|  |  |
| --- | --- |
| **UNIVERSIDAD AUTONOMA DE COAHUILA** |  |
| BASE DE DATOS I |
| Proyecto Final. | |

TABLA DE CONTENIDO

1. Propósito del Documento. 2

2. Antecedentes. 2

3. Propósito del Proyecto. 2

3.1 Objetivo general. 2

4. Alcance del Proyecto. 2

4.1 Objetivos Específicos. 2

5. Diagramas Entidad-Relacción. 2

5.1 Diagrama E-R y cardinalidad. 2

6. Diccionario de datos. 3

6.1 Diccionario de datos tablas, atributos, trigger, view, function y stored procedure. 3

6.2 Script de creación de tablas, llaves primarias y fopraneas. 3

6.3 Script para poblar las tablas. 3

6.4 Script para crear las vistas, trigger, function y stored procedure. 4

6.5 Script de para obtener los requerimientos del proyecto.. 5

7. Implementación del modelo Físico. 7

# Propósito del Documento.

El presente documento tiene como propósito dar a conocer el contexto del proyecto, los requrimientos solicitados, la forma de elaboración y el termino del proyecto con funcionalidad.

# Antecedentes.

Descripción general y detallada del proyecto, sus alcances y reglas de negocio del proyecto.

# Propósito del Proyecto.

## Objetivo general.

El propósito de este proyecto es el de diseñar una base de datos donde se pueda almacenar la información de las consultas médicas por especialidad que se realizan en la clínica “SANITO”.

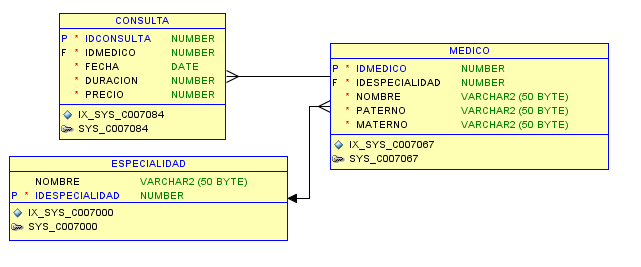
# Alcance del Proyecto.

Descripción detallada de la propuesta de solución.

## Objetivos Específicos.

# Diagramas Entidad-Relacción.

## Diagrama E-R y cardinalidad.



# Diccionario de datos.

## Diccionario de datos tablas, atributos, trigger, view, function y stored procedure.

## Script de creación de tablas, llaves primarias y fopraneas.

Script para crear tabla Consulta

CREATE TABLE CONSULTA(

IDCONSULTA NUMBER NOT NULL,

DURACION NUMBER NOT NULL,

PRECIO NUMBER NOT NULL,

FECHA DATE NOT NULL,

IDMEDICO NUMBER NOT NULL,

PRIMARY KEY (IDCONSULTA),

FOREIGN KEY (IDMEDICO) REFERENCES MEDICO

);

Script para crear tabla Especialidad

CREATE TABLE ESPECIALIDAD(

NOMBRE VARCHAR2(50),

IDESPECIALIDAD NUMBER NOT NULL,

PRIMARY KEY (IDESPECIALIDAD)

);

Script para crear tabla Médico

CREATE TABLE MEDICO (

IDMEDICO NUMBER NOT NULL,

IDESPECIALIDAD NUMBER NOT NULL,

NOMBRE VARCHAR2(50) NOT NULL,

PATERNO VARCHAR2(50) NOT NULL,

MATERNO VARCHAR2(50) NOT NULL,

PRIMARY KEY (IDMEDICO),

FOREIGN KEY (IDESPECIALIDAD) REFERENCES ESPECIALIDAD

);

## Script para poblar las tablas.

Script para poblar tabla Consulta

INSERT INTO CONSULTA (IDCONSULTA, IDMEDICO, FECHA, DURACION, PRECIO) VALUES(1,1,TO\_DATE('01062016','DDMMYYYY'),1,800);

Script para poblar tabla Especialidad

INSERT INTO ESPECIALIDAD (IDESPECIALIDAD, NOMBRE) VALUES(1,'Alergología');

Script para poblar tabla Medico

INSERT INTO MEDICO (IDMEDICO, IDESPECIALIDAD, NOMBRE, PATERNO, MATERNO) VALUES(1,15,'Luis Felipe','Montes','Valero');

## Script para crear las vistas, trigger, function y stored procedure.

Script para crear Vistas

CREATE OR REPLACE VIEW MEDICO\_Y\_CONSULTAS

AS

SELECT M.IDMEDICO

,M.IDESPECIALIDAD

,M.NOMBRE

,M.PATERNO

,M.MATERNO

,C.IDCONSULTA

,C.FECHA

,C.DURACION

,CPRECIO

FROM MEDICO M

INNER JOIN CONSULTA C

ON M.IDMEDICO = C.IDMEDICO;

CREATE OR REPLACE VIEW MEDICO\_HORAS

AS

SELECT M.NOMBRE , I.PATERNO , I.MATERNO, SUM(I.DURACION) HORAS FROM INFORMACION\_COMPLETA I

INNER JOIN MEDICO M

ON I.IDMEDICO=M.IDMEDICO

GROUP BY M.NOMBRE, I.PATERNO, I.MATERNO;

CREATE OR REPLACE VIEW INFORMACION\_COMPLETA

AS

SELECT E.NOMBRE ESPECIALIDAD

,M.IDMEDICO

,M.IDESPECIALIDAD

,M.NOMBRE

,M.PATERNO

,M.MATERNO

,M.IDCONSULTA

,M.FECHA

,M.DURACION

,M.PRECIO

FROM MEDICO\_Y\_CONSULTAS M

INNER JOIN ESPECIALIDAD E

ON M.IDESPECIALIDAD=E.IDESPECIALIDAD;

CREATE OR REPLACE VIEW ESPECIALIDAD\_HORAS

AS

SELECT E.NOMBRE , SUM(I.DURACION) HORAS FROM INFORMACION\_COMPLETA I

INNER JOIN ESPECIALIDAD E

ON I.IDESPECIALIDAD=E.IDESPECIALIDAD

GROUP BY E.NOMBRE;

## Script de para obtener los requerimientos del proyecto..

Script para Obtener Requerimiento (Stored Procedure)

Soluciona el requerimiento número 3: Ingreso por consultas en un rango de fechas por cada especialidad.

CREATE OR REPLACE PROCEDURE INGRESO\_ESPECIALIDAD\_FECHAS(FECHA01 IN DATE, FECHA02 IN DATE)

IS

NUM INFORMACION\_COMPLETA.PRECIO%TYPE;

NOM ESPECIALIDAD.NOMBRE%TYPE;

CURSOR C1 IS

SELECT E.NOMBRE

,SUM (I.PRECIO) PRECIO

FROM INFORMACION\_COMPLETA I

INNER JOIN ESPECIALIDAD E

ON E.IDESPECIALIDAD=I.IDESPECIALIDAD

WHERE I.FECHA BETWEEN FECHA01 AND FECHA02

GROUP BY E.NOMBRE;

BEGIN

OPEN C1;

DBMS\_OUTPUT.PUT\_LINE('El ingreso entre ' || FECHA01 || ' y ' || FECHA02 || ' es :');

LOOP

FETCH C1 INTO NOM , NUM;

EXIT WHEN C1%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Especialidad: ' || NOM);

DBMS\_OUTPUT.PUT\_LINE('Ingresos: ' || NUM );

DBMS\_OUTPUT.PUT\_LINE('--------------------------------------------');

COMMIT;

END LOOP;

CLOSE C1;

END;

Soluciona el requerimiento número 4: Ingreso por médico en un rango de fechas ordenado de mayor a menor.

CREATE OR REPLACE

PROCEDURE INGRESO\_MEDICO\_FECHAS (FECHA01 IN DATE , FECHA02 IN DATE)

IS

NUM INFORMACION\_COMPLETA.PRECIO%TYPE;

NOM MEDICO.NOMBRE%TYPE;

PAT MEDICO.PATERNO%TYPE;

MAT MEDICO.MATERNO%TYPE;

CURSOR C1 IS

SELECT M.NOMBRE, M.PATERNO, M.MATERNO

,SUM (I.PRECIO) PRECIO

FROM INFORMACION\_COMPLETA I

INNER JOIN MEDICO M

ON M.IDMEDICO=I.IDMEDICO

WHERE I.FECHA BETWEEN FECHA01 AND FECHA02 GROUP BY M.NOMBRE, M.PATERNO, M.MATERNO;

BEGIN

OPEN C1;

DBMS\_OUTPUT.PUT\_LINE('El ingreso por médico entre ' || FECHA01 || ' y ' || FECHA02 || ' es :');

LOOP

FETCH C1 INTO NOM , PAT , MAT, NUM;

EXIT WHEN C1%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Médico: ' || NOM || ' ' || PAT || ' ' || MAT);

DBMS\_OUTPUT.PUT\_LINE('Ingresos: ' || NUM );

DBMS\_OUTPUT.PUT\_LINE('--------------------------------------------');

COMMIT;

END LOOP;

CLOSE C1;

END;

Script para Obtener Requerimiento (Function)

Utilizado Para Requerimiento 1: Total en horas de consulta realizadas por médico

CREATE OR REPLACE

FUNCTION TOTAL\_DE\_HORAS\_POR\_MEDICO (ID NUMBER)

RETURN NUMBER IS

TOTAL NUMBER;

BEGIN

SELECT SUM(DURACION) INTO TOTAL

FROM CONSULTA

WHERE IDMEDICO=ID;

RETURN TOTAL;

END;

Utilizado Para Requerimiento 4: Ingreso por médico en un rango de fechas ordenado de mayor a menor.

CREATE OR REPLACE

FUNCTION TOTAL\_INGRESOS(ID NUMBER)

RETURN NUMBER IS

TOTAL NUMBER;

BEGIN

SELECT SUM(PRECIO) INTO TOTAL

FROM CONSULTA

WHERE IDMEDICO = ID;

RETURN TOTAL;

END;

# Implementación del modelo Físico.