form['txt_fkcodexcu']
            xfecha   =   request.form['txt_fecha']
            xhora   =   request.form['txt_hora']

            #print("Datos del retardos a registrar...")
            #print(str(xfkcedemple))
            #print(str(xfkcodexcu))
            #print(str(xfecha))
            #print(str(xhora))

            xresultados = add_retardo(xfkcedemple, xfkcodexcu,
                                      xfecha, xhora)


            if   xresultados == 1:
                xmen = "¡¡¡ El retardo de  " + xfkcedemple + " fue Registrado !!!"
            else:
                xmen = "¡¡¡ El retardo de  " + xfkcedemple + " No fue Registrado !!!"

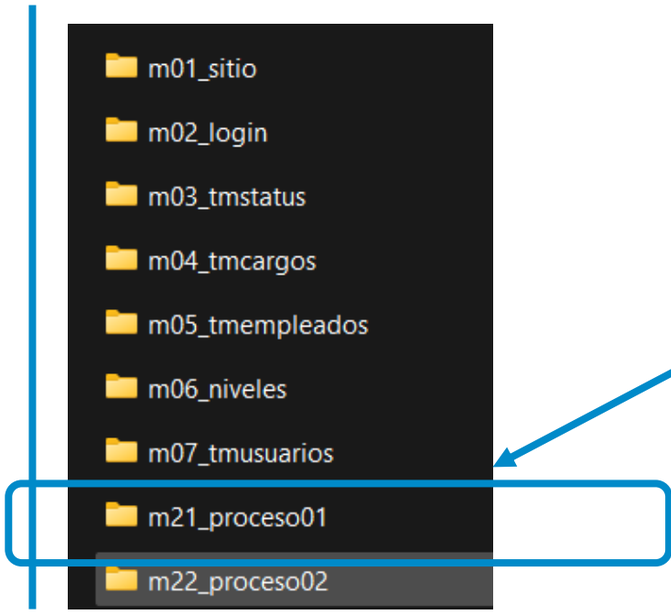

            return render_template('m21_proceso01/paso01.html',
                                   yempleado = " ",
                                   ymensaje = xmen)
        else:
            return redirect('/login/entrar')


#4.4.- RUTA ---> /proceso01/limpiar
@obj_proceso01.route('/limpiar')
Def   limpiar():
    if (session["xlogin"]   and    session["xnivel"] in (2, 4)   ):
        return render_template('m21_proceso01/paso01.html',
                               xempleado = " ", ymensaje = " ")
    else:
        return redirect('/login/entrar')

#.....................................................................
```

Paso 3.- Construir las vistas

templates



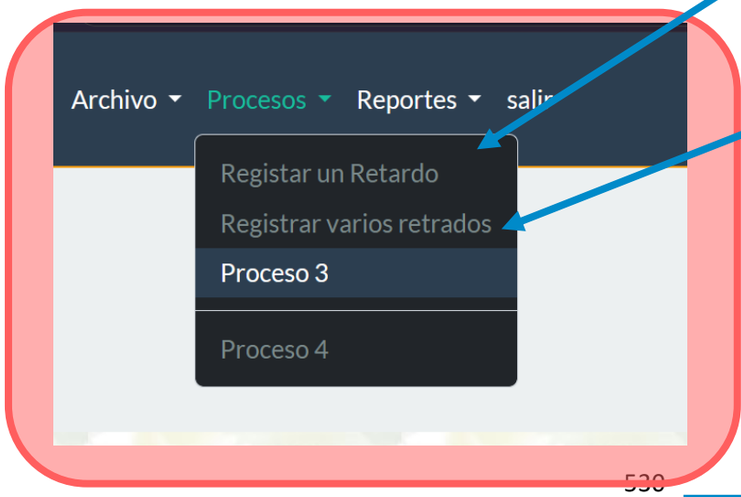
Las vistas que corresponden al proceso "**Registrar un Retardo**" son:

- paso01
- paso03
- proceso01_zmenu_n2

Las cuales deben quedar dentro de la carpeta **m21_proceso01**

Ruta /proceso01/paso01

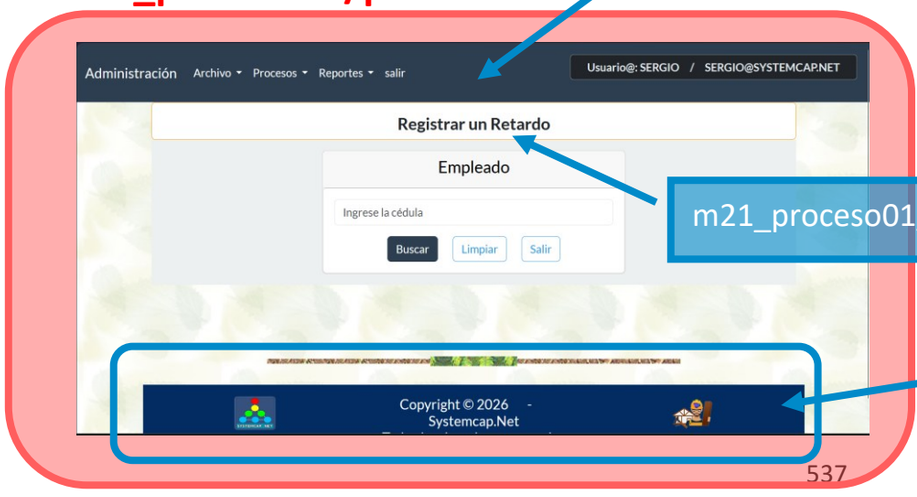
Menú principal del sistema



Ruta /proceso02/paso01

m02_login/paso03_admin_zmenu_n1.htm

M21_proceso01/paso01.html



m21_proceso01/**proceso01_zmenu_n2.html**

m01_sitio/zpie.html

3.1.- proceso01_zmenu_n2.html

```
<!-- Programa: proceso01_zmenu_n2.html -->
<!-- Autor: Sergio Capacho / Jueves 29/Oct/2026 11:00 Pm -->
<!-- Objetivo: Menú del Nivel 2 para el proceso 01 registrar un retardos -->
<div class="container text-center">
    <div class="card border-warning mb-0" >
        <h4 class="card-title mt-3">
            <b>Registrar un Retardo</b>
        </h4>

        </div>
    </div>

</div>
<!-- Fin  Bloque Sub Menú Proceso01. -->
```

3.2.- paso01.html

```
<!-- Programa: Proceso 01 - Paso01.html -->
<!-- Autor: Sergio Capacho / Jueves 29/Oct/2026 11:00 Pm -->
<!-- Objetivo: Paso #1 para registrar el retardo de un empleado -->
<!DOCTYPE html>
<html lang="es">
    <!-- Inicio del Bloque Head -->
    {% include 'm01_sitio/zhead.html' %}
    <!-- Inicio del Bloque Head -->

    <!-- Inicio del Body -->
    <body    onload="startTime()">
        <!--- Inicio del Bloque Menú Nivel #1 (Módulo Administración) -->
        {% include 'm02_login/paso03_admin_zmenu_n1.html' %}
        <!--- Fin del Bloque Menú Nivel #1 (Módulo Administración) -->

        <!--- Inicio del Bloque Menú Nivel #2 (Proceso01) -->
        {% include 'm21_proceso01/proceso01_zmenu_n2.html' %}
        <!--- Inicio del Bloque Menu Nivel #2 (Proceso01-->


        <!--Inicio del Bloque Centro /  Container Principal  -->
        <div class="container text-center">
            <div class="mb-0 p-3 bg-light">
                div class="row">  <!— Inicio Fila --->
                    <div class="col-md-3">   <!— Inicio de tres columnas  -->
                    </div>  <!---  Fin de tres columnas  --->
```

```
584                     <div class="col-md-6"> <!—Inicio de 6 columnas -->
585                         <div class="card"> <!--  Inicio de la tarjeta -->
586                             <div class="card-header text-center">
587                                 <h4 style="color:black">Empleado</h4>
588                             </div>
589
590                             {% if   (ymensaje != " ")   %}
591                                 <div class="card-footer text-center">
592                                     <div class="alert alert-danger"
593                                                 role="alert">
594                                         <strong>
595                                             {{ ymensaje }}
596                                         </strong>
597                                     </div>
598                                 </div>
599                             {% endif %}
600
601                             <div  class="card card-body">
602 <!-- Inicio del Formulario para la  lectura de cédula del empleado -->
603 <form  action="/proceso01/paso02" method="POST">
604         <!-- Inicio de la lectura del número de cédula -->
605         <div class="mb-3">
606             <input   type="text"    id="txt_cedemple"   name="txt_cedemple"
607                         placeholder="Ingrese la cédula"   class="form-control"/>
608         </div>
609         <!-- Fin de la lectura del número de cédula -->
610
611         <div class="text-center">
612             <button   type="submit"    value="Submit"   class="btn btn-primary"
613                                             title="Click para realizar la busqueda...">
614                         Buscar
615             </button>
616                              
617             <a href="/proceso01/limpiar"    class="btn btn-outline-info"
618                                             title="Click para limpiar el formulario...">
619                         Limpiar
620             </a>
621                              
622             <a href="/login/admin"    class="btn btn-outline-info"
623                                             title="Click para Salir...">
624                         Salir
625             </a>
626         </div>
627 </form>
628 <!-- Fin del Formulario para la lectura de cédula del empleado -->
629                             </div>
630                         </div> <!--  Fin de la tarjeta -->
631                     </div> <!—Fin de 6 columnas -->
632                     <div class="col-md-3"> <!—Inicio de 3 columnas --->
```

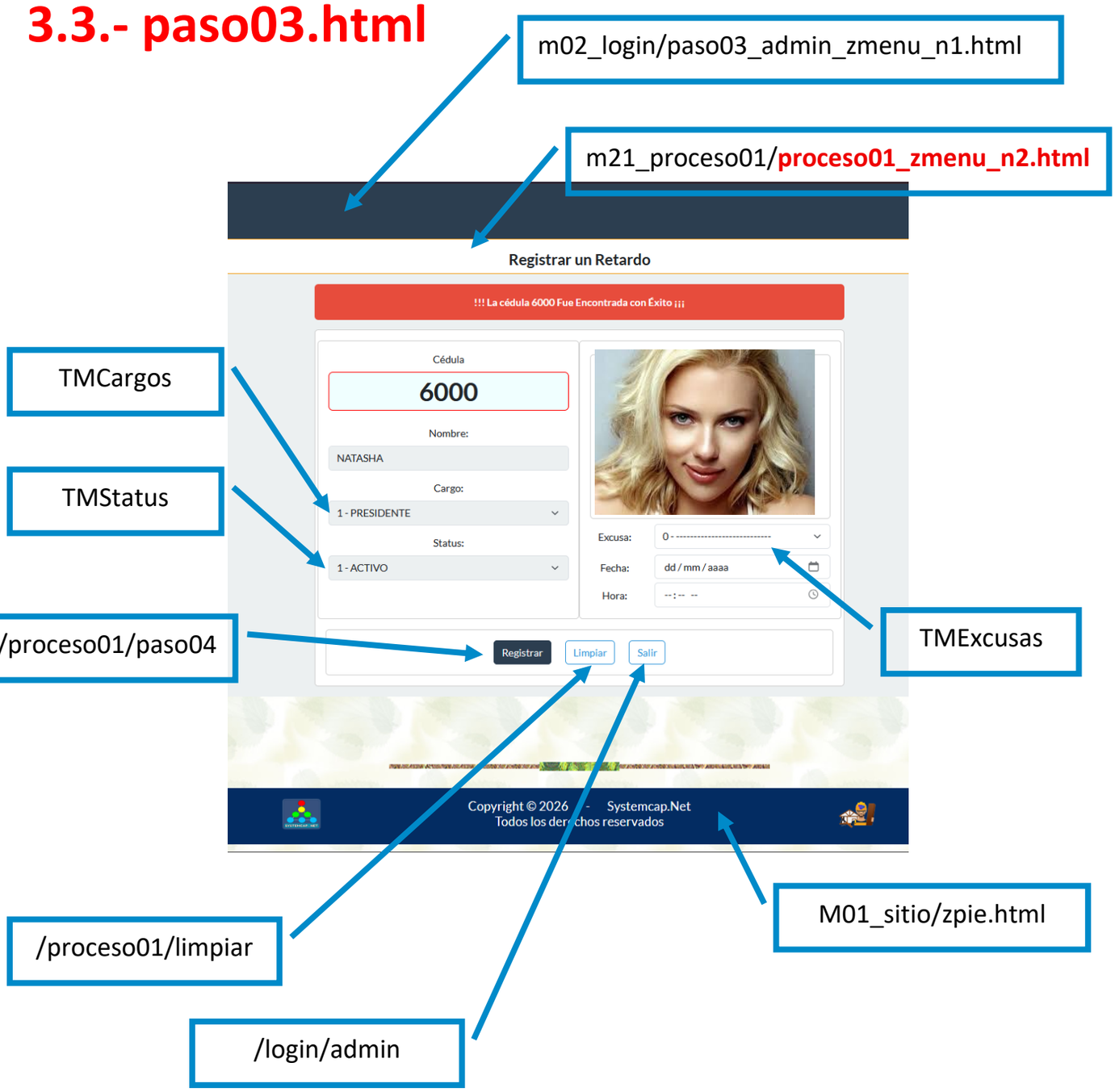
El formulario donde se lee la cédula del empleado

```
                    </div>  <!--  Fin de 3 columnas  -->
                </div> <!—Fin  de la  fila --->
            </div>
            <br/>
            <br/>
        </div>
        <!--Fin del Bloque Centro / Cuerpo Container Principal -->
        <BR/>

        <!--Inicio del Bloque Pie de página  -->
        {% include 'm01_sitio/zpie.html' %}
        <!--Fin del Bloque Pie de Página -->

    </body>
    <!-- Fin del Body -->
</html>
```

3.3.- paso03.html



```
683
684 <!-- Programa: paso03.html -->
685 <!-- Autor: Sergio Capacho / Domingo 10/Mayo/2026 04:29 Pm -->
686 <!-- Objetivo: Pase #3 para registrar el retardo de un empleado -->
687 <!DOCTYPE html>
688 <html lang="es">
689         <!-- Inicio del Bloque Head -->
690         {% include 'm01_sitio/zhead.html' %}
691         <!-- Inicio del Bloque Head -->
692         <!-- Inicio del Body -->
693         <body    onload="startTime()">
694                 <!--- Inicio del Bloque Menú Nivel #1 (Módulo Administración) -->
695                 {% include 'm02_login/paso03_admin_zmenu_n1.html' %}
696                 <!--- Fin del Bloque Menú Nivel #1 (Módulo Administración) -->
697                 <!--- Inicio del Bloque Menú Nivel #2 (Proceso01) -->
698                 {% include 'm21_proceso01/proceso01_zmenu_n2.html' %}
699                 <!--- Inicio del Bloque Menu Nivel #2 (Proceso01-->
700                 <!--Inicio del Bloque Centro /  Container Principal  -->
701                 <div class="container text-center">
702                         <div class="mb-0 p-3 bg-light">
703                                 <div class="row">   <!--  Inicio Fila  1 -->
704                                         <div class="col-md-2">  </div>
705
706                                         <div class="col-md-8">
707                                                 {%   if   (ymensaje != " ")  %}
708                                                         <div class="card-footer text-center">
709                                                                 <div class="alert alert-danger"
710                                                                                                 role="alert">
711                                                                         <strong>
712                                                                                 {{ ymensaje }}
713                                                                         </strong>
714                                                                 </div>
715                                                         </div>
716                                                 {%   endif   %}
717
718 <!-- Inicio del Formulario para registrar un retardo --->
719 <form action="/proceso01/paso04" method="POST" class="card card-body">
720         <div class="row">  <!--  Inicio Sub -Fila  1.1 -->
721                 <div class="col-md-6 card card-body">  <!--  Inicio Fila  1.1 (Columna A) -->
722                         <div class="mb-3">   <!--  Bloque Cédula del empleado -->
723                                 <label for="txt_cedemple"  class="form-label">Cédula</label>
724                                 <input   type="text"  id="txt_cedemple"
725                                                                 name="txt_cedemple"
726                                                                 class="form-control"
727                                                         value = "  {{ yempleado[0]}}  "  hidden  />
728                                 <h2  id="h2_estilo_codigo"><b>  {{yempleado[0] }} </b></h2>
729                         </div>
730
731
```

El formulario para leer los datos del retardo

Muestra la Cédula del empleado

```
            <div class="mb-3"> <!--   Bloque Nombre del empleado  -->
                <label for="txt_nomemple" class="form-label">Nombre: </label>
                <input   type="text"
                         id="txt_nomemple"   name="txt_nomemple"
                         class="form-control"
                         value ="  {{ yempleado[1] }}  " disabled />
            </div>

            <div class="mb-3"> <!--- Bloque  cargo del empleado -->
                <label for="txt_fkcodcar" class="form-label">Cargo:</label>

                <select id="txt_fkcodcar" name="txt_fkcodcar"
                                    class="form-select" disabled >
                    {%   for   xcar   in  ytmcargos   %}
                    {%  if  yempleado[3] ==  xcar[0]  %}
                        <option value="  {{ xcar[0] }}  " selected>
                            {{ xcar[0] }}   -   {{ xcar[1] }}
                        </option>
                    {% else %}
                        <option value="  {{ xcar[0] }}  ">
                            {{ xcar[0] }}   -   {{ xcar[1] }}
                        </option>
                    {% endif %}
                {%  endfor  %}
                </select>
            </div>

            <div class="mb-3"> <!-- Bloque status del empleado -->
                <label for="txt_fkcods" class="form-label">Status:</label>
                <select id="txt_fkcods" name="txt_fkcods"
                        class="form-select" disabled  >
                    {% for   xs   in   ytmstatus %}
                        {%  if  yempleado[4]   ==   xs[0]  %}
                            <option value="  {{ xs[0] }}  " selected>
                                {{ xs[0] }}   -   {{ xs[1] }}
                            </option>
                        {% else %}
                            <option value="  {{ xs[0] }}  ">
                                {{ xs[0] }}   -   {{ xs[1] }}
                            </option>
                        {% endif %}
                    {% endfor %}
                </select>
            </div>
        </div>   <!--  Fin Fila  1.1 (Columna A) -->
```

Muestra el nombre del empleado

Muestra el nombre del cargo asociado a la fkcodcar del empleado en una lista desplegable

Muestra la descripción del status asociado a la fkcods del empleado en una lista desplegable

```
<!--  Inicio de la Tarjeta de la Excusa -->
<div class="col-md-6 card card-body">   <!--  Inicio Fila  1.1 (Columna B) -->
    <div class="card card-body mt-1"> <!-- Foto del empleado -->
        <img src="../static/img/fotos/foto_perfil.png"
             class="rounded mx-auto d-block"
             alt="Foto perfil"/>
    </div>
    <div class="mt-2">  <!—Bloque Combo List de las excusas -->
        <div class="row">
            <div class="col-md-3">
                <label for="txt_fkcodexcu" class="form-label mt-2">
                    Excusa:
                </label>
            </div>
            <div class="col-md-9">
                <select   id="txt_fkcodexcu"
                          name="txt_fkcodexcu"
                          class="form-select">
                    {%  for   excu   in   ytmexcusas   %}
                        {%  if  excu[0]   ==   0  %}
                            <option value="{{ excu[0] }}"
                                    selected>
                                {{ excu[0] }}  -  {{ excu[1] }}
                            </option>
                        {% else %}
                            <option value="{{ excu[0] }}">
                                {{ excu[0] }} - {{ excu[1] }}
                            </option>
                        {% endif %}
                    {% endfor %}
                </select>
            </div>
        </div>
    </div>
    <div class="mt-2">  <!--  Bloque fecha de la excusa -->
        <div class="row">
            <div class="col-md-3">
                <label for="txt_fecha" class="form-label mt-2">
                    Fecha:
                </label>
            </div>
            <div class="col-md-9">
                <input type="date"
                       id="txt_fecha"   name="txt_fecha"
                       class="form-control" />
            </div>
        </div>
    </div>
```

Foto del empleado

Lista desplegable de las excusas

Fecha de la excusa

```
                <div class="mt-1">   <!--  Bloque hora de la excusa -->
                    <div class="row">
                        <div class="col-md-3">
                            <label for="txt_hora" class="form-label mt-2">
                                Hora:
                            </label>
                        </div>
                        <div class="col-md-9">
                            <input type="time"
                                   id="txt_hora"   name="txt_hora"
                                   class="form-control" />
                        </div>
                    </div>
                </div>
            </div> <!--  Fin Fila  1.1 (Columna B) -->
            <!--  Fin de la Tarjeta de la Excusa -->
        </div>  <!--  Fin Sub -Fila  1.1 -->

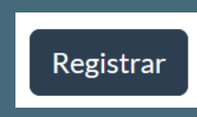
        <div class="row">  <!--  Inicio Sub -Fila  1.2 -->
            <div class="col-md-12">      <!--  Bloque de los  botones -->
                <div class="card card-body mt-3">
                    <div class="text-center">
                        <button   type="submit"   value="Submit"
                                  class="btn btn-primary"
                                  title="Click para registrar el retardo...">
                            Registrar
                        </button>
                                         
                        <a href="/proceso01/limpiar"
                                  class="btn btn-outline-info"
                                  title="Click para limpiar el formulario...">
                            Limpiar
                        </a>
                                         
                        <a href="/login/admin"
                                  class="btn btn-outline-info"
                                  title="Click para Salir...">
                            Salir
                        </a>
                    </div>
                </div>
            </div>
        </div>  <!--  Fin Sub -Fila  1.2 -->
</form>
<!-- Fin del Formulario para registrar un retardo --->
                    </div>
```

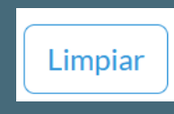
Hora de la excusa

Botón Registrar. Asociado a la ruta del formulario **/proceso01/paso04**

Botón Limpiar.

Botón Salir







```
                        <div class="col-md-2"></div>
                    </div>    <!--  Fin Fila  1 -->
                </div>
                <br/>
                <br/>
            </div>
            <!--Fin del Bloque Centro / Cuerpo Container Principal -->
            <BR/>

            <!--Inicio del Bloque Pie de página  -->
            {% include 'm01_sitio/zpie.html' %}
            <!--Fin del Bloque Pie de Página -->

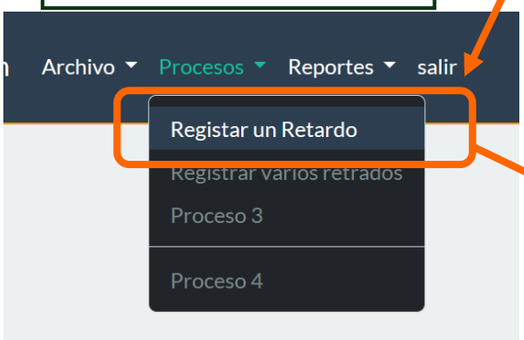
        </body>
        <!-- Fin del Body -->
</html>
```

Paso 4.- Agregar la ruta “/proceso01/paso01” en el programa paso03_admin_zmenu_n1.html

m02_login/paso03_admin_zmenu_n1.html

Agregar la ruta **“/proceso01/paso01”** dentro del código del programa **paso03_admin_zmenu_n1.html**

Buscar el programa paso03_admin_zmenu_n1.html en la carpeta m02_login




```
<li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" data-bs-toggle="dropdown" href="#"
            role="button" aria-haspopup="true" aria-expanded="false">
                Procesos
    </a>
    <div class="dropdown-menu">
            <a class="dropdown-item" href="/proceso01/paso01">
                Registar un Retardo
            </a>
            <a class="dropdown-item" href="/proceso02/paso01">
                Registrar varios retrados
            </a>
            <a class="dropdown-item" href="#">Proceso 3</a>
            <div class="dropdown-divider"></div>
            <a class="dropdown-item" href="#">Proceso 4</a>
    </div>
</li>
```

```
<a class="dropdown-item" href="/proceso01/paso01">
    Registar un Retardo
</a>
```

Paso 5: En el programa app.py

5.1.- Importar el controlador para registrar un retardo; en la práctica es importa la variable objeto llamada: obj_proceso01, la cual fue creada en el programa bp21_proceso01 (ver página 9 de esta guía).

```
#02.02 ---> Importación del objeto controlador de la TMCARGOS (bp04_tm
#           que está en la carpeta blueprints
from blueprints.bp04_tmcargos import obj_tmcargos


#02.03 ---> Importación el objeto TMEMPLEADOS (pb05_tmempleados / obj_tmempleados)

#02.04 ---> Importación el objeto TMEXCUSAS (pb06_tmexcusas / obj_tmexcusas)

#02.05 ---> Importación el objeto TMNIVELES (pb07_tmniveles / obj_tmniveles)

#02.06 ---> Importación el objeto TMUSUARIOS (pb08_tmusuarios / obj_tmusuarios)

#02.21 ---> Importación el objeto PROCESO01 (obj_proceso01)
#           Controlador de para el proceso: Registrar un Retardo
from blueprints.bp21_proceso01 import  obj_proceso01
```

Ojo después de este este punto

5.3.- Registrar o Crear la ruta asociada al controlador para el Registrar un Retardo (PROCESO01). En este momento se construye la ruta llamada "/proceso01" a través de la cual se tiene acceso a las rutas:

- **/proceso01/paso01** --→ **Cargar paso01.html** (Formulario para leer el número de cédula del empleado)
- **/proceso01/paso02 - POST** -→ Buscar los datos (empleado, cargo, status y excusas) y **Carga paso03.html** (Formulario para capturar el retardo del empleado – Código del retardo, fecha y hora)
- **/proceso01/paso04** - POST --→ **Registrar los datos del** retardo del empleado
- **/proceso01/limpiar** -→ **Cargar paso01.html** (Formulario para leer el número de cédula del empleado)

Las cuales está declaradas en el programa bp21_proceso01.

```
mi_app.register_blueprint(obj_proceso01, url_prefix='/proceso01')
```

```
#05.07.- RUTA MODULO TMCARGOS ----> /TMCARGOS
#        Registrando el Blueprints de tmcargos
#.....
mi_app.register_blueprint(obj_tmcargos, url_prefix='/tmcargos')


#05.08.- RUTA MÓDULO TMEMPLEADOS ----> /TMEMPLEADOS
#        Registrando el Blueprints de tmempleados
#.....


#05.09.- RUTA MÓDULO TMEXCUSAS ----> /TMEXCUSAS
#        Registrando el Blueprints de tmexcusas
#.....


#05.10.- RUTA MÓDULO TMNIVELES ----> /TMNIVELES
#        Registrando el Blueprints de tmniveles
#.....


#05.11.- RUTA MÓDULO TMUSUARIOS ----> /TMUSUARIOS
#        Registrando el Blueprints de tmusuarios
#.....


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

Ojo después de este este punto

Paso 6: Ejecutar el proyecto

```
C:\jardin>pjardin\scripts\activate

(pjardin) C:\jardin>py app.py
```

