



# PERSISTENCIA DE DATOS CON Python

# 3



[www.sena.edu.co](http://www.sena.edu.co)



# CONECTANDO CON PROCEDIMIENTOS ALMACENADOS

[igalindos@sena.edu.co](mailto:igalindos@sena.edu.co) 2023



# SOFTWARE RECOMENDADO



**MARIADB SERVER**

**POSTGRES SERVER**

**SQLITE SERVER**

**REPOSITORIO GITHUB**



# AGENDA DE TEMAS



- PROCEDIMIENTOS ALMACENADOS
- EJECUTANDO PROCEDIMIENTOS ALMACENADOS
- EJECUTANDO PROCEDIMIENTOS ALMACENADOS DESDE PYTHON
- CREACIÓN DE USUARIOS DE BASE DE DATOS
- OTORGAR Y REVOCAR PERMISOS A USUARIOS DE BASE DE DATOS
- CREANDO UN LOGIN CON AUTENTICACIÓN DE BASE DE DATOS EN FLASK
- CREANDO DESENCADENADORES (TRIGGER)
- CREANDO LOG DE AUDITORÍA





# INSTRUCCIONES PARA DESARROLLAR EL TALLER



- EL USUARIO ROOT DEBE TENER CLAVE
- EL USUARIO ROOT SOLO SE USA PARA CREAR OBJETO DE BASE DE DATOS (DDL) O PARA OTORGAR Y REVOCAR PERMISOS (DCL)
- NO PERMITIR QUE EL USUARIO ROOT SE AUTENTIQUE EN LA APLICACIÓN FLASK.
- NO SE PERMITE UTILIZAR BASE DE DATOS SQLITE
- NO SE PERMITE QUE EL DESARROLLADOR FRONTEND CONOZCA LA TABLA DE BASE DE DATOS QUE ESTÁ AFECTANDO.
- EL DESARROLLADOR FRONTEND DEBE SOLO EJECUTAR PROCEDIMIENTOS.
- TODOS LOS PROCEDIMIENTOS DEBEN TENER AL MENOS UN PARÁMETRO INOUT Y CONTROLAR ERRORES MEDIANTE EXCEPCIONES.
- TODOS LOS ENLACES “REGRESAR” DEBE IR AL ENDPOINT RAÍZ.
- EL LOG DE APLICACIONES DEBE SER CREADO POR FECHA EN LA CARPETA UTILS.





# CREANDO PROCEDIMIENTOS ALMACENADOS



MANUAL




# CREANDO PROCEDIMIENTOS ALMACENADOS



## CREAR LA TABLA CIUDADES

```
CREATE TABLE CIUDAD(  
  IDCIUDAD INT PRIMARY KEY AUTO_INCREMENT,  
  NOMCIUDAD TEXT  
);
```

 ciudad
123 <b>IDCIUDAD</b>
A-Z NOMCIUDAD



# CREANDO PROCEDIMIENTOS ALMACENADOS



## CREAR PROCEDIMIENTO

```
delimiter $$
create or replace procedure GestionarCiudad(
in paccion enum('I','A','L','D'),
in pidciudad int,
in pnom text
)
begin
    if paccion='I' then
        insert into ciudad(nomciudad) values(pnom);
    end if;
    if paccion='A' then
        update ciudad set nomciudad=pnom where idciudad=pidciudad;
    end if;
    if paccion='L' then
        if pidciudad is not null then
            select * from ciudad where idciudad=pidciudad ;
        else
            select * from ciudad;
        end if;
    end if;
end if;
```

```
if paccion='D' then
    if pidciudad is not null then
        delete from ciudad where
            idciudad=pidciudad ;
    else
        delete from ciudad;
    end if;
end if;
end $$
delimiter ;
```

```
call Ejemplo.GestionarCiudad('I',null, 'MEDELLIN');
call Ejemplo.GestionarCiudad('A',1, 'BOGOTA');
call Ejemplo.GestionarCiudad('L',null,null);
call Ejemplo.GestionarCiudad('L',2,null);
call Ejemplo.GestionarCiudad('D',2,null);
```



# LLAMAR PROCEDIMIENTOS ALMACENADOS DESDE PYTHON



## CREAMOS EL ARCHIVO app.py

```
from flask import Flask, jsonify, request
import json
from flask_mysql import MySQL

app=Flask(__name__)
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ''
app.config['MYSQL_DB'] = 'ejemplo'
mysql = MySQL(app)
if __name__ == '__main__':
    app.run(debug=True, port=8000, host='0.0.0.0')
```

## ACTUALIZAMOS UNA CIUDAD

```
@app.route("/c/a/<id>/<nom>")
def EditarCiudad(id,nom):
    cur = mysql.connection.cursor()
    try:
        cur.callproc('GestionarCiudad', ['A', id, nom])
        mysql.connection.commit()
        return "200"
    except Exception as e:
        print("Fallo", e)
        return "FALLO"
```

## INSERTAMOS UNA CIUDAD

```
@app.route("/c/i/<nom>")
def CrearCiudad(nom):
    cur = mysql.connection.cursor()
    try:
        cur.callproc('GestionarCiudad', ['I', None, nom])
        mysql.connection.commit()
        return "200"
    except Exception as e:
        print("Fallo", e)
        return "FALLO"
```

## LISTAMOS TODAS LAS CIUDADES

```
@app.route("/c/l/<int:id>")
def ListarCiudad(id):
    cur = mysql.connection.cursor()
    try:
        if id==0 :
            cur.callproc('GestionarCiudad', ['L', None, None])
        else:
            cur.callproc('GestionarCiudad', ['L', id, None])
        k=list(cur.fetchall())
        return k
    except Exception as e:
        print("Fallo", e)
        return "FALLO"
```

# PROCEDIMIENTO ALMACENADO PARA EJECUTAR SENTENCIAS DINAMICAMENTE



```

c:\XAMPP for Windows - MYSQL -u root -p
# MYSQL -u root -p 1
Enter password: ****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 18
Server version: 10.4.32-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use mysql 2
Database changed
MariaDB [mysql]> DELIMITER $$ 3
MariaDB [mysql]> CREATE OR REPLACE PROCEDURE ChangeUserPassword(usuario TEXT, clave TEXT) 4
-> BEGIN
->     DECLARE sm TEXT;
->     -- Concatenar la consulta SQL correctamente
->     SET sm = CONCAT('ALTER USER ', usuario, '@'localhost' IDENTIFIED BY ', clave, ');
->     -- Preparar la consulta SQL
->     PREPARE stmt FROM sm;
->     -- Ejecutar la consulta preparada
->     EXECUTE stmt;
->     -- Liberar la consulta preparada
->     DEALLOCATE PREPARE stmt;
-> END$$
Query OK, 0 rows affected (0.173 sec)

MariaDB [mysql]> DELIMITER ; 5
MariaDB [mysql]> CALL ChangeUserPassword('jgalindos','prueba123'); 6
Query OK, 0 rows affected (0.057 sec)

MariaDB [mysql]> _
```

# TE RETO



# PRACTICA TRABAJO AUTÓNOMO



**Crear una tabla llamada JEFE con los datos de employees relacionándolo entre las dos tablas y construir un procedimiento CRUD para la tabla employees, controlando los errores que pudieran suceder.**



**MYSQL**

**PROCEDIMIENTOS**





# PERMISOS DCL



# OTORGANDO Y REVOCANDO PERMISOS A USUARIOS MYSQL



## CREAMOS USUARIO jgalindos CON EL USUARIO HR

```
XAMPP for Windows - mysql -u root -p

yayof@DESKTOP-09G6PMC c:\xampp8
# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 25
Server version: 10.4.32-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create user 'jgalindos'@'localhost' identified by 'prueba1234';
Query OK, 0 rows affected (0.118 sec)

MariaDB [(none)]>
```

## INGRESAMOS CON EL USUARIO jgalindos

```
XAMPP for Windows - mysql -u jgalindos -p

# mysql -u jgalindos -p 1
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 28
Server version: 10.4.32-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases; 2
+-----+
| Database |
+-----+
| hr       |
| information_schema |
+-----+
2 rows in set (0.001 sec)

MariaDB [(none)]> use hr 3
Database changed
MariaDB [hr]> show tables; 4
+-----+
| Tables_in_hr |
+-----+
| employees    |
| regions      |
+-----+
2 rows in set (0.001 sec)

MariaDB [hr]>
```

## OTORGAMOS PERMISO DE SELECT EN EMPLOYEES Y REGIONS

```
XAMPP for Windows - mysql -u root -p

Setting environment for using XAMPP for Windows.
yayof@DESKTOP-09G6PMC c:\xampp8
# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 26
Server version: 10.4.32-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> grant select on hr.employees to jgalindos;
Query OK, 0 rows affected (0.112 sec)

MariaDB [(none)]> grant select on hr.regions to jgalindos;
Query OK, 0 rows affected (0.110 sec)

MariaDB [(none)]>
```



# OTORGANDO Y REVOCANDO PERMISOS A USUARIOS MYSQL



SE QUIERE INSERTAR UNA NUEVA REGIÓN SIN TENER EL PERMISO PARA INSERTAR

```
c:\XAMPP for Windows - mysql -u jgalindos -p

MariaDB [hr]> SELECT CURRENT_USER; 1
+-----+
| CURRENT_USER |
+-----+
| jgalindos@localhost |
+-----+
1 row in set (0.000 sec)

MariaDB [hr]> DESCRIBE REGIONS; 2
+-----+-----+-----+-----+-----+-----+
| Field      | Type                | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| region_id  | int(11) unsigned    | NO   | PRI | NULL    |       |
| region_name| varchar(25)         | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.003 sec)

MariaDB [hr]> INSERT INTO REGIONS(REGION_NAME) VALUES('SELVATICA'); 3
ERROR 1142 (42000): INSERT command denied to user 'jgalindos'@'localhost' for table 'hr`.`regions`
MariaDB [hr]>
```

# OTORGANDO Y REVOCANDO PERMISOS A USUARIOS MYSQL



OTORGAMOS EL PERMISO PARA INSERTAR EN LA TABLA REGIONS

```
C:\> XAMPP for Windows - mysql -u root -p

MariaDB [(none)]> grant insert on hr.regions to jgalindos;
Query OK, 0 rows affected (0.103 sec) 1

MariaDB [(none)]>
```



SE INTENTA DE NUEVO INSERTAR EN LA TABLA REGIONS

```
C:\> XAMPP for Windows - mysql -u jgalindos -p

MariaDB [hr]> INSERT INTO REGIONS(REGION_NAME) VALUES('SELVATICA'); 1
Query OK, 1 row affected, 1 warning (0.068 sec)

MariaDB [hr]> SELECT * FROM REGIONS; 2

+-----+-----+
| region_id | region_name |
+-----+-----+
| 0         | SELVATICA   | 3
| 1         | Europe      |
| 2         | Americas    |
| 3         | Asia        |
| 4         | Middle East and Africa |
+-----+-----+
5 rows in set (0.000 sec)

MariaDB [hr]>
```



# TE RETO



# PRACTICA TRABAJO AUTÓNOMO



Con el usuario jgalindos, se quiere introducir la región llamada “ANTÁRTICA”, luego se desea cambiar el registro creado por el nombre “ÁRTICO” y por último borrar este registro creado. Se debe realizar un video del paso a paso del proceso realizado y subirlo a Youtube.



MYSQL

PROCEDIMIENTOS



# PROYECTO FLASK PARA LOGIN

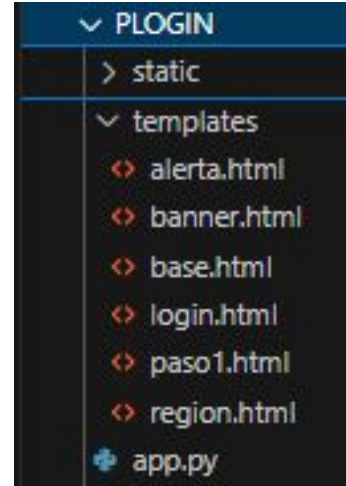
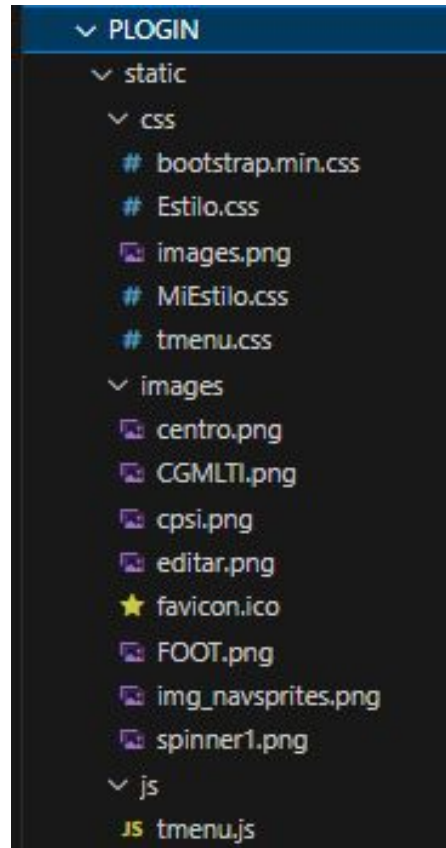
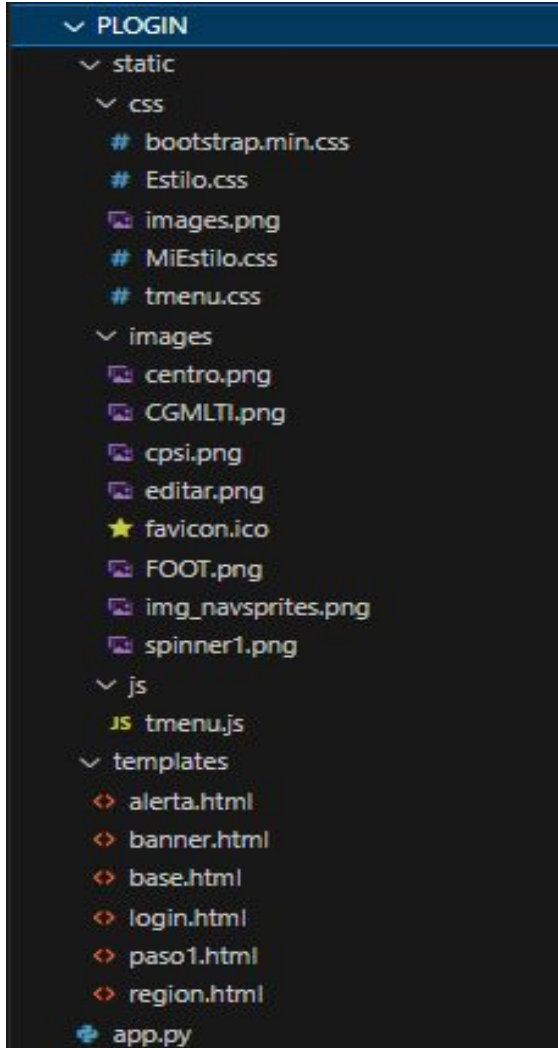


MANUAL





# PREPARANDO UN PROYECTO FLASK EN PYTHON



## PERSISTENCIA DE DATOS CON Python

@SENAComunica [www.sena.edu.co](http://www.sena.edu.co)



## CONECTANDO CON PROCEDIMIENTOS ALMACENADOS

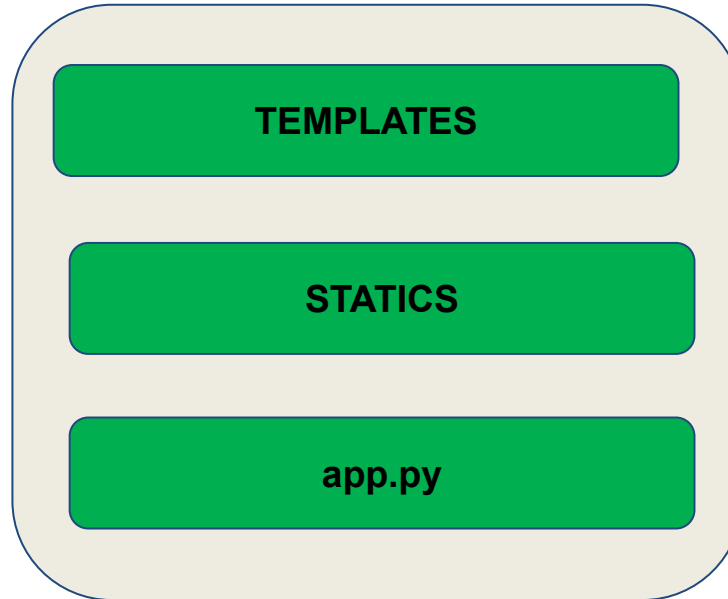
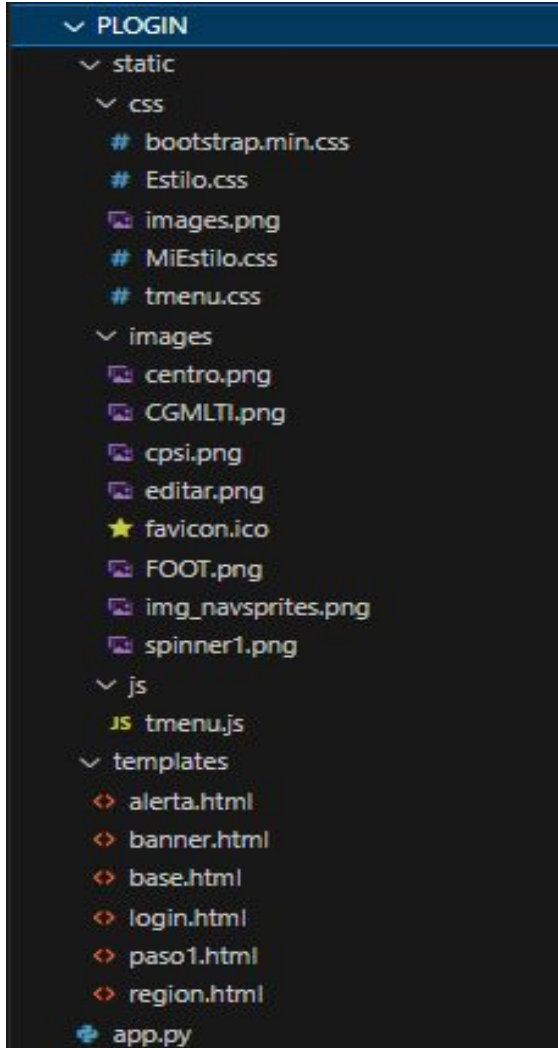
[jgalindos@sena.edu.co](mailto:jgalindos@sena.edu.co) 2023



@SENAComunica  
[www.sena.edu.co](http://www.sena.edu.co)



# PREPARANDO UN PROYECTO FLASK EN PYTHON



# CREANDO LA PLANTILLA base.html



```
<!doctype html>
<html>
  <head>
{% block head %}<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWFSpd3yD65VohhpuuCOMLASjC" crossorigin="anonymous">
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
<link href="{url_for('static',filename='css/MiEstilo.css')}" rel="stylesheet">
<meta name="viewport" content="width=device-width, initial-scale=1">
<script src="https://kit.fontawesome.com/a076d05399.js" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,100;0,300;0,400;0,500;0,700;0,900;1,100;1,300;1,400;1,500;
1,700;1,900&display=swap" rel="stylesheet">
<script src="dist/sweetalert.min.js"></script>
<link rel="stylesheet" type="text/css" href="dist/sweetalert.css">
<script src="{url_for('static',filename='js/MiJs.js')}"></script>
{% endblock %}
```

```
</head>
<body>
  {% include 'banner.html' %}
</div>

  <div style="overflow:auto" class="sena-p7
sena-card3-none">
    {% block content %}
    q{% endblock content %}
  </div>

  <br>
  <br>
</body>
<link
href="{url_for('static',filename='css/bootstr
ap.min.css')}" rel="stylesheet">

</html>
```

# CREANDO LA PLANTILLA alerta.html



```
{% extends "base.html" %}
{% block content %}

    <center>
        <br><br><br><br><br><br>
        <div class="coard mx-auto border" style="width:
22rem;border:2px;-webkit-box-shadow: 10px 10px 5px 0px rgba(0,0,0,0.75);
-moz-box-shadow: 10px 10px 5px 0px rgba(0,0,0,0.75);
box-shadow: 10px 10px 5px 0px rgba(0,0,0,0.75);">

            <div class="card-header w3-black">
                <h5 class="card-title"><font color="yellow">ALERTA</font></h5>
            </div>
            <div class="card-body">
                <font color=red size=4 class="caret-sombra-red"><b>{{
msgito}}</font></center>
            <br><br>
        </div>
        <br><br>
    </div>
    <br><br>
</pre>
```

```
<script>
    function Voy(){
        {% if regreso%}
            location.href="{{regreso}}"
        {% else %}
            location.href="/"
        {% endif %}
    }
    setTimeout('Voy()',5000)
</script>
{% endblock content %}
```

# CREANDO LA PLANTILLA banner.html



```
<center>
  <div style="width:70%;overflow:auto" class="w3-w3-green mx-auto">
    <br>
  </div>
```



# CREANDO LA PLANTILLA paso1.html



```
{% extends "base.html" %}
{% block content %}
<A href="/">REGRESAR</A>
<div style="width:70%;padding: 10,10,10,10;">
<center>
  <a href="/region" class="btn btn-primary" id="bt" style="width: 40%;padding-bottom:
10px;">Regiones</a><br>
  <a href="/region" class="btn btn-primary" id="bt" style="width: 40%;padding: 10,10,10,10;">Paises</a><br>

</center>
</div>
{% endblock content %}
```

# CREANDO LA PLANTILLA region.html



```
{% extends "base.html" %}
{% block content %}<center><BR>
    <div class="coard  mx-auto border" style="width: 50rem;">
<table class="table table-striped">
    <th>NOMBRE REGION</th></tr>
{% for datos in cadena %}
<tr><td>{{datos[1]}}</td>
</tr>
{% endfor %}
</table>
</div>
<A class="btn btn-primary" href="/pas01">REGRESAR</A>
{% endblock content %}
```

# CREANDO LA PLANTILLA login.html



```
{% extends "base.html" %}
{% block content %}
<br>
<div class="coard mx-auto border" style="width: 22rem;">
  <div class="card-header w3-black">
    <h5 class="card-title"><font color="yellow">INGRESO</font></h5>
  </div>
  <div class="card-body">
    <form method="post" action="/v">
      <table class="table table-striped">
        <tr>
          <td><label class="form-label">Usuario:</label></td>
          <td><input class="form-control" type="text" id="usua" name="usua"></td>
        </tr>
        <tr>
          <td><label class="form-label">Clave:</label></td>
          <td><input class="form-control" type="password" id="pw" name="pw"></td>
        </tr>
      </table>
      <center><input class="btn btn-outline-primary col-10" type="submit" value="Ingresar">
    </form>
  </div>
</div>
{% endblock content %}
```

# CONSTRUYENDO LOS ENDPOINT DE LA APLICACIÓN



## CREAMOS EL ARCHIVO app.py

```
from flask import Flask, jsonify, request
import json
from flask_mysql import MySQL

app=Flask(__name__)
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'prueba'
app.config['MYSQL_PASSWORD'] = 'prueba'
app.config['MYSQL_DB'] = 'ejemplo'
mysql = MySQL(app)

if __name__=='__main__':
    app.run(debug=True,port=8000,host='0.0.0.0')
```

## ENDPOINT RAIZ Y PASO1

```
@app.route("/")
def Raiz():
    return render_template("login.html")

@app.route("/paso1")
def Paso1():
    try:
        cur = mysql.connection.cursor()
        return render_template("paso1.html")
    except Exception as e:
        msgito="NO HA VALIDADO CREDENCIALES"
        regreso="/"
        return render_template("alerta.html", msgito=msgito, regreso=regreso)
```

## ENDPOINT VALIDACION

```
@app.route("/v", methods=['POST'])
def Raiz1():
    if request.method == 'POST':
        usua = request.form.get('usua')
        pw = request.form.get('pw')
        try:
            app.config['MYSQL_HOST'] = 'localhost'
            app.config['MYSQL_USER'] = usua
            app.config['MYSQL_PASSWORD'] = pw
            app.config['MYSQL_DB'] = 'hr'
            cur = mysql.connection.cursor()
            # cur.execute("select count(*) from regions")
            # cadena=cur.fetchall()
            # return render_template("region.html", cadena=cadena)
            msgito="BIENVENIDO"
            regreso="/paso1"
            return render_template("alerta.html", msgito=msgito, regreso=regreso)
        except Exception as e:
            msgito="USUARIO O CREDENCIALES NO VALIDOS"
            regreso="/"
            return render_template("alerta.html", msgito=msgito, regreso=regreso)

    return usua
```



# CONSTRUYENDO LOS ENDPOINT DE LA APLICACIÓN



```
@app.route("/region")
def Region():
    try:
        cur = mysql.connection.cursor()
        cur.execute("select * from regions")
        cadena=cur.fetchall()
        return render_template("region.html",cadena=cadena)
    except Exception as e:
        msgito="NO TIENE ACCESO"
        regreso="/paso1"
        return render_template("alerta.html", msgito=msgito,regreso=regreso)

{% endblock content %}
```

# UTILIZANDO LA APLICACIÓN



END POINT /



**INGRESO**

Usuario:

Clave:



END POINT /v



END POINT /paso1



[REGRESAR](#)

[Regiones](#)

[Paises](#)

END POINT /region

NOMBRE REGION
SELVATICA
Europe
Americas
Asia
Middle East and Africa

[REGRESAR](#)



# TE RETO



# PRACTICA TRABAJO AUTÓNOMO



**Construya un template llamado empleados que maneje los datos de la tabla “employees”, mediante procedimientos CRUD, que pueda seleccionar e insertar el usuario cramirez y seleccionar, actualizar y borrar al usuario mcardenas**







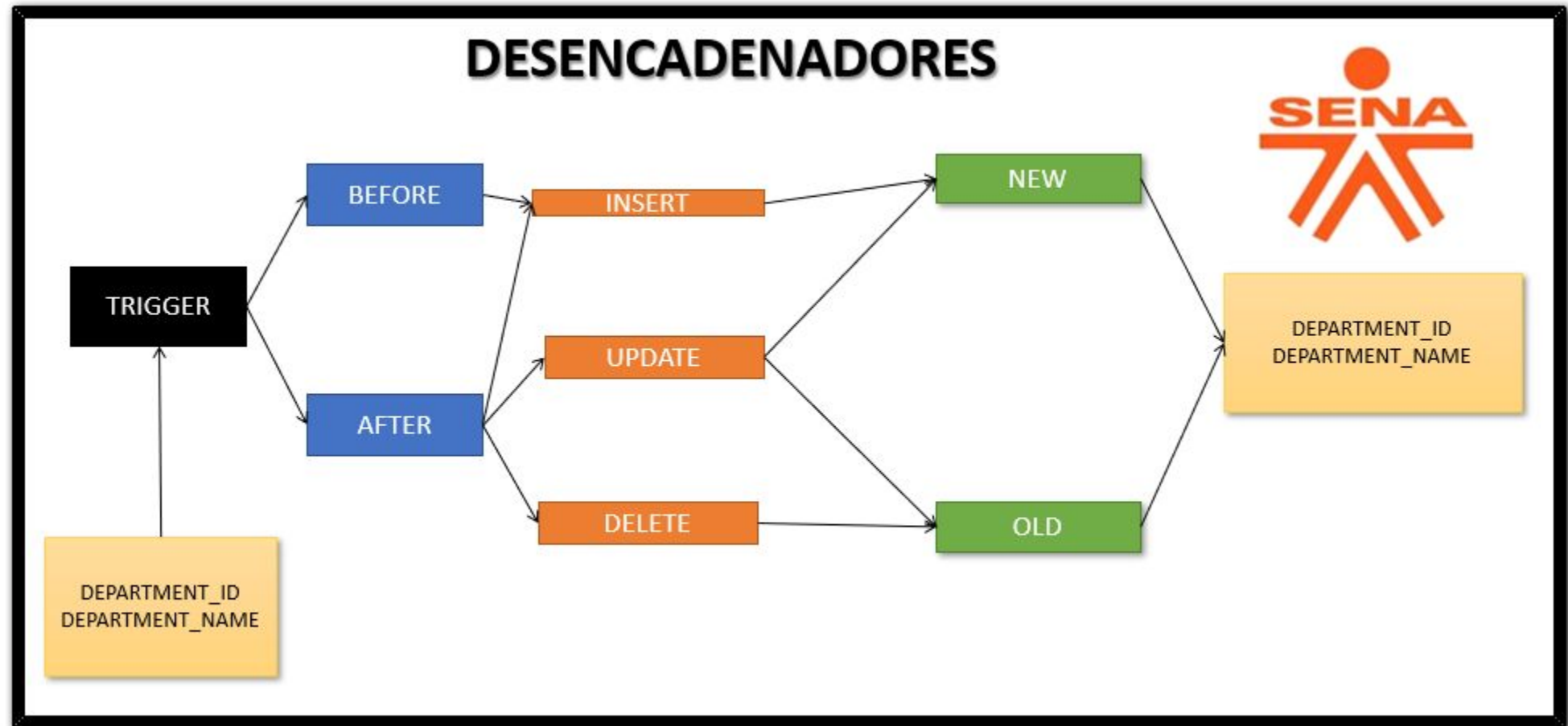
# DESENCADENADORES LOG DE TRANSACCIONES



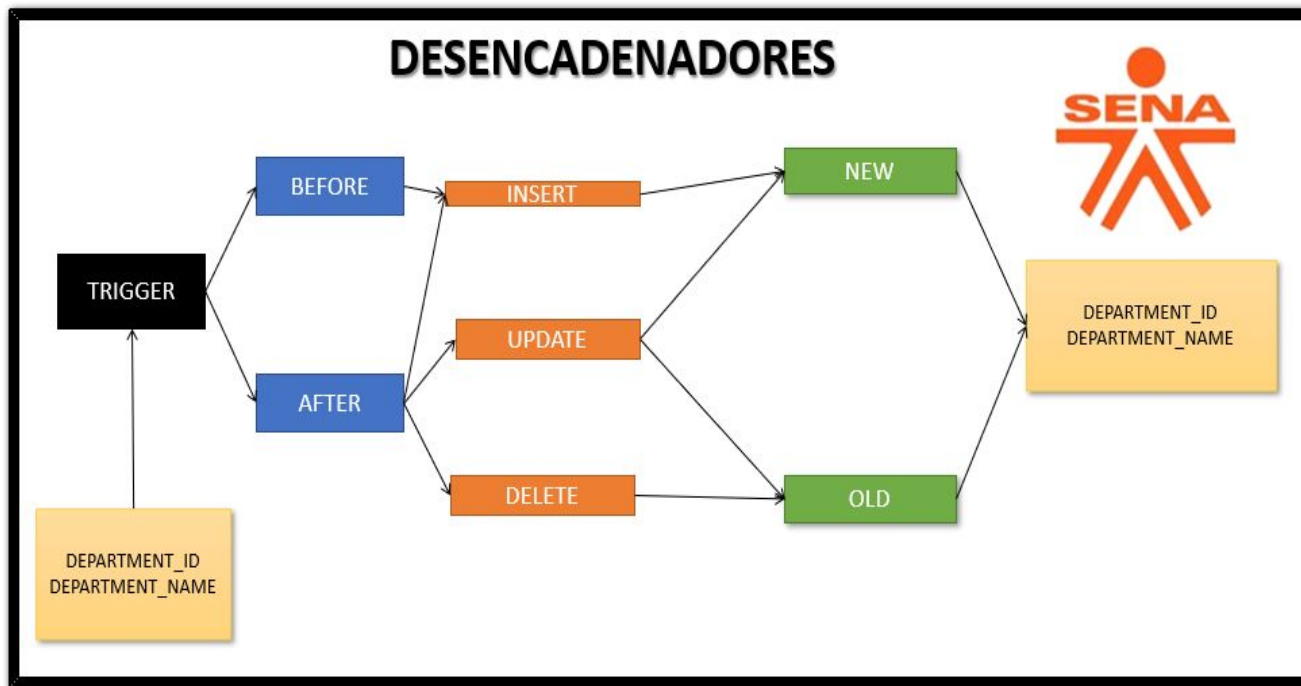
MANUAL



# DESENCADENADORES



# DESENCADENADORES



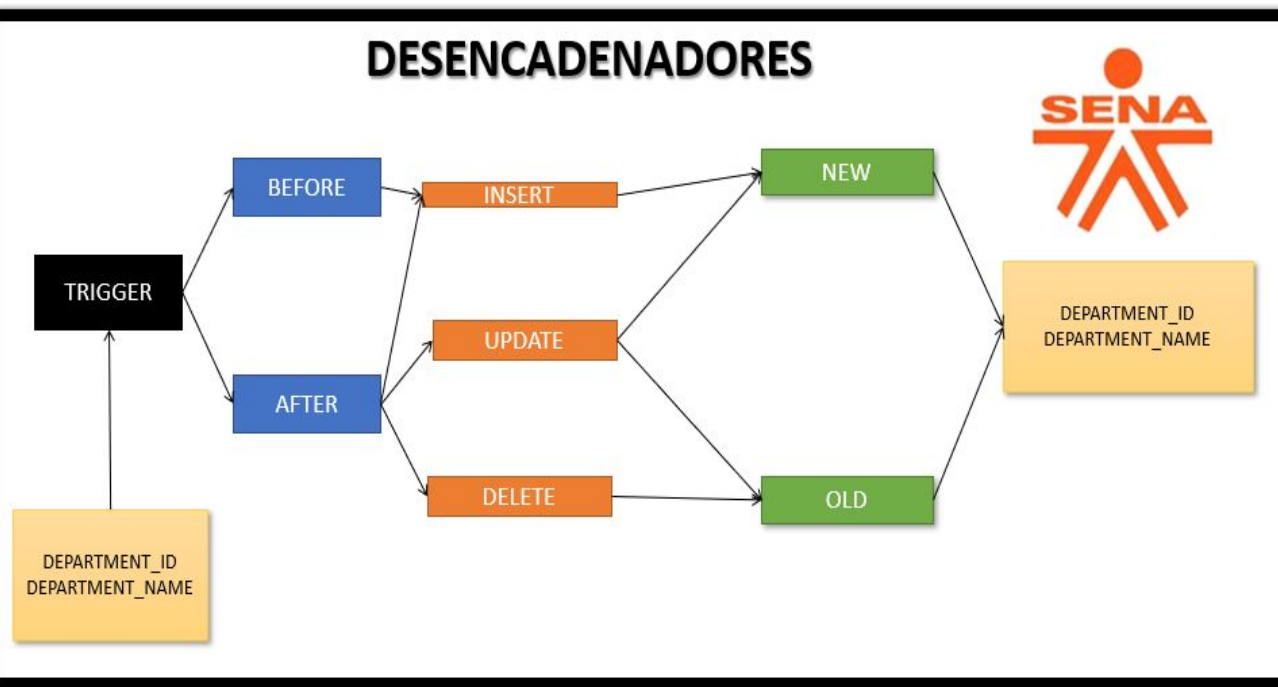
Los desencadenadores son sentencias que utilizan para la gestión de la información en una base de datos.

La incorporación de los TRIGGER se comenzó a utilizar desde la versión Mysql 5.0.2, son objetos con nombre asociado a una tabla y es activada cuando se realiza un evento INSERT, UPDATE o DELETE.

# DESENCADENADORES



## DESENCADENADORES



**CREATE TRIGGER Nombre Momento  
Evento ON nombre\_tabla**

**FOR EACH ROW**

**BEGIN**

**... Cuerpo o llamada**

**END;**



# DESENCADENADORES



**Momento**: **BEFORE** antes de grabar el registro en la tabla, **AFTER** después de grabar el registro en la tabla.

**Evento**: **INSERT** un nuevo registro, **UPDATE** actualizar el registro, **DELETE** borra un registro.

Evento	Toma de datos
INSERT	new.nombredecampodelatabla
UPDATE	new.nombredecampodelatabla old.nombredecampodelatabla
DELETE	old.nombredecampodelatabla

# DESENCADENADORES INSERT



```
XAMPP for Windows - mysql -u root -p
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use hr 1
Database changed
MariaDB [hr]> CREATE TABLE AUDITOR( 2
    -> FECHA DATETIME DEFAULT CURRENT_TIME,
    -> ACCION CHAR(1),
    -> USUARIO VARCHAR(30) DEFAULT CURRENT_USER,
    -> TABLA TEXT
    -> );
Query OK, 0 rows affected (0.164 sec)

MariaDB [hr]> DELIMITER $$ 3
MariaDB [hr]> CREATE OR REPLACE TRIGGER TGREGION AFTER INSERT ON REGIONS FOR EACH ROW 4
    -> BEGIN
    ->     INSERT INTO AUDITOR(ACCION,TABLA) VALUES('I','REGIONES');
    -> END $$
Query OK, 0 rows affected (0.210 sec)

MariaDB [hr]> DELIMITER ; 5
MariaDB [hr]>
MariaDB [hr]>
```

# DESENCADENADORES INSERT



```
XAMPP for Windows - mysql -u root -p

MariaDB [hr]> INSERT INTO REGIONS VALUES(22, 'ALTO MAGDALENA');
Query OK, 1 row affected (0.215 sec)

MariaDB [hr]> SELECT * FROM AUDITOR;

+-----+-----+-----+-----+
| FECHA          | ACCION | USUARIO          | TABLA    |
+-----+-----+-----+-----+
| 2024-11-16 12:00:37 | I      | root@localhost   | REGIONES |
+-----+-----+-----+-----+

1 row in set (0.000 sec)

MariaDB [hr]>
```



# DESENCADENADORES INSERT



40

JOSE FERNANDO GALINDO SUAREZ

C:\> XAMPP for Windows - mysql -u jgalindos -p

Setting environment for using XAMPP for Windows.

yayof@DESKTOP-09G6PMC c:\xampp8

# mysql -u jgalindos -p

1

Enter password: \*\*\*\*\*

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 9

Server version: 10.4.32-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use hr

2

Database changed

MariaDB [hr]> INSERT INTO REGIONS VALUES(23,'BAJO MAGDALENA');

3

Query OK, 1 row affected (0.126 sec)

MariaDB [hr]>



# DESENCADENADORES INSERT



41

JOSE FERNANDO GALINDO SUAREZ

C:\> XAMPP for Windows - mysql -u root -p

MariaDB [hr]> SELECT \* FROM AUDITOR; 1

FECHA	ACCION	USUARIO	TABLA
2024-11-16 12:00:37	I	root@localhost	REGIONES
2024-11-16 12:04:27	I	jgalindos@localhost	REGIONES

2

2 rows in set (0.001 sec)

MariaDB [hr]>

# DESENCADENADORES UPDATE



42

```
XAMPP for Windows - mysql -u root -p
MariaDB [HR]> ALTER TABLE AUDITOR ADD (DESCRIPCION TEXT);
Query OK, 0 rows affected (0.210 sec)
Records: 0 Duplicates: 0 Warnings: 0

MariaDB [HR]> DESCRIBE AUDITOR;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default      | Extra |
+-----+-----+-----+-----+-----+-----+
| FECHA      | datetime  | YES  |     | curtime()    |       |
| ACCION     | char(1)   | YES  |     | NULL         |       |
| USUARIO    | varchar(30)| YES  |     | current_user()|       |
| TABLA      | text      | YES  |     | NULL         |       |
| DESCRIPCION| text      | YES  |     | NULL         |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.003 sec)

MariaDB [HR]>
```

```
XAMPP for Windows - mysql -u root -p
MariaDB [HR]> CREATE OR REPLACE TRIGGER TREGIONU
-> AFTER UPDATE ON REGIONS
-> FOR EACH ROW
-> BEGIN
->   INSERT INTO AUDITOR(ACCION, TABLA, DESCRIPCION)
->   VALUES('U', CONCAT('REGIONS [' , OLD.REGION_ID, ']'), CONCAT(OLD.REGION_NAME, '-', NEW.REGION_NAME
));
-> END $$
Query OK, 0 rows affected (0.152 sec)

MariaDB [HR]>
MariaDB [HR]> DELIMITER ;
MariaDB [HR]>
```

```
XAMPP for Windows - mysql -u jgalindos -p
MariaDB [HR]> UPDATE REGIONS SET REGION_NAME='BAJO MAGDALENA' WHERE REGION_ID=23;
Query OK, 1 row affected (0.152 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
XAMPP for Windows - mysql -u root -p
MariaDB [HR]> SELECT * FROM AUDITOR;
+-----+-----+-----+-----+-----+
| FECHA      | ACCION | USUARIO      | TABLA      | DESCRIPCION |
+-----+-----+-----+-----+-----+
| 2024-11-16 12:00:37 | I      | root@localhost | REGIONES   | NULL        |
| 2024-11-16 12:04:27 | I      | jgalindos@localhost | REGIONES   | NULL        |
| 2024-11-16 13:03:55 | U      | jgalindos@localhost | REGIONS [23] | MAGDALENA MEDIO-BAJO MAGDALENA |
+-----+-----+-----+-----+-----+
3 rows in set (0.000 sec)

MariaDB [HR]>
```



# DESENCADENADORES DELETE



43

JOSE FERNANDO GALINDOS SUAREZ

```
XAMPP for Windows - mysql -u root -p
MariaDB [HR]> DELIMITER $$
MariaDB [HR]> CREATE OR REPLACE TRIGGER TGRECIOND
-> AFTER DELETE ON REGIONS
-> FOR EACH ROW
-> BEGIN
->     INSERT INTO AUDITOR(ACCION, TABLA, DESCRIPCION)
->     VALUES('D', CONCAT('REGIONS [' , OLD.REGION_ID, ''])
->     ,OLD.REGION_NAME);
-> END $$
Query OK, 0 rows affected (0.194 sec)

MariaDB [HR]> DELIMITER ;
MariaDB [HR]>
```

```
XAMPP for Windows - mysql -u jgalindos -p
MariaDB [HR]> DELETE FROM REGIONS WHERE REGION_ID=23;
Query OK, 1 row affected (0.163 sec)

MariaDB [HR]>
```

```
XAMPP for Windows - mysql -u root -p
MariaDB [HR]> SELECT * FROM AUDITOR;
+-----+-----+-----+-----+-----+
| FECHA          | ACCION | USUARIO          | TABLA      | DESCRIPCION |
+-----+-----+-----+-----+-----+
| 2024-11-16 13:30:52 | I      | root@localhost   | REGIONES   | NULL        |
| 2024-11-16 13:30:54 | I      | root@localhost   | REGIONES   | NULL        |
| 2024-11-16 13:37:23 | U      | jgalindos@localhost | REGIONS [23] | BAJO MAGDALENA-MAGDALENA MEDIO |
| 2024-11-16 13:38:00 | D      | jgalindos@localhost | REGIONS [23] | MAGDALENA MEDIO |
+-----+-----+-----+-----+-----+
4 rows in set (0.000 sec)

MariaDB [HR]>
```

```
XAMPP for Windows - mysql -u jgalindos -p
MariaDB [HR]> SELECT * FROM REGIONS;
+-----+-----+
| region_id | region_name |
+-----+-----+
| 0         | SELVATICA   |
| 1         | Europe      |
| 2         | Americas    |
| 3         | Asia        |
| 4         | Middle East and Africa |
| 22        | ALTO MAGDALENA |
+-----+-----+
6 rows in set (0.000 sec)

MariaDB [HR]>
```

# TE RETO





# PRACTICA TRABAJO AUTÓNOMO



**Construya triggers que maneje los eventos insert, update y delete en la tabla employees y este guardados en la tabla “LOGAUDITOR” y que deje pista de auditoria en cada registro de la tabla employees.**



**MYSQL**

**PROCEDIMIENTOS**



# LOG DE AUDITORÍA



MANUAL



# LOG DE APLICACIONES



```
from flask import request
import logging
import os
from datetime import datetime

class Auditor():
    logger=None
    def __init__(self):

        fecha=datetime.now()
        fe=str(fecha.year)+str(fecha.month)+str(fecha.day)
        print(fe)
        os.makedirs('/log/'+fe,exist_ok=True)
        logger = logging.getLogger('werkzeug')
        self.logger =logger
        logging.basicConfig(format='%(asctime)s %(message)s ',filename='/log/'+fe+'/login.log',
encoding='utf-8',level=logging.ERROR)
        self.logger.setLevel(logging.ERROR)

    def logstart(self):
        return self.logger

    def registra(self,tipo,msg,usua="-"):
        client_ip = request.remote_addr
        if tipo==10:
            self.logger.debug(client_ip+' '+msg+' ['+usua+']')
        elif tipo==20:
            self.logger.info(client_ip+' '+msg+' '+usua)
        elif tipo==30:
            self.logger.warning(client_ip+' '+msg+' '+usua)
        elif tipo==40:
            print(client_ip+' '+msg+' '+usua)
            self.logger.error(client_ip+' '+msg+' ['+usua+']')
        elif tipo==50:
            self.logger.critical(client_ip+' '+msg+' '+usua)
```



```
@app.route("/v",methods=['POST'])
def Raiz1():
    if request.method == 'POST':
        usua = request.form.get('usua')
        pw = request.form.get('pw')
        try:
            app.config['MYSQL_HOST'] = 'localhost'
            app.config['MYSQL_USER'] = usua
            app.config['MYSQL_PASSWORD'] = pw
            app.config['MYSQL_DB'] = 'hr'
            cur = mysql.connection.cursor()
            # cur.execute("select count(*) from regions")
            # cadena=cur.fetchall()
            # return render_template("region.html", cadena=cadena)
            msgito="BIENVENIDO"
            regreso="/pasol"
            # logger.error('INFO: ingresa '+usua)
            Au.registra(40,msgito,usua )
            return render_template("alerta.html", msgito=msgito,regreso=regreso)
        except Exception as e:
            msgito="USUARIO O CREDENCIALES NO VALIDOS"
            regreso="/"
            usua=''
            Au.registra(40,msgito)
            return render_template("alerta.html", msgito=msgito,regreso=regreso)

    return usua
```

```
log > login.log
1 2024-11-20 09:07:16,712 ERROR: 127.0.0.1 USUARIO O CREDENCIALES NO VALIDOS -
2 2024-11-20 09:07:44,538 INFO: 127.0.0.1 BIENVENIDO root
3 2024-11-20 09:08:00,767 INFO: 127.0.0.1 Ingresa a regions root
4 2024-11-20 09:09:52,095 INFO: 127.0.0.1 Ingresa a regions root
5 2024-11-20 09:10:21,308 INFO: 127.0.0.1 BIENVENIDO jgalindos
```

LOGGER

# TE RETO





# PRACTICA TRABAJO AUTÓNOMO



**Registre todos los eventos desde el login y los CRUD realizados por los procedimientos ejecutados en log de la aplicación .**



**MYSQL**



# CURSORES



MANUAL



# CURSORES



## CURSORES

### Descripción de la práctica:

Esta práctica esta orientada al funcionamiento y uso de los cursores en las bases de datos, para esto se hace necesario conocer las etapas que se desarrollan al construir el cursor. Traemos un ejemplo donde construimos una tabla llamada "IT\_WORKER", que guardará las filas generadas por el curso "MICURSOR" que se ejecuta mediante el procedimiento "PRDEMOCURSOR", que extraerá el valor neto y el impuesto del IVA del salario de los programadores.



# CURSORES



JOSE FERNANDO CALINDO SIBARRIZ



Para esto, crearemos un procedimiento llamado “PR\_DEMOCURSOR” que contiene un cursor que carga las filas de la tabla EMPLOYEES que su JOB\_ID pertenezcan a “IT\_PROG”, así:

```
MariaDB [HR]> select EMPLOYEE_ID,CONCAT(LAST_NAME,' ',FIRST_NAME) EMPLEADO,SALARY from employees where job_id='IT_PROG';
```

EMPLOYEE_ID	EMPLEADO	SALARY
103	Hunold Alexander	9000.00
104	Ernst Bruce	6000.00
105	Austin David	4800.00
106	Pataballa Valli	4800.00
107	Lorentz Diana	4200.00

5 rows in set (0.00 sec)

Creamos la tabla “IT\_WORKER” solo la estructura

```
MariaDB [HR]> CREATE TABLE IT_WORKER AS  
-> select EMPLOYEE_ID,CONCAT(LAST_NAME,' ',FIRST_NAME) EMPLEADO,SALARY,0 IVA,0 NETO  
-> from employees where 1=2;  
Query OK, 0 rows affected (0.05 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

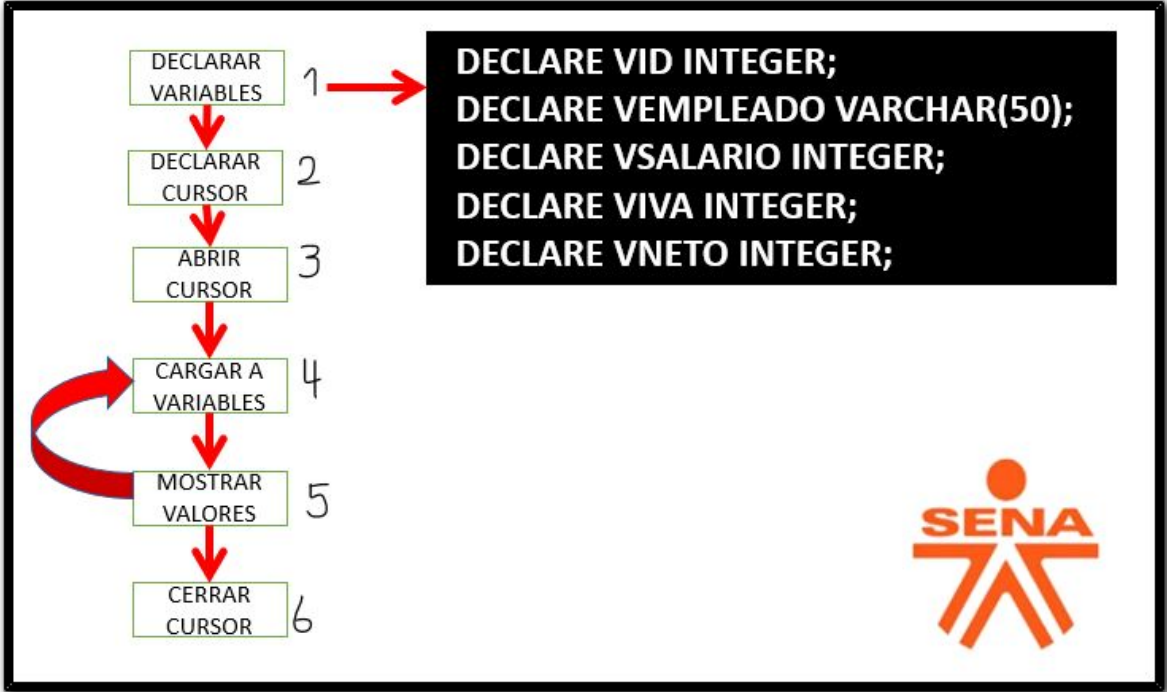
```
MariaDB [HR]> DESC IT_WORKER;
```

Field	Type	Null	Key	Default	Extra
EMPLOYEE_ID	int(11) unsigned	NO		NULL	
EMPLEADO	varchar(46)	YES		NULL	
SALARY	decimal(8,2)	NO		NULL	
IVA	int(1)	NO		NULL	
NETO	int(1)	NO		NULL	

5 rows in set (0.04 sec)

```
MariaDB [HR]> SELECT * FROM IT_WORKER;  
Empty set (0.00 sec)
```

La idea es de correr el procedimiento “PR\_DEMOCURSOR” y utilizar el cursor llamado “MICURSOR” que calcula el IVA y el valor neto de su salario.





# CURSORES



```
DECLARE MICURSOR CURSOR FOR  
SELECT EMPLOYEE_ID, CONCAT(LAST_NAME, '  
, FIRST_NAME) EMPLEADO, SALARY, 0 IVA, 0 NETO  
FROM EMPLOYEES  
WHERE JOB_ID='IT_PROG';
```

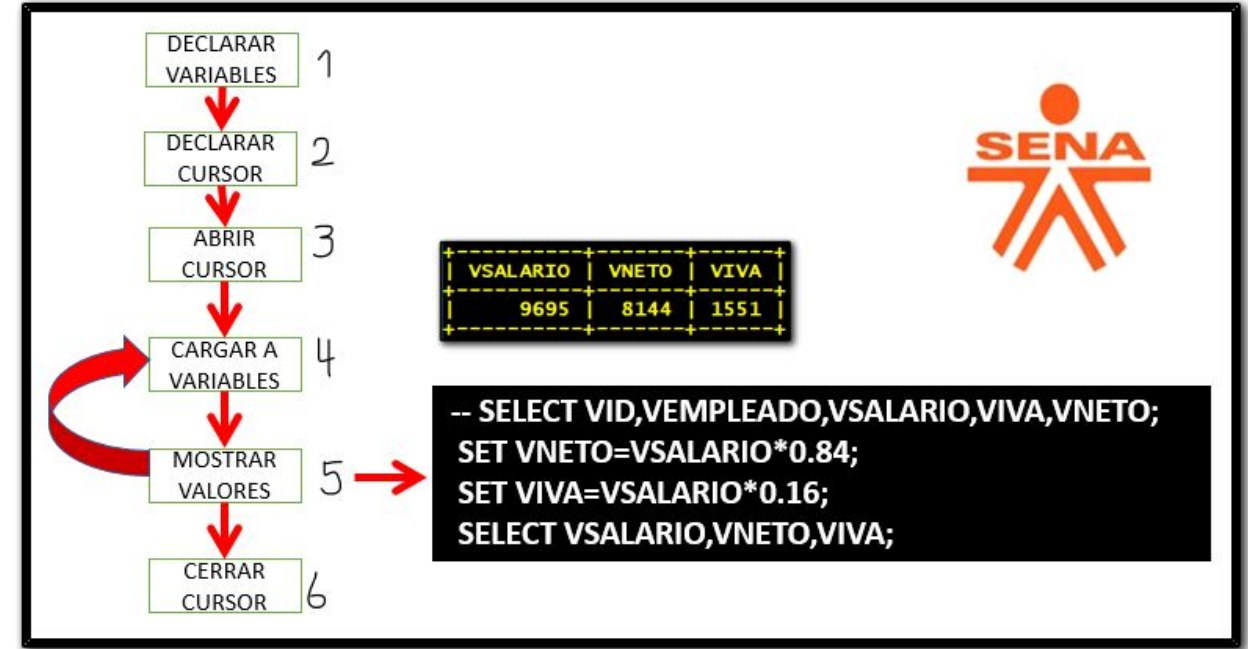
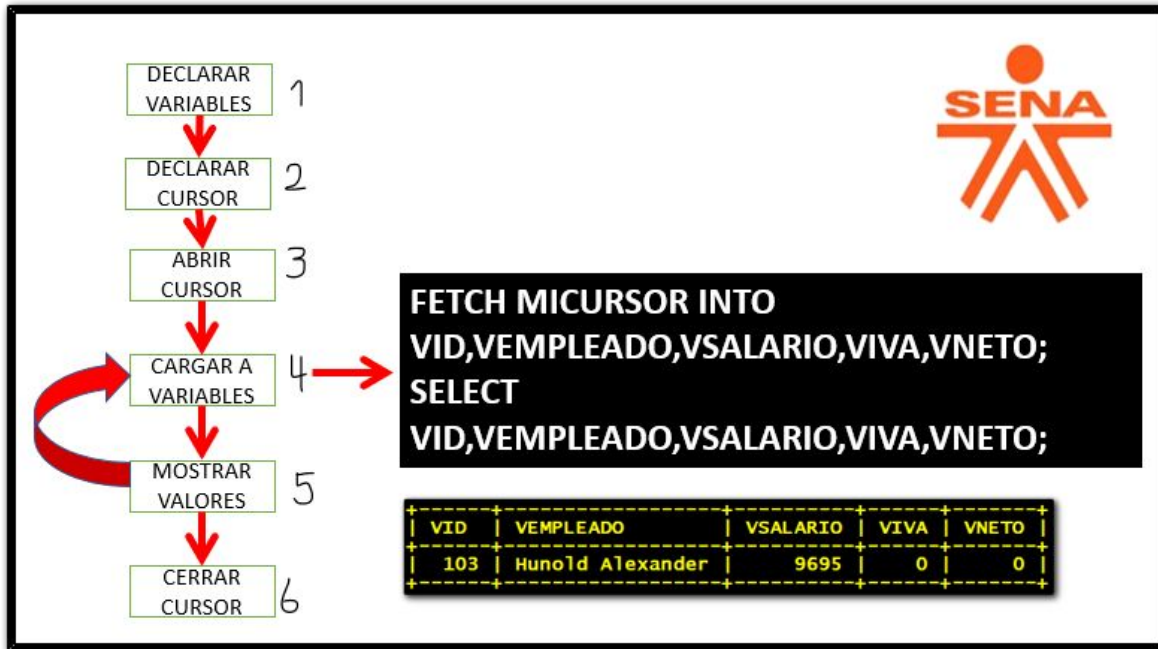
EMPLOYEE_ID	EMPLEADO	SALARY	IVA	NETO
103	Hunold Alexander	9695.00	0	0
104	Ernst Bruce	6464.00	0	0
105	Austin David	5170.00	0	0
106	Pataballa Valli	5170.00	0	0
107	Lorentz Diana	4524.00	0	0



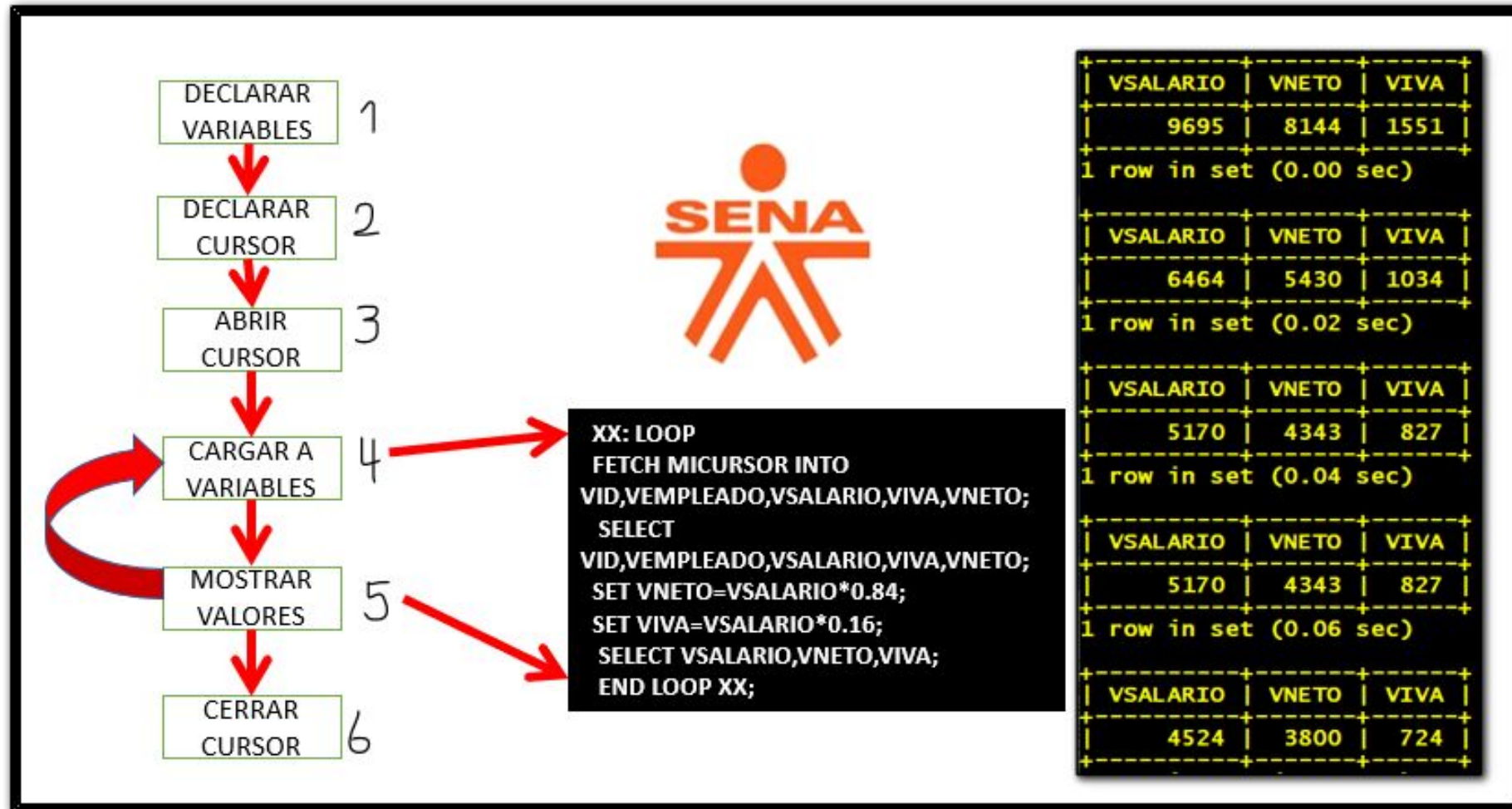
```
OPEN MICURSOR;
```



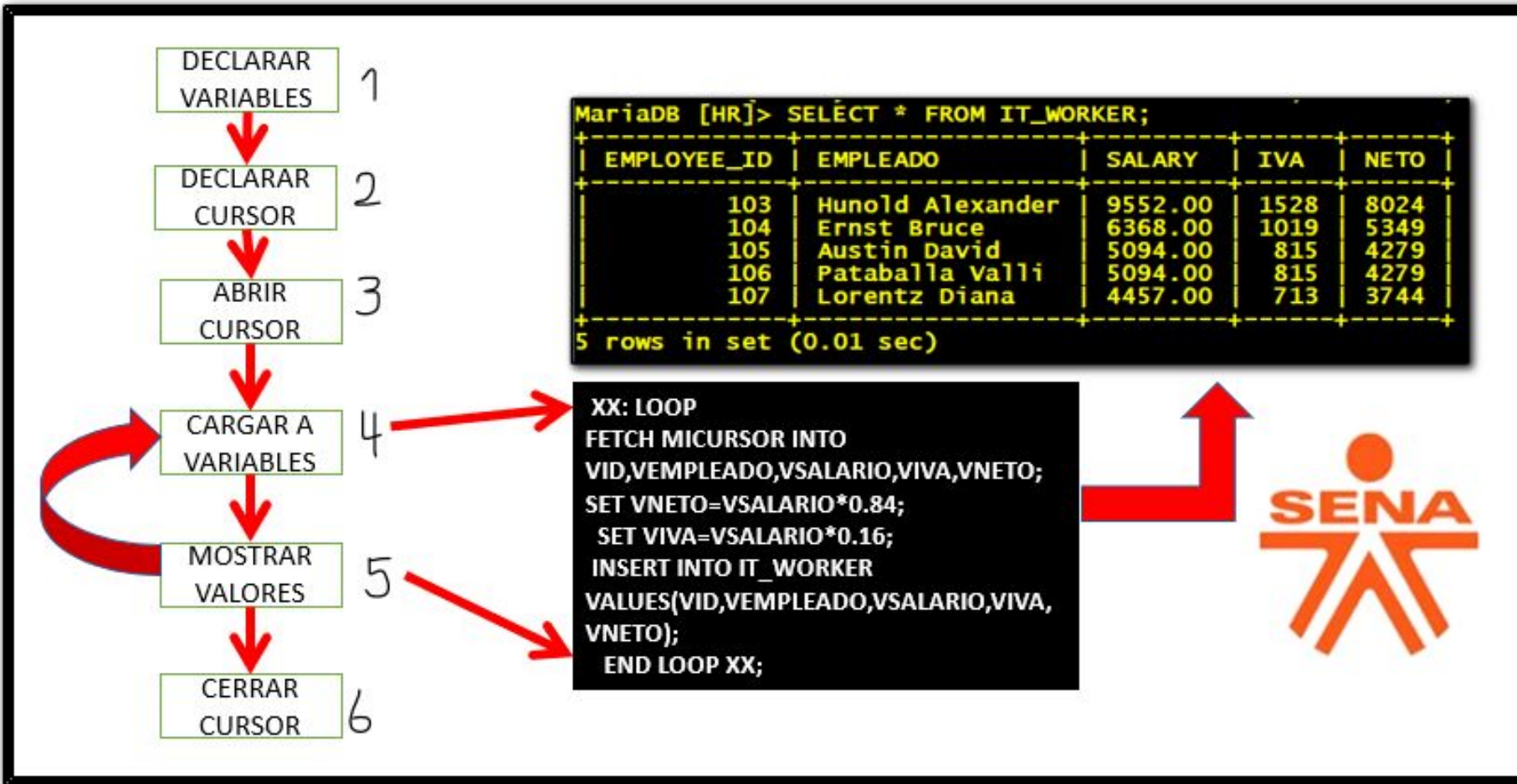
# CURSORES



# CURSORES

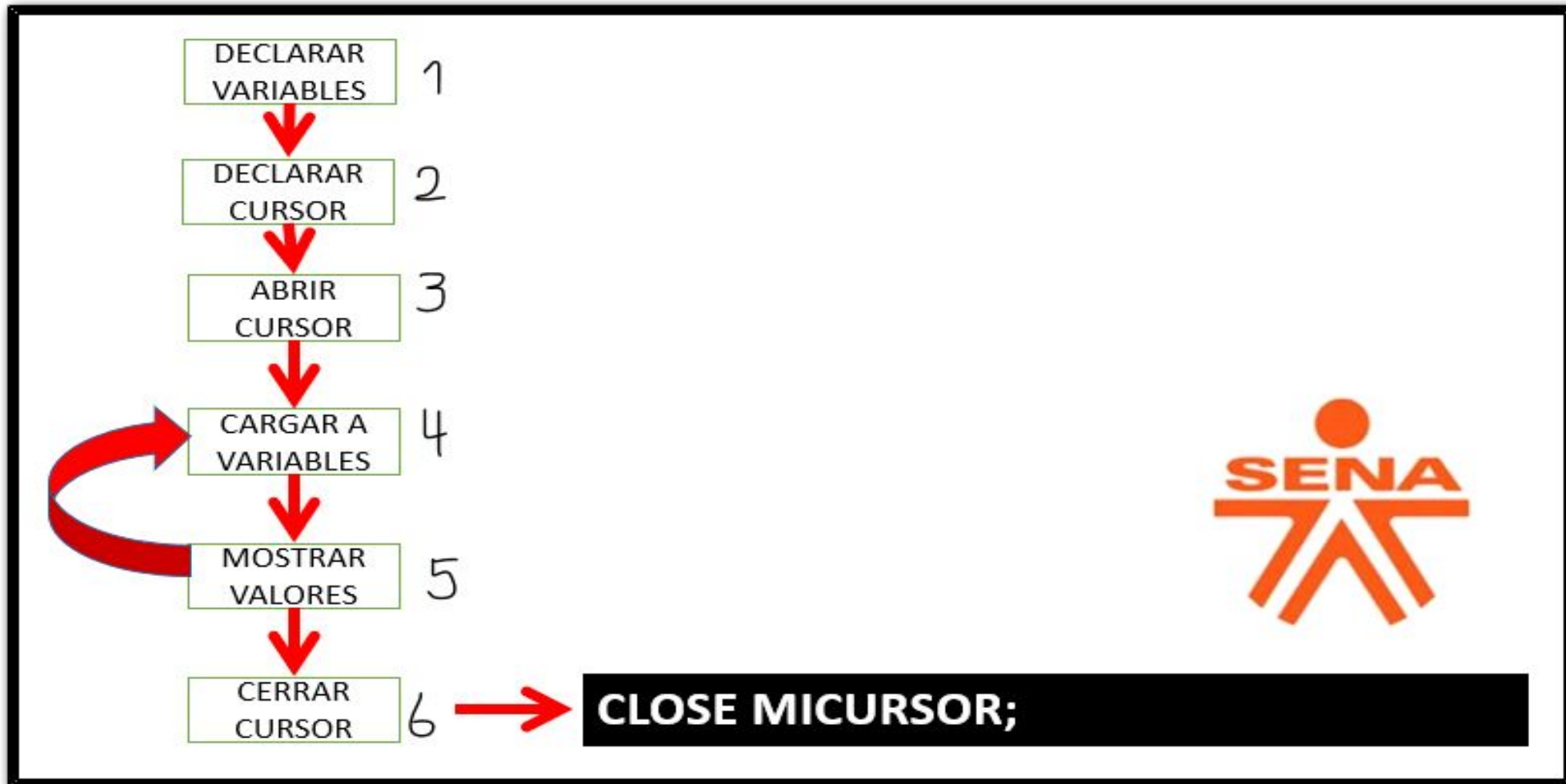


# CURSORES





# CURSORES



# CURSORES



```
DELIMITER $
CREATE OR REPLACE PROCEDURE PRDEMOCURSOR()
BEGIN
    DECLARE VID INTEGER;
    DECLARE VEMPLEADO VARCHAR(50);
    DECLARE VSALARIO INTEGER;
    DECLARE VIVA INTEGER;
    DECLARE VNETO INTEGER;

    DECLARE MICURSOR CURSOR FOR
    SELECT EMPLOYEE_ID, CONCAT(LAST_NAME, ' ', FIRST_NAME) EMPLEADO,
    SALARY, 0 IVA, 0 NETO FROM EMPLOYEES WHERE JOB_ID='IT_PROG';

    OPEN MICURSOR;
    XX: LOOP
        FETCH MICURSOR INTO VID, VEMPLEADO, VSALARIO, VIVA, VNETO;
        SET VNETO=VSALARIO*0.84;
        SET VIVA=VSALARIO*0.16;
        INSERT INTO IT_WORKER VALUES(VID, VEMPLEADO, VSALARIO, VIVA, VNETO);
    END LOOP XX;
    CLOSE MICURSOR;
END;
$
DELIMITER ;
```



```
MariaDB [HR]> DELIMITER $
MariaDB [HR]> CREATE OR REPLACE PROCEDURE PRDEMOCURSOR()
--> BEGIN
--> DECLARE VID INTEGER;
--> DECLARE VEMPLEADO VARCHAR(50);
--> DECLARE VSALARIO INTEGER;
--> DECLARE VIVA INTEGER;
--> DECLARE VNETO INTEGER;
-->
--> DECLARE MICURSOR CURSOR FOR
--> SELECT EMPLOYEE_ID, CONCAT(LAST_NAME, ' ', FIRST_NAME) EMPLEADO,
--> SALARY, 0 IVA, 0 NETO FROM EMPLOYEES WHERE JOB_ID='IT_PROG';
-->
--> OPEN MICURSOR;
--> XX: LOOP
-->     FETCH MICURSOR INTO VID, VEMPLEADO, VSALARIO, VIVA, VNETO;
-->     SET VNETO=VSALARIO*0.84;
-->     SET VIVA=VSALARIO*0.16;
-->     INSERT INTO IT_WORKER VALUES(VID, VEMPLEADO, VSALARIO, VIVA, VNETO);
--> END LOOP XX;
--> CLOSE MICURSOR;
--> END;
--> $
Query OK, 0 rows affected (0.01 sec)

MariaDB [HR]> DELIMITER ;
MariaDB [HR]> CALL PRDEMOCURSOR();
ERROR 1329 (02000): No data - zero rows fetched, selected, or processed
MariaDB [HR]> SELECT * FROM IT_WORKER;
```

EMPLOYEE_ID	EMPLEADO	SALARY	IVA	NETO
103	Hunold Alexander	9695.00	1551	8144
104	Ernst Bruce	6464.00	1034	5430
105	Austin David	5170.00	827	4343
106	Pataballa Valli	5170.00	827	4343
107	Lorentz Diana	4524.00	724	3800

```
5 rows in set (0.00 sec)
```



# TE RETO



# PRÁCTICA TRABAJO AUTÓNOMO



CARGO	INCREMENTO
IT_PROG	1.5%
HR_REP	1.6%
SA_REP	1.7%
ST_CLERK	1.5%
SH_CLERK	1.5%
AC_ACCOUNT	1.7%
FI_ACCOUNT	1.6%
El resto de <u>cargos</u>	1.2%

## RETO POR REALIZAR

Realizar un procedimiento llamado MODIFICARSALARIO que utilice un cursor con el id, el salario y el cargo que ocupa. De acuerdo al cargo se hará un incremento así:

Cuando se actualice el salario en la tabla EMPLOYEES se debe construir un trigger llamado TRGSALARIO donde se guardará el id, el salario antiguo, el nuevo salario, el cargo, la fecha y el usuario que realizó la transacción en la tabla LOGSALARIO.



# MATERIALES DE FORMACIÓN



**Centro de Gestión de Mercados,  
Logística y Tecnologías  
De la Información**  
**Regional Distrito Capital**



**DATASET NEPTUNO**

**EXCEL NEPTUNO**



# DUDAS Y PREGUNTAS



# CUENTAME CÓMO TE FUE?



JOSE FERNANDO GALINDO SUAREZ



# CREDITOS

Por: José Fernando Galindo Suarez  
jgalindos@sena.edu.co







# GRACIAS

Línea de atención al ciudadano: 01 8000 910270  
Línea de atención al empresario: 01 8000 910682



[www.sena.edu.co](http://www.sena.edu.co)