



# UNIVERSIDAD SAN BUENAVENTURA CALI

## GESTION DE LA CONFIGURACION

Versión 1.0

### RETO BASES DE DATOS

Autor(es):

Manuel A. González González

### Control de versiones

Versión	Fecha	Autor	Revisor	Descripción del cambio
1.0	23/05/2020	Manuel A. González G	Harold Adrián Bolaños	Creación Documento



# UNIVERSIDAD SAN BUENAVENTURA CALI

Gestión de la Configuración

Versión 1.0

---

## Tabla de Contenido

Introducción .....	3
Desarrollo del reto .....	3
Creación de una nueva huella.....	4
Validar una huella con la actual del sistema.....	6
Conclusiones .....	10



# UNIVERSIDAD SAN BUENAVENTURA CALI

## Gestión de la Configuración

Versión 1.0

---

### Introducción

En este documento se encuentra la solución de un reto en la asignatura gestión de la configuración que tiene que ver con el proceso de huellas y cifrados en bases de datos MYSQL

Se realizó una aplicación móvil híbrida realizada en IONIC para la representación de la gestión de huellas de una base de datos MYSQL.

### Desarrollo del reto

Como lo dije anteriormente para el desarrollo de este reto se realizó una aplicación móvil que me permitirá llevar la gestión en cuanto a ítems de configuración de una base de datos. Para ello se utilizaron las siguientes tecnologías:

FRONTEND: IONIC framework 5

BACKEND: NODE JS

BASE DE DATOS: MYSQL

### Creación de una nueva huella.

Esta función se realiza con un procedimiento almacenado que me recibe el nombre de la el nombre de la huella y la versión y este procedimiento realiza todo el proceso y luego desde el front solo pedirle a través de un botón sus datos.

Nueva huella

Nombre

Versión

REGISTRAR HUELLA

Huellas

Nueva huella

```
drop procedure generatedPawPrintF;
DELIMITER //
CREATE PROCEDURE generatedPawPrintF(IN name VARCHAR(50),in version varchar(5
))
BEGIN
declare t_name varchar(64);
declare p_name varchar(64);
declare np varchar(30);
declare id int;
declare v varchar(5);
    declare done boolean DEFAULT false;
DECLARE tablas_vistas CURSOR FOR
    SELECT table_name FROM information_schema.COLUMNS WHERE table_schema =
'pawprints' GROUP BY table_name;
```

```
DECLARE procedimientos CURSOR FOR
    SELECT routines.routine_name FROM information_schema.routines LEFT JOIN
    information_schema.parameters ON routines.specific_name=parameters.specific_name
    WHERE parameters.specific_schema='pawprints' group by routines.routine_name
    ORDER BY routines.routine_name;

    DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET done = true;

    select namepawprint, version into np, v from pawprint where namepawprint=name and version=version;
    IF(np is null) THEN

        insert into pawprint values(null,name,version,now());
        set id=LAST_INSERT_ID() ;

        SET done = false;
        OPEN tablas_vistas;
        get_record: LOOP
            FETCH tablas_vistas INTO t_name;
            IF done = true THEN
                LEAVE get_record;
            END IF;
            INSERT INTO configurationitem VALUES(null,'Tabla/Vista',t_name,sha256tableF(t_name), id);
        END LOOP get_record;
        SET done = false;
        CLOSE tablas_vistas;

        OPEN procedimientos;
        get_record: LOOP
            FETCH procedimientos INTO p_name;
            IF done = true THEN
                LEAVE get_record;
            END IF;
            INSERT INTO configurationitem VALUES(null,'Procedure/Function',p_name,sha256tableF(p_name),id);
        END LOOP get_record;
        SET done = false;
```

```
        CLOSE procedimientos;  
  
    END IF;  
  
END  
//  
DELIMITER ;  
Call generatedPawPrintF("V5",2);
```

Este procedimiento almacenado guarda la información en dos tablas relacionadas en la base de datos que se llaman pawprint y configurationitem, en donde en la tabla pawprint se almacenan el nombre de la huella, la versión y la fecha en que se generó la huella. Y en la tabla configurationitem se almacena todos los ítem de configuración correspondientes a el id de esa huella como el tipo que me dice si es una tabla, una vista, un procedimiento o una función, el ítem que me determina el nombre de la tabla, vista o procedimiento y la contraseña encriptada a sha265.

### Validar una huella con la actual del sistema.


Esta función se realiza con un procedimiento almacenado que me recibe como parámetro el id de la huella a comparar, este me toma una foto de la huella actual del sistema y la compara con la huella inicial registrada y me identifica si hubo o no hubo cambios en los ítems de configuración de la huella inicial a comparar.

## UNIVERSIDAD SAN BUENAVENTURA CALI

Gestión de la Configuración


Versión 1.0

### Huellas de base de datos

 HUELLA 1


SELECCIONAR

Versión: 1.0 - Fecha: 5/23/20, 12:00 AM

 HUELLA 2

SELECCIONAR


Versión: 1.2 - Fecha: 5/23/20, 12:00 AM

 HUELLA 3

SELECCIONAR

Versión: 1.4 - Fecha: 5/23/20, 12:00 AM

Opciones

 Comparar huellas

✕ Cancel

```
drop procedure validatepawPrint;
delimiter //
create procedure validatepawPrint(in namReal int)
Begin

    declare t varchar(50);
    declare n varchar(30);
    declare p varchar(500);
    declare p2 varchar(500);
```

```
declare idp int;
declare id varchar(30);
declare done boolean DEFAULT false;

DECLARE result CURSOR FOR select type,item,password, case type WHEN 'Ta
bla/Vista' THEN sha256tableF(item) ELSE sha256tableF(item) END AS Password2,
idpaw from configurationitem c where idpaw= namReal;

DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET done = true;
if not exists(select idpaw from pawPrint_temp where idpaw=namreal) then
    OPEN result;
    get_record: LOOP
        FETCH result INTO t,n,p,p2,idp;
        IF done = true THEN
            LEAVE get_record;
        END IF;
        INSERT INTO pawPrint_temp VALUES(null,t,n,p,p2,idp);
    END LOOP get_record;
CLOSE result;
ELSE
    OPEN result;
    get_record: LOOP
        FETCH result INTO t,n,p,p2,idp;
        IF done = true THEN
            LEAVE get_record;
        END IF;
        UPDATE pawPrint_temp SET password2=p2 WHERE idpaw=idp
and item=n;
    END LOOP get_record;
CLOSE result;
end if;
select type, item, password, password2 from pawprint_temp where
idpaw=namreal ;
End
//
delimiter ;
call validatepawPrint("V2");
```



### < Huella 1

#### CATEGORY

Fecha: 5/23/20, 12:00 AM

Tipo: Tabla/Vista - Version: 1.0

#### CONFIGURATIONITEM

Fecha: 5/23/20, 12:00 AM

Tipo: Tabla/Vista - Version: 1.0

#### PAWPRINT

Fecha: 5/23/20, 12:00 AM

Tipo: Tabla/Vista - Version: 1.0

#### PAWPRINT\_TEMP

Fecha: 5/23/20, 12:00 AM

Tipo: Tabla/Vista - Version: 1.0

#### SIG\_MATERIALS

Fecha: 5/23/20, 12:00 AM

Tipo: Tabla/Vista - Version: 1.0

#### SIG\_TEMP

Fecha: 5/23/20, 12:00 AM

Tipo: Tabla/Vista - Version: 1.0

#### SIG\_TOOL

Fecha: 5/23/20, 12:00 AM

Tipo: Tabla/Vista - Version: 1.0

#### SIG\_WORKFORCES

Fecha: 5/23/20, 12:00 AM

Tipo: Tabla/Vista - Version: 1.0

GENERATED BY DDINIT

En la imagen anterior se muestra todo el detalle de la huella con los ítems de configuración evaluados



# UNIVERSIDAD SAN BUENAVENTURA CALI

Gestión de la Configuración

Versión 1.0

---

## Conclusiones

Con la realización de este reto logramos aprender sobre la gestión de huellas en bases de datos, esto es de vital importancia ya que nos permite identificar si en algún momento nuestra base de datos ha sido modificada y poder de alguna forma tener la versión inicial para poder hacer el cambio correspondiente.