

**MANUAL TÉCNICO**

**Gestión de Calidad**

**Historial de Cambios**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Versión | Item | Autor | Descripción | Fecha |
| v 0.1 | Implementación | Jhon Jaimes Cayetano |  | 07/11/2024 |

**Índice**

Tabla de contenido

[Procesos 4](#_Toc181693148)

[Importación de datos 4](#_Toc181693149)

[Generación de Reportes 5](#_Toc181693150)

[Modelo de Base de Datos 6](#_Toc181693151)

[Script de Base de Datos 6](#_Toc181693152)

Descripción del Sistema

El sistema de gestión de calidad está diseñado para supervisar y evaluar los indicadores de calidad asociados a los procesos de la institución. Cada indicador refleja los resultados obtenidos a lo largo de los años, proporcionando una visión clara del rendimiento y la efectividad de los procesos. Además, el sistema permite generar reportes detallados que presentan estos resultados a lo largo del tiempo, facilitando la identificación de tendencias y áreas de mejora en la gestión de calidad.

A screenshot of a computer

Description automatically generated

# Procesos

## Importación de datos

En el proceso de importación de datos, se puede descargar una plantilla de Excel que contiene datos actualizados hasta la fecha de la descarga. Posteriormente, el archivo se rellena de forma manual con la información necesaria y se carga en la plataforma para su procesamiento. Una vez procesada, la información se inserta en la base de datos del sistema.

A screenshot of a computer screen

Description automatically generated

## Generación de Reportes

En el proceso de generación de reportes, primero se selecciona un indicador específico. A continuación, se indica un rango de años correspondiente. Una vez realizados estos pasos, se acepta la selección, lo que permite al sistema obtener los resultados históricos almacenados para la generación del reporte.

A screenshot of a computer screen

Description automatically generated

# Modelo de Base de Datos

A black screen with white text

Description automatically generated

# Script de Base de Datos

-- Campo LEGACYEFQM

INSERT INTO CONTROL (CAMPO, ANO, NUMERO, DETALLE, VISIBLE) VALUES ('LEGACYEFQM', '0000', '0', '1=antiguo, 0=nuevo', 1);

-- Generador

CREATE GENERATOR GEN\_CC\_AGRUPADOR\_ID;

-- Tabla CC\_ANO

CREATE TABLE CC\_ANO (  
    ANO     ANO NOT NULL /\* ANO = VARCHAR(5) \*/,  
    INICIO  ANO /\* ANO = VARCHAR(5) \*/,  
    FIN     ANO /\* ANO = VARCHAR(5) \*/  
);

ALTER TABLE CC\_ANO ADD CONSTRAINT PK\_CC\_ANO PRIMARY KEY (ANO);

insert into CC\_ANO (ANO, INICIO, FIN)  
select ANO, cast(cast(ANO as integer) - 4 as ANO), ANO  
from ANO;

-- Tabla para procesos adicionales

CREATE TABLE CC\_PROCESOS\_PADRE (

    IDPROCESO        INTEGER NOT NULL,

    IDPROCESO\_PADRE  INTEGER NOT NULL

);

-- Tabla para agrupadores

CREATE TABLE CC\_AGRUPADOR (

    IDAGRUPADOR        INTEGER NOT NULL,

    DESCRIPCION        VARCHAR(200) CHARACTER SET ISO8859\_1,

    IDAGRUPADOR\_PADRE  INTEGER

);

-- Procedimiento SP\_LISTAR\_SERIES\_POR\_INDICADOR

create or alter procedure SP\_LISTAR\_SERIES\_POR\_INDICADOR (

ID\_INDICADOR type of column CC\_ENCUESTA\_INDICADOR.IDINDICADOR)

returns (

IDENCUESTA\_IND type of column CC\_ENCUESTA\_INDICADOR.IDENCUESTA\_IND,

ANO type of column QUIZ\_ASIGNA.ANO,

IDQUIZASIGNA type of column QUIZ\_ASIGNA.IDQUIZASIGNA,

IDQUIZ type of column QUIZ\_ASIGNA.IDQUIZ,

IDPREGUNTA type of column QUIZ\_PREGUNTA.IDPREGUNTA,

ORDEN type of column QUIZ\_PREGUNTA.ORDEN,

PREGUNTA type of column QUIZ\_PREGUNTA.PREGUNTA,

IDOPCRPTA type of column QUIZ\_PREGUNTA\_OPC.IDOPCRPTA,

ORDENOPC type of column QUIZ\_PREGUNTA\_OPC.ORDEN,

OPCION type of column QUIZ\_PREGUNTA\_OPC.OPCION,

ORIGEN varchar(1))

as

begin

for

select \*

from (select

EI.IDENCUESTA\_IND,

QA.ANO,

QA.IDQUIZASIGNA,

QA.IDQUIZ,

QP.IDPREGUNTA,

QP.ORDEN,

QP.PREGUNTA,

QPO.IDOPCRPTA,

QPO.ORDEN as ORDENOPC,

QPO.OPCION,

cast('Z' as varchar(1)) as ORIGEN

from QUIZ\_ASIGNA QA

inner join QUIZ\_PREGUNTA QP on QP.IDQUIZ = QA.IDQUIZ

inner join QUIZ\_TIPO\_PREGUNTA QTP on QTP.IDTIPOPREGUNTA = QP.IDTIPOPREGUNTA

inner join QUIZ\_PREGUNTA\_OPC QPO on QPO.IDQUIZ = QP.IDQUIZ and QPO.IDPREGUNTA = QP.IDPREGUNTA

inner join CC\_ENCUESTA\_INDICADOR EI on EI.IDQUIZASIGNA = QA.IDQUIZASIGNA and EI.IDPREGUNTA = QP.IDPREGUNTA and (EI.IDINDICADOR = :id\_indicador) and EI.ORIGENENCUESTA = 'Z'

union all

select EI.IDENCUESTA\_IND, QA.ANO, QA.IDQUIZASIGNA, QA.IDQUIZ, QP.IDPREGUNTA, QP.ORDEN, QP.PREGUNTA, QPO.IDOPCRPTA,

QPO.ORDEN as ORDENOPC, QPO.OPCION, cast('Z' as varchar(1)) as ORIGEN

from QUIZ\_ASIGNA QA

inner join QUIZ\_PREGUNTA QP on QP.IDQUIZ = QA.IDQUIZ

inner join QUIZ\_TIPO\_PREGUNTA QTP on QTP.IDTIPOPREGUNTA = QP.IDTIPOPREGUNTA

inner join QUIZ\_TIPO\_PREGUNTA\_OPC QPO on QPO.IDTIPOPREGUNTA = QTP.IDTIPOPREGUNTA

inner join CC\_ENCUESTA\_INDICADOR EI on EI.IDQUIZASIGNA = QA.IDQUIZASIGNA and EI.IDPREGUNTA = QP.IDPREGUNTA and (EI.IDINDICADOR = :id\_indicador) and EI.ORIGENENCUESTA = 'Z'

union all

select EI.IDENCUESTA\_IND, extract(year from HF.FH\_INICIO) as ANO, HF.ID\_HISTORIA\_FORMULARIO, HF.ID\_FORMULARIO,

PG.ID\_PREGUNTA, PG.ORDEN,

cast(substring(coalesce(PG.PREGUNTA, '') from 1 for 490) as varchar(500)) as PREGUNTA,

O.ID\_OPCION as IDOPCRPTA, 1 as ORDENOPC,

cast(substring(coalesce(O.VALOR, '') from 1 for 490) as varchar(500)) as OPCION,

cast('N' as varchar(1)) as ORIGEN

from FRM\_HISTORIA\_FORMULARIO HF

inner join FRM\_FORMULARIO F on F.ID\_FORMULARIO = HF.ID\_FORMULARIO

inner join FRM\_PREGUNTA PG on PG.ID\_FORMULARIO = HF.ID\_FORMULARIO

inner join FRM\_OPCION\_PREGUNTA OP on OP.ID\_PREGUNTA = PG.ID\_PREGUNTA

inner join FRM\_OPCION O on O.ID\_OPCION = OP.ID\_OPCION and O.PESO != - 999

inner join CC\_ENCUESTA\_INDICADOR EI on EI.IDQUIZASIGNA = HF.ID\_HISTORIA\_FORMULARIO and EI.IDPREGUNTA = PG.ID\_PREGUNTA and EI.ORIGENENCUESTA = 'N' and (EI.IDINDICADOR = :id\_indicador)) as T1

order by T1.ANO, T1.IDQUIZ, T1.IDPREGUNTA, T1.IDOPCRPTA

into :IDENCUESTA\_IND, :ANO, :IDQUIZASIGNA, :IDQUIZ, :IDPREGUNTA, :ORDEN, :PREGUNTA, :IDOPCRPTA, :ORDENOPC, :OPCION, :ORIGEN

do

begin

suspend;

end

end;

-- Tabla de Indicadores de Cumplimiento

CREATE TABLE CC\_INDICADOR\_CUMPLIMIENTO (

    IDINDICADOR        INTEGER NOT NULL,

    TIPO\_CUMPLIMIENTO  INTEGER NOT NULL,

    DESCRIPCION        VARCHAR(100) CHARACTER SET ISO8859\_1,

    VALORMINIMO        DECIMAL(10,2),

    VALORMAXIMO        DECIMAL(10,2)

);

-- Tabla de Tendencias

CREATE TABLE CC\_INDICADOR\_TENDENCIA (

    ID\_TENDENCIA  INTEGER NOT NULL,

    DESCRIPCION   VARCHAR(100) CHARACTER SET ISO8859\_1

);

-- Edición de campos

ALTER TABLE cc\_reporte\_manual\_detalle ADD NUMERADOR NUMERIC(8,2);

ALTER TABLE cc\_reporte\_manual\_detalle ADD DENOMINADOR NUMERIC(8,2);

ALTER TABLE cc\_reporte\_manual\_detalle ADD IDAGRUPADOR INTEGER;

ALTER TABLE cc\_reporte\_manual\_detalle ADD IDDETALLE BIGINT;

ALTER TABLE cc\_indicador ADD META\_GLOBAL INTEGER;

ALTER TABLE cc\_indicador ADD FORMULA VARCHAR(500) CHARACTER SET ISO8859\_1;

ALTER TABLE cc\_indicador ADD TENDENCIA INTEGER DEFAULT 0 NOT NULL;

ALTER TABLE cc\_indicador ADD FRECUENCIA VARCHAR(500) CHARACTER SET ISO8859\_1;

ALTER TABLE cc\_indicador ADD AGRUPAMIENTO SMALLINT;

ALTER TABLE CC\_PROCESOS\_PADRE ADD CONSTRAINT PK\_CC\_PROCESOS\_PADRE PRIMARY KEY (IDPROCESO, IDPROCESO\_PADRE);

ALTER TABLE CC\_PROCESOS\_PADRE ADD CONSTRAINT FK\_CC\_PROCESO\_HIJO FOREIGN KEY (IDPROCESO) REFERENCES CC\_PROCESO (IDPROCESO) ON DELETE CASCADE;

ALTER TABLE CC\_PROCESOS\_PADRE ADD CONSTRAINT FK\_CC\_PROCESO\_PADRE FOREIGN KEY (IDPROCESO\_PADRE) REFERENCES CC\_PROCESO (IDPROCESO);

ALTER TABLE CC\_AGRUPADOR ADD CONSTRAINT PK\_CC\_AGRUPADOR PRIMARY KEY (IDAGRUPADOR);

ALTER TABLE CC\_AGRUPADOR ADD CONSTRAINT FK\_CC\_AGRUPADOR\_1 FOREIGN KEY (IDAGRUPADOR\_PADRE) REFERENCES CC\_AGRUPADOR (IDAGRUPADOR);

ALTER TABLE CC\_INDICADOR\_CUMPLIMIENTO ADD CONSTRAINT PK\_CC\_INDICADOR\_CUMPLIMIENTO PRIMARY KEY (IDINDICADOR, TIPO\_CUMPLIMIENTO);

ALTER TABLE CC\_INDICADOR\_CUMPLIMIENTO ADD CONSTRAINT FK\_CC\_INDICADOR\_CUMPLIMIENTO\_1 FOREIGN KEY (IDINDICADOR) REFERENCES CC\_INDICADOR (IDINDICADOR);

ALTER TABLE CC\_INDICADOR\_TENDENCIA ADD CONSTRAINT PK\_CC\_INDICADOR\_TENDENCIA PRIMARY KEY (ID\_TENDENCIA);

-- Procedimiento para la migración de datos  
CREATE OR ALTER PROCEDURE INSERT\_CC\_PROCESOS\_PADRE  
AS  
  DECLARE VARIABLE IDPROCESO INTEGER;  
  DECLARE VARIABLE IDPROCESO\_PADRE INTEGER;  
BEGIN  
  FOR SELECT IDPROCESO, IDPROCESO\_PADRE  
      FROM cc\_proceso  
      WHERE IDPROCESO IS NOT NULL AND IDPROCESO\_PADRE IS NOT NULL  
      INTO :IDPROCESO, :IDPROCESO\_PADRE  
  DO  
  BEGIN  
    INSERT INTO CC\_PROCESOS\_PADRE (IDPROCESO, IDPROCESO\_PADRE)  
    VALUES (:IDPROCESO, :IDPROCESO\_PADRE);  
  END  
END

-- Trigger para actualizaciones en CC\_PROCESO

CREATE OR ALTER TRIGGER CC\_PROCESO\_AU0 FOR CC\_PROCESO

ACTIVE AFTER UPDATE POSITION 0

as

begin

  if (new.IDPROCESO\_PADRE <> old.IDPROCESO\_PADRE) then

    update CC\_PROCESOS\_PADRE

    set IDPROCESO\_PADRE = new.IDPROCESO\_PADRE

    where new.IDPROCESO = IDPROCESO;

end;

-- Trigger para inserciones en CC\_PROCESO

CREATE OR ALTER TRIGGER CC\_PROCESO\_AI0 FOR CC\_PROCESO

ACTIVE AFTER INSERT POSITION 0

as

begin

  insert into CC\_PROCESOS\_PADRE (IDPROCESO, IDPROCESO\_PADRE)

  values (new.IDPROCESO, new.IDPROCESO\_PADRE);

end;

-- Trigger antes de las inserciones en CC\_AGRUPADOR

CREATE OR ALTER TRIGGER CC\_AGRUPADOR\_BI FOR CC\_AGRUPADOR

ACTIVE BEFORE INSERT POSITION 0

as

begin

  if (new.idagrupador is null) then

    new.idagrupador = gen\_id(gen\_cc\_agrupador\_id,1);

end;

-- Creación de vista V\_CC\_PROCESO\_RUTA\_PADRE

CREATE OR ALTER VIEW V\_CC\_PROCESO\_RUTA\_PADRE(

    ORDERBY,

    RUTA\_PROCESO,

    RUTA\_PROCESO\_ABR,

    IDPROCESO,

    TITULO,

    NIVEL\_PROCESO,

    ORDEN,

    IDPROCESO\_PADRE,

    ABREVIATURA)

AS

select T.\*

from (select lower(((

             case

               when CP4.IDPROCESO is null then ''

               else (CP4.NIVEL\_PROCESO || '.' || coalesce(CP4.ORDEN, 0) || '.' || coalesce(CP4.TITULO, '') || '.' || '\_')

             end) || '' ||(

             case

               when CP3.IDPROCESO is null then ''

               else (CP3.NIVEL\_PROCESO || '.' || coalesce(CP3.ORDEN, 0) || '.' || coalesce(CP3.TITULO, '') || '.' || '\_')

             end) || '' ||(

             case

               when CP2.IDPROCESO is null then ''

               else (CP2.NIVEL\_PROCESO || '.' || coalesce(CP2.ORDEN, 0) || '.' || coalesce(CP2.TITULO, '') || '.' || '\_')

             end) || '' ||(

             case

               when CP1.IDPROCESO is null then ''

               else (CP1.NIVEL\_PROCESO || '.' || coalesce(CP1.ORDEN, 0) || '.' || coalesce(CP1.TITULO, '') || '.' || '\_')

             end))) as ORDERBY,

             ((

             case

               when CP4.IDPROCESO is null then ''

               else ((coalesce(CP4.TITULO, '')) || '' || ' / ')

             end) || '' ||(

             case

               when CP3.IDPROCESO is null then ''

               else ((coalesce(CP3.TITULO, '')) || '' || ' / ')

             end) || '' ||(

             case

               when CP2.IDPROCESO is null then ''

               else ((coalesce(CP2.TITULO, '')) || '' || ' / ')

             end) || '' ||(

             case

               when CP1.IDPROCESO is null then ''

               else ((coalesce(CP1.TITULO, '')) || '' || '')

             end)) as RUTA\_PROCESO,

             ((

             case

               when CP4.IDPROCESO is null then ''

               else ((coalesce(CP4.ABREVIATURA, '')) || '' || ' / ')

             end) || '' ||(

             case

               when CP3.IDPROCESO is null then ''

               else ((coalesce(CP3.ABREVIATURA, '')) || '' || ' / ')

             end) || '' ||(

             case

               when CP2.IDPROCESO is null then ''

               else ((coalesce(CP2.ABREVIATURA, '')) || '' || ' / ')

             end) || '' ||(

             case

               when CP1.IDPROCESO is null then ''

               else ((coalesce(CP1.ABREVIATURA, '')) || '' || '')

             end)) as RUTA\_PROCESO\_ABR,

             CP1.\*

      from CC\_PROCESO CP1

      left join CC\_PROCESO CP2 on CP2.IDPROCESO = CP1.IDPROCESO\_PADRE

      left join CC\_PROCESO CP3 on CP3.IDPROCESO = CP2.IDPROCESO\_PADRE

      left join CC\_PROCESO CP4 on CP4.IDPROCESO = CP3.IDPROCESO\_PADRE) T

order by T.ORDERBY asc;

-- Creación de procedimiento SP\_LISTAR\_INDICADORES

create or alter procedure SP\_LISTAR\_INDICADORES (

COLCOD type of column COLEGIO.COLCOD)

returns (

IDINDICADOR type of column CC\_INDICADOR.IDINDICADOR,

IDPROCESO type of column CC\_PROCESO.IDPROCESO,

IDTIPO type of column CC\_INDICADOR.IDTIPO,

IDCAMPO type of column CC\_INDICADOR.IDCAMPO,

IDCRITERIO type of column CC\_INDICADOR.IDCRITERIO,

TITULO type of column CC\_INDICADOR.TITULO,

ORDEN type of column CC\_INDICADOR.ORDEN,

TIPO\_REPORTE type of column CC\_INDICADOR.TIPO\_REPORTE,

GRAFICO\_DEFECTO type of column CC\_INDICADOR.GRAFICO\_DEFECTO,

MOSTRAR\_VALGRAF type of column CC\_INDICADOR.MOSTRAR\_VALGRAF,

ASIGNACIONES varchar(500) character set ISO8859\_1,

RUTA\_PROCESO\_ABR varchar(100) character set ISO8859\_1,

META\_GLOBAL type of column CC\_INDICADOR.META\_GLOBAL,

FORMULA type of column CC\_INDICADOR.FORMULA,

TENDENCIA type of column CC\_INDICADOR.TENDENCIA,

FRECUENCIA type of column CC\_INDICADOR.FRECUENCIA,

AGRUPAMIENTO type of column CC\_INDICADOR.AGRUPAMIENTO,

TIPO\_CUMPLIMIENTO type of column CC\_INDICADOR.TIPO\_CUMPLIMIENTO)

as

begin

for select CI.IDINDICADOR, CI.IDPROCESO, CI.IDTIPO, CI.IDCAMPO, CI.IDCRITERIO, CI.TITULO, CI.ORDEN, CI.TIPO\_REPORTE,

CI.GRAFICO\_DEFECTO, CI.MOSTRAR\_VALGRAF, CI.META\_GLOBAL, CI.FORMULA, CI.TENDENCIA, CI.FRECUENCIA,

CI.AGRUPAMIENTO, coalesce((

case CI.TIPO\_REPORTE

when 'A' then (select list(ANOS)

from (select distinct ANIO as ANOS

from CC\_INDICADOR\_ADJUNTO

where IDINDICADOR = CI.IDINDICADOR

order by ANIO desc) as T)

when 'M' then (select list(ANOS)

from (select distinct ANIO as ANOS

from CC\_REPORTE\_MANUAL RM

inner join CC\_REPORTE\_MANUAL\_DETALLE RMD on RMD.IDREPORTE = RM.IDREPORTE

where RM.IDINDICADOR = CI.IDINDICADOR

order by ANIO desc) as T)

when 'E' then (select list(ANOS)

from (select distinct QA.ANO as ANOS

from CC\_ENCUESTA\_INDICADOR EI

inner join QUIZ\_ASIGNA QA on EI.IDQUIZASIGNA = QA.IDQUIZASIGNA

inner join QUIZ Q on Q.IDQUIZ = QA.IDQUIZ

inner join QUIZ\_PREGUNTA QP on QP.IDQUIZ = Q.IDQUIZ and EI.IDPREGUNTA = QP.IDPREGUNTA

where EI.IDINDICADOR = CI.IDINDICADOR and

EI.ORIGENENCUESTA = 'Z'

union

select distinct extract(year from QA.FH\_INICIO) as ANOS

from CC\_ENCUESTA\_INDICADOR EI

inner join FRM\_HISTORIA\_FORMULARIO QA on EI.IDQUIZASIGNA = QA.ID\_HISTORIA\_FORMULARIO

inner join FRM\_FORMULARIO Q on Q.ID\_FORMULARIO = QA.ID\_FORMULARIO

inner join FRM\_PREGUNTA QP on QP.ID\_FORMULARIO = Q.ID\_FORMULARIO and EI.IDPREGUNTA = QP.ID\_PREGUNTA

where EI.IDINDICADOR = CI.IDINDICADOR and

EI.ORIGENENCUESTA = 'N'

order by 1 desc) as T)

when 'S' then (select list(CCR.REPCOD)

from CC\_REPORTES CCR

inner join CC\_GRUPO\_REP CCGR on CCR.GREPCOD = CCGR.GREPCOD

where CCR.COLCOD = :COLCOD and

CCR.ESTADO = 'A' and

CCR.IDINDICADOR = CI.IDINDICADOR)

else ''

end), '') as ASIGNACIONES, CP.RUTA\_PROCESO\_ABR ,CI.TIPO\_CUMPLIMIENTO

from CC\_INDICADOR CI

inner join(select lower(((

case

when CP4.IDPROCESO is null then ''

else (CP4.NIVEL\_PROCESO || '.' || coalesce(CP4.ORDEN, 0) || '.' || coalesce(CP4.TITULO, '') || '.' || '\_')

end) || '' ||(

case

when CP3.IDPROCESO is null then ''

else (CP3.NIVEL\_PROCESO || '.' || coalesce(CP3.ORDEN, 0) || '.' || coalesce(CP3.TITULO, '') || '.' || '\_')

end) || '' ||(

case

when CP2.IDPROCESO is null then ''

else (CP2.NIVEL\_PROCESO || '.' || coalesce(CP2.ORDEN, 0) || '.' || coalesce(CP2.TITULO, '') || '.' || '\_')

end) || '' ||(

case

when CP1.IDPROCESO is null then ''

else (CP1.NIVEL\_PROCESO || '.' || coalesce(CP1.ORDEN, 0) || '.' || coalesce(CP1.TITULO, '') || '.' || '\_')

end))) as ORDERBY,

((

case

when CP4.IDPROCESO is null then ''

else ((coalesce(CP4.TITULO, '')) || '' || ' / ')

end) || '' ||(

case

when CP3.IDPROCESO is null then ''

else ((coalesce(CP3.TITULO, '')) || '' || ' / ')

end) || '' ||(

case

when CP2.IDPROCESO is null then ''

else ((coalesce(CP2.TITULO, '')) || '' || ' / ')

end) || '' ||(

case

when CP1.IDPROCESO is null then ''

else ((coalesce(CP1.TITULO, '')) || '' || '')

end)) as RUTA\_PROCESO,

((

case

when CP4.IDPROCESO is null then ''

else ((coalesce(CP4.ABREVIATURA, '')) || '' || ' / ')

end) || '' ||(

case

when CP3.IDPROCESO is null then ''

else ((coalesce(CP3.ABREVIATURA, '')) || '' || ' / ')

end) || '' ||(

case

when CP2.IDPROCESO is null then ''

else ((coalesce(CP2.ABREVIATURA, '')) || '' || ' / ')

end) || '' ||(

case

when CP1.IDPROCESO is null then ''

else ((coalesce(CP1.ABREVIATURA, '')) || '' || '')

end)) as RUTA\_PROCESO\_ABR,

CP1.\*

from CC\_PROCESO CP1

left join CC\_PROCESO CP2 on CP2.IDPROCESO = CP1.IDPROCESO\_PADRE

left join CC\_PROCESO CP3 on CP3.IDPROCESO = CP2.IDPROCESO\_PADRE

left join CC\_PROCESO CP4 on CP4.IDPROCESO = CP3.IDPROCESO\_PADRE) CP on CP.IDPROCESO = CI.IDPROCESO

order by CP.ORDERBY asc, CI.ORDEN asc, CI.TITULO asc

into IDINDICADOR, IDPROCESO, IDTIPO, IDCAMPO, IDCRITERIO, TITULO, ORDEN, TIPO\_REPORTE, GRAFICO\_DEFECTO,

MOSTRAR\_VALGRAF, META\_GLOBAL, FORMULA, TENDENCIA, FRECUENCIA, AGRUPAMIENTO, ASIGNACIONES, RUTA\_PROCESO\_ABR , TIPO\_CUMPLIMIENTO

do

begin

suspend;

end

end;

-- Inserción de Tendencias

INSERT INTO CC\_INDICADOR\_TENDENCIA (ID\_TENDENCIA, DESCRIPCION) VALUES (0, 'AUMENTAR');

INSERT INTO CC\_INDICADOR\_TENDENCIA (ID\_TENDENCIA, DESCRIPCION) VALUES (1, 'DISMINUIR');

INSERT INTO CC\_INDICADOR\_TENDENCIA (ID\_TENDENCIA, DESCRIPCION) VALUES (2, 'MANTENER');

ALTER TABLE CC\_INDICADOR ADD TIPO\_CUMPLIMIENTO VARCHAR(1) CHARACTER SET ISO8859\_1 NOT NULL;

UPDATE CC\_INDICADOR SET TIPO\_CUMPLIMIENTO = 'P';

ALTER TABLE CC\_REPORTE\_MANUAL ADD IS\_CORFEP\_EFQM SMALLINT DEFAULT 0 ;

ALTER TABLE CC\_REPORTE\_MANUAL\_DETALLE ADD CORFEP\_EFQM NUMERIC(6,2);

COMMENT ON COLUMN CC\_REPORTE\_MANUAL.IS\_CORFEP\_EFQM IS '0 = No 1 = Si';

-- Vista de V\_AGRUPADORES\_CTE

CREATE OR ALTER VIEW V\_AGRUPADORES\_CTE(  
    IDAGRUPADOR,  
    DESCRIPCION,  
    IDAGRUPADOR\_PADRE,  
    NIVEL)  
AS  
with recursive AGRUPADORES\_CTE (IDAGRUPADOR, DESCRIPCION, IDAGRUPADOR\_PADRE, NIVEL)  
as (select IDAGRUPADOR, DESCRIPCION, IDAGRUPADOR\_PADRE, 0 as NIVEL  
    from CC\_AGRUPADOR CA  
    where IDAGRUPADOR\_PADRE is null -- Selecciona los padres    union all    select A.IDAGRUPADOR, A.DESCRIPCION, A.IDAGRUPADOR\_PADRE, CTE.NIVEL + 1  
    from CC\_AGRUPADOR A  
    join AGRUPADORES\_CTE CTE on A.IDAGRUPADOR\_PADRE = CTE.IDAGRUPADOR)select IDAGRUPADOR, DESCRIPCION, IDAGRUPADOR\_PADRE, NIVEL  
from AGRUPADORES\_CTE;

-- Secuencia GEN\_CC\_REPORTE\_MANUAL

CREATE SEQUENCE GEN\_CC\_REPORTE\_MANUAL\_ID;

-- Trigger CC\_REPORTE\_MANUAL\_BI  
create trigger cc\_reporte\_manual\_bi for cc\_reporte\_manual  
active before insert position 0  
as  
begin  
  if (new.idreporte is null) then  
    new.idreporte = gen\_id(gen\_cc\_reporte\_manual\_id,1);  
end;select gen\_id(gen\_cc\_reporte\_manual\_id,(select max(idreporte) from cc\_reporte\_manual)) from colegio;

-- Inserción o actualización del Menú

UPDATE OR INSERT INTO MENUU (SISCOD, MENUCOD, COLCOD, MENUNOM, MENUPAD, TIPCOD, TIPO, ORDEN, NOM\_IMAGEN, URL, IDMENUNOM, AYUDA\_MANUAL, AYUDA\_VIDEO, URL\_OLD, HABILITAR\_URL\_OLD)  
VALUES ('21', '810000004', '0000', 'Tablero de Resultados', '810000000', NULL, 1, 2, 'list\_alt', 'intranet.gestionCalidad.resultados', NULL, NULL, NULL, NULL, 0)  
MATCHING (SISCOD, MENUCOD);

CREATE OR ALTER TRIGGER CC\_INDICADOR\_AI0 FOR CC\_INDICADOR

ACTIVE AFTER INSERT POSITION 0

as

begin

insert into CC\_ACCESO\_INDICADOR (IDINDICADOR, TIPCOD)

values (new.IDINDICADOR, '001');

end;

CREATE GENERATOR GEN\_CC\_ACCESO\_INDICADOR\_ID;

CREATE OR ALTER TRIGGER CC\_ACCESO\_INDICADOR\_BI FOR CC\_ACCESO\_INDICADOR ACTIVE BEFORE INSERT POSITION 0

as

begin

if (new.idacceso is null) then

new.idacceso = gen\_id(gen\_cc\_acceso\_indicador\_id,1);

end;

ALTER TABLE CC\_INDICADOR DROP TIPO\_CUMPLIMIENTO;

ALTER TABLE CC\_INDICADOR ALTER COLUMN TIPO\_CUMPLIMIENTO SET DEFAULT 'P';