Universidad Nacional Experimental del Táchira  
Vicerrectorado Académico  
Decanato de Docencia  
Departamento de Ingeniería Informática  
Lapso 2021-3

**Desarrollo de una base de datos para una aplicación de gestión de horarios de una universidad privada**

**Autores**

Camargo Meaury, Franklin Josué; C.I: V.-27655989

Contreras Rojas, Reyner David; C.I: V.-26934400

Vargas Rueda, Brandon José; C.I: V.-26566047

**Número de grupos:** 3 y 4

**Materia y sección:** Base de datos I sección 01

Octubre 2021

**Introducción**

Para toda organización, es importante mantener un orden concreto para lograr alcanzar los objetivos de la misma; para ello existe personal que se encarga de coordinar a los demás empleados y a su vez utilizan herramientas y sistemas para dicho fin.

Una de las herramientas utilizadas para la organización del personal son los horarios, que dependiendo del mismo indica las horas y los días en que los mismos deben realizar sus tareas de trabajo. Estos horarios se pueden crear y mantener utilizando sistemas informáticos de manera que su gestión sea más sencilla.

Las universidades son organizaciones cuyo fin es la enseñanza y están conformadas por el personal que trabaja en la misma (entre los que se encuentra el personal docente) y por estudiantes. Para poder mantener el orden es necesario que tenga un sistema de información que funcione correctamente de manera que se facilite la organización de la misma, como por ejemplo un sistema de gestión de horarios.

Para el proyecto actual, un vicerrector académico de una universidad privada necesita que se desarrolle un sistema de base de datos relacional en Oracle para una aplicación que genera y gestiona horarios de profesores y de estudiantes. En una reunión previa entre los involucrados del proyecto ya se definieron las restricciones, necesidades y requerimientos de datos e información necesarios para el análisis, diseño e implementación de la base de datos.

**Objetivos**

**General:** Desarrollar una base de datos para una aplicación de gestión de horarios de una universidad privada.

**Específicos:**

* Monitorear, controlar y dar seguimiento de la asistencia a clases de los profesores
* Generar información oportuna de la disponibilidad horaria del personal docente.
* Optimizar el uso de las aulas de clase.
* Evitar el choque de horarios de las asignaturas inscritas por los estudiantes
* Brindar un servicio oportuno, eficiente y efectivo de asignación de horarios en forma óptima.

**Fase de requisitos**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CONTROL DE VERSIONES | | | | | |
| Versión | Hecho por | Revisado por | Aprobado por | Fecha | Motivo |
| 1.0.0 | Brandon. V -Franklin. C -  Reyner. C |  |  | 11/10/2021 |  |
| 1.1.0 | Brandon. V -Franklin. C -  Reyner. C |  |  | 23/10/2021 |  |
| 1.2.0 |  |  |  | 01/11/2021 |  |

*D O C U M E N T A C I Ó N D E R E Q U I S I T O S*

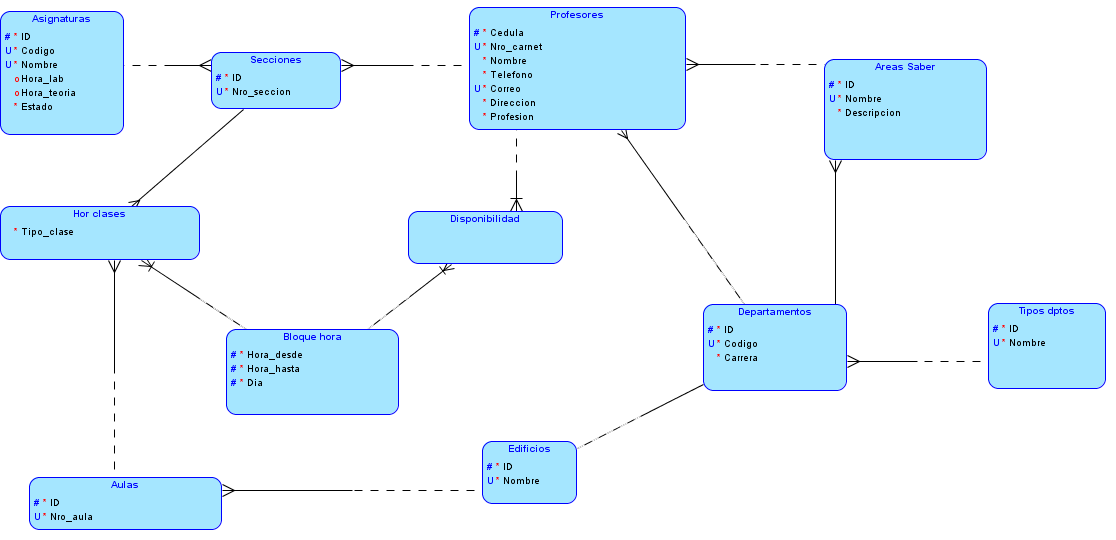
*D E D A T O S*

|  |  |
| --- | --- |
| NOMBRE DEL ESQUEMA DE BASE DATOS | SIGLAS |
| Gestión de horarios | GDH |

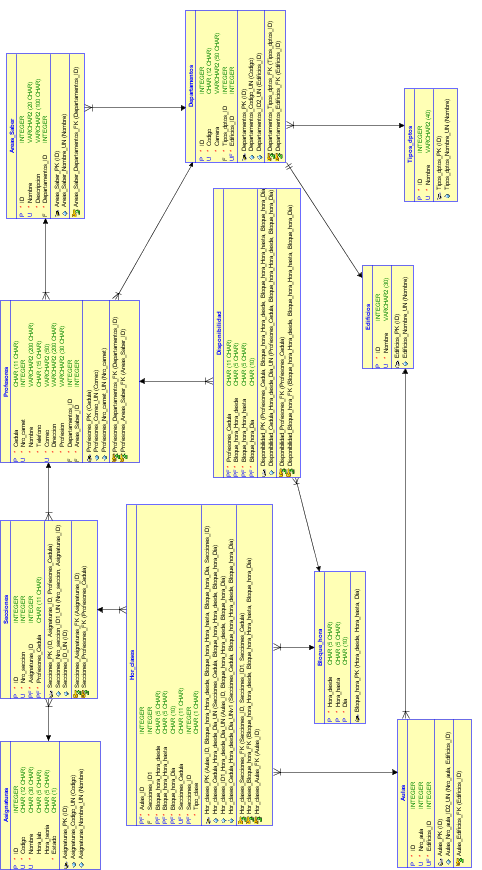
|  |  |  |  |
| --- | --- | --- | --- |
| NECESIDAD DEL NEGOCIO U OPORTUNIDAD A APROVECHAR: DESCRIBIR LAS LIMITACIONES DE LA SITUACIÓN ACTUAL Y LAS RAZONES POR LAS CUÁLES SE EMPRENDE EL PROYECTO | | | |
| Actualmente en la universidad privada no hay buen manejo en cuanto a la gestión de horarios. Como consecuencia, tenemos frecuentes choques de horario al momento de inscripción de las materias. También se puede observar que no se tiene una buena gestión que permita dar información indicando asistencia de los profesores a clases, A demás no se tiene información para la disponibilidad docente. Son muchas cosas importantes que se deben llevar a cabo en una universidad, ya que la información de datos es sumamente importante para llevar control. Por eso se ha decidido emprender un proyecto de Bases de datos. Que nos brinden esa información para la disponibilidad de horarios, para dar la información necesaria a la Universidad. Todo el campus Universitarios. Se beneficia con el rendimiento y eficacia de esta nueva base de datos, tanto para mejorar los horarios y disponibilidad de los profesores para los estudiantes, y sus clases. | | | |
| DEFINIR EL ALCANCE: DEFINIR CON CLARIDAD EL AREA DEL NEGOCIO A LA CUAL LA BASE DE DATOS DARÁ APOYO | | | |
| El propósito de este proyecto en analizar, diseñar e implementar una base de datos relacional en Oracle lo suficientemente eficiente, efectiva y flexible que sirva de soporte a la aplicación de gestión de horarios, y brindar información detallada tanto del docente y su función. Para ello se aplicara la metodología de desarrollo de base de datos. | | | |
| OBJETIVOS DEL NEGOCIO: DEFINIR CON CLARIDAD LOS OBJETIVOS DEL NEGOCIO PARA EL ALCANCE ESTABLECIDO | | | |
| Monitorear, controlar y dar seguimiento de la asistencia a clases de los profesores  Generar información oportuna de la disponibilidad horaria del personal docente.  Optimizar el uso de las aulas de clase.  Evitar el choque de horarios de las asignaturas inscritas por los estudiantes  Brindar un servicio oportuno, eficiente y efectivo de asignación de horarios en forma óptima. | | | |
| REQUISITOS DE DATOS: DEFINIR CON CLARIDAD LAS NECESIDADES DE DATOS A CONSIDERAR PARA EL DISEÑO DE LA BASE DE DATOS | | | |
| STAKEHOLDER (INTERESADO | PRIORIDAD OTORGADA POR EL STAKEHOLDER | REQUISITOS | |
| CODIGO | DESCRIPCION |
| El Vicerrector Académico | 1 |  |  |
| Docentes disponibles | 2 |  | Disponibilidad de docentes |
| Inscripciones máximas | 3 |  | Cuantos alumnos se pueden inscribir por sección |
| Horas máxima de trabajo | 4 |  | Cuantas horas estipuladas de trabajo tiene cada docente |
| Secciones máxima | 5 |  | Máximo de secciones abiertas por asignatura |
|  |  |  |  |
| REQUISITOS NO FUNCIONALES: DESCRIBIR LOS REQUISITOS TALES CÓMO HARDWARE, SOFTWARE, NIVEL DE SERVICIO, PERFOMANCE, SEGURIDAD, ETC. | | | |
| STAKEHOLDER (INTERESADO) | PRIORIDAD OTORGADA POR EL STAKEHOLDER | REQUISITOS | |
| CODIGO | DESCRIPCION |
| Oracle | 1 |  | Interprete de base de datos |
| SQL Developer Data Modeler | 2 |  | Modelador de E-R |
| Aulas | 3 |  | Disposición de aulas |
|  |  |  |  |
|  |  |  |  |
| REGLAS DEL NEGOCIO Y RESTRICCIONES: REGLAS PRINCIPALES QUE FIJAN EL MANEJO Y RESTICCIONES DE LOS DATOS | | | |
| Las asignaturas son identificadas por código y un nombre, y se le especifican las horas de clase de teoría o de laboratorio o ambas.  Una asignatura puede tener abiertas una o más secciones dependiendo de las demanda.  Los profesores son identificados por su número de carnet, cedula, nombre, dirección, teléfono, correo y profesión.  Las secciones de las asignaturas son impartidas por los profesores, y los profesores no puede impartir más de 16 horas semanales.  Los profesores están adscritos a los departamentos académicos. Los departamentos son de dos tipos, de carrera y de apoyo (ciencias básicas). Un profesor está adscrito a un solo de departamento académico, bien sea de carrera o de apoyo, y un departamento puede tener adscritos muchos profesores. Los departamentos académicos son identificados por un código, carrera y ubicación (edificio), en un edificio no puede haber más de un departamento.  Las aulas están ubicadas en los edificios, y están edificadas con una numeración.  Los horarios de las asignaturas y sus secciones son asignadas a las aulas. Las aulas tienen horarios de clases asignadas por cada día de la semana donde se especifica la hora desde y hasta de la clase (bloque de clase), y el profesor que la dictara. No deben existir diferentes clases simultáneas en una misma aula, en el mismo bloque de horario y el mismo profesor.  Debe contemplar los horarios de disponibilidad de los profesores por día en la semana donde se especifique la hora desde y hora hasta disponible.  La plataforma tecnológica de base de datos a utilizar es en Oracle Enterprice en la nube (AWS) como servidor de base de datos y sistema operativo Linux. | | | |

**Fase de diseño conceptual o lógico**

A continuación se presenta una imagen con el diseño conceptual o lógico realizado para el proyecto:



**Modelo relacional o diagrama de tablas**



**Checks (nota, faltan checks en el documento)**

Entidad Tipos dptos

check(Tipos\_dptos in('carrera','apoyo'))

Entidad Aulas

check(Nro\_aula>0)

Entidad Bloques horas

check(Dia in (‘lunes’, ‘martes’, ‘miércoles’, ‘jueves’, ‘viernes’, ‘sábado’, ‘domingo’))

Entidad Personas

check(Cedula>0)

check(Correo like '%\_\_\_@\_\_\_%')

check(Telefono like ‘%[0-9]%’)

Entidad Asignaturas

check(Hora\_lab is not null or Hora\_teoria is not null)

check(Estado in (‘A’, ‘I’, ‘en actualizacion’)

Entidad Secciones

check(Nro\_seccion > 0 AND Nro\_seccion < 100)

**Código SQL para generar la base de datos**



/\*

Creado por los equipos 3 y 4

Base de datos I sección 1 2021-3

Autores:

Camargo Meaury, Franklin Josué; C.I: V.-27655989

Contreras Rojas, Reyner David; C.I: V.-26934400

Vargas Rueda, Brandon José; C.I: V.-26566047

\*/

CREATE TABLE tipos\_dptos(

id INT GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1) NOT NULL,

nombre VARCHAR2(40) NOT NULL,

CONSTRAINT tipos\_dptos\_pk PRIMARY KEY (id),

CONSTRAINT tipos\_dptos\_ck CHECK (nombre IN ('CARRERA', 'APOYO')),

CONSTRAINT tipos\_dptos\_uk UNIQUE (nombre)

);

/\* Trigger de tipos\_dptos para que el nombre se almacene en mayusculas \*/

CREATE OR REPLACE TRIGGER touppercasetiposdpto

BEFORE INSERT OR UPDATE ON tipos\_dptos

FOR EACH ROW

BEGIN

:NEW.nombre := UPPER(:NEW.nombre);

END;

/

CREATE TABLE edificios(

id INT GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1) NOT NULL,

nombre VARCHAR2(30) NOT NULL,

CONSTRAINT edificios\_uk UNIQUE (nombre),

CONSTRAINT edificios\_pk PRIMARY KEY (id)

);

CREATE TABLE departamentos(

id INT GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1) NOT NULL,

codigo VARCHAR2(12) NOT NULL,

carrera VARCHAR2(50) NOT NULL,

tipos\_dptos\_id INT NOT NULL,

edificios\_id INT NOT NULL,

CONSTRAINT dptos\_uk1 UNIQUE (codigo),

CONSTRAINT dptos\_uk2 UNIQUE (edificios\_id),

CONSTRAINT dptos\_edificios\_fk

FOREIGN KEY (edificios\_id) REFERENCES edificios(id)

ON DELETE SET NULL,

CONSTRAINT dptos\_tipos\_fk

FOREIGN KEY (tipos\_dptos\_id) REFERENCES tipos\_dptos(id)

ON DELETE SET NULL,

CONSTRAINT dptos\_pk PRIMARY KEY (id)

);

CREATE TABLE aulas(

id INT GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1) NOT NULL,

nro\_aula smallint NOT NULL,

edificios\_id int NOT NULL,

CONSTRAINT aulas\_pk PRIMARY KEY (id),

CONSTRAINT aulas\_uk UNIQUE (nro\_aula, edificios\_id),

CONSTRAINT aulas\_ck CHECK (nro\_aula > 0) ,

CONSTRAINT aulas\_edificios\_fk

FOREIGN KEY (edificios\_id) REFERENCES edificios(id)

ON DELETE CASCADE

);

CREATE TABLE areas\_saber(

id INT GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1) NOT NULL,

nombre VARCHAR2(20) NOT NULL,

descripcion VARCHAR2(100) NOT NULL,

dpto\_id INT NOT NULL,

CONSTRAINT areas\_saber\_pk PRIMARY KEY (id),

CONSTRAINT areas\_saber\_uk UNIQUE (nombre),

CONSTRAINT areas\_saber\_dpto\_fk

FOREIGN KEY (dpto\_id) REFERENCES departamentos(id)

ON DELETE CASCADE

);

CREATE TABLE asignaturas(

id INT GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1) NOT NULL,

codigo VARCHAR2(12) NOT NULL,

nombre VARCHAR2(30) NOT NULL,

hora\_lab CHAR(5) DEFAULT NULL,

hora\_teoria CHAR(5) DEFAULT NULL,

estado CHAR(1) NOT NULL,

CONSTRAINT asignaturas\_pk PRIMARY KEY (id),

CONSTRAINT asignaturas\_uk1 UNIQUE (codigo),

CONSTRAINT asignaturas\_uk2 UNIQUE (nombre),

CONSTRAINT asignaturas\_ck1 CHECK (estado IN ('A', 'I')),

CONSTRAINT asignaturas\_ck2

CHECK (hora\_lab IS NOT NULL OR hora\_teoria IS NOT NULL)

);

/\* Trigger de asignaturas para que el nombre se almacene en mayusculas \*/

CREATE OR REPLACE TRIGGER touppercaseasig

BEFORE INSERT OR UPDATE ON asignaturas

FOR EACH ROW

BEGIN

:NEW.nombre := UPPER(:NEW.nombre);

END;

/

CREATE TABLE profesores(

cedula CHAR(11) NOT NULL,

nro\_carnet INT NOT NULL,

nombre VARCHAR2(200) NOT NULL,

telefono CHAR(15) NOT NULL,

correo VARCHAR2(50) NOT NULL,

direccion VARCHAR2(200) NOT NULL,

profesion VARCHAR2(30) NOT NULL,

dpto\_id INT NOT NULL,

areas\_saber\_id INT NOT NULL,

CONSTRAINT profesores\_pk PRIMARY KEY (cedula),

CONSTRAINT profesores\_uk1 UNIQUE (nro\_carnet),

CONSTRAINT profesores\_uk2 UNIQUE (correo),

/\*

Por alguna extrana razon el checks de telefono no validan

correctamente. Problemas al validar que todo sea numeros.

\*/

CONSTRAINT profesores\_ck1

CHECK (

REGEXP\_LIKE(cedula, '((^E)|(^V))[0-9]')

AND TO\_NUMBER(SUBSTR(cedula, 2)) > 0

),

CONSTRAINT profesores\_ck2 CHECK (nro\_carnet > 0),

CONSTRAINT profesores\_ck3 CHECK (REGEXP\_LIKE(telefono, '[0-9]+')),

/\* Check para longitud de numeros, variable por si incluye codigo de pais \*/

CONSTRAINT profesores\_ck4 CHECK (LENGTH(telefono) > 10),

CONSTRAINT profesores\_ck5 CHECK

(REGEXP\_LIKE(correo,'^[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}$')),

CONSTRAINT profesores\_dpto\_fk

FOREIGN KEY (dpto\_id) REFERENCES departamentos(id)

ON DELETE SET NULL,

CONSTRAINT profesores\_areas\_saber\_fk

FOREIGN KEY (areas\_saber\_id) REFERENCES areas\_saber(id)

ON DELETE SET NULL

);

/\* Trigger de profesores para que el nombre se almacene en mayusculas \*/

CREATE OR REPLACE TRIGGER touppercaseprof

BEFORE INSERT OR UPDATE ON profesores

FOR EACH ROW

BEGIN

:NEW.nombre := UPPER(:NEW.nombre);

:NEW.cedula := UPPER(:NEW.cedula);

END;

/

CREATE TABLE secciones(

id INT GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1) NOT NULL,

nro\_seccion INT NOT NULL,

asignaturas\_id INT NOT NULL,

profesores\_cedula CHAR(11) NOT NULL,

CONSTRAINT secciones\_uk1 UNIQUE (nro\_seccion, asignaturas\_id),

CONSTRAINT secciones\_uk2 UNIQUE (id),

CONSTRAINT secciones\_ck CHECK (nro\_seccion > 0 AND nro\_seccion < 100),

CONSTRAINT secciones\_asignaturas\_fk

FOREIGN KEY (asignaturas\_id) REFERENCES asignaturas(id)

ON DELETE CASCADE,

CONSTRAINT secciones\_profesores\_fk

FOREIGN KEY (profesores\_cedula) REFERENCES profesores(cedula)

ON DELETE SET NULL,

CONSTRAINT secciones\_pk PRIMARY KEY (id, profesores\_cedula)

);

/\* Trigger de secciones para que cedula se almacene en mayusculas \*/

CREATE OR REPLACE TRIGGER touppercasesecc

BEFORE INSERT OR UPDATE ON secciones

FOR EACH ROW

BEGIN

:NEW.profesores\_cedula := UPPER(:NEW.profesores\_cedula);

END;

/

CREATE TABLE bloque\_hora(

hora\_desde CHAR(5) NOT NULL,

hora\_hasta CHAR(5) NOT NULL,

dia CHAR(10) NOT NULL,

CONSTRAINT bloque\_hora\_pk PRIMARY KEY (hora\_desde, hora\_hasta, dia),

CONSTRAINT bloque\_hora\_ck1

CHECK (dia IN ('LUNES', 'MARTES', 'MIERCOLES', 'JUEVES',

'VIERNES', 'SABADO', 'DOMINGO')),

CONSTRAINT bloque\_hora\_ck2

CHECK (

REGEXP\_LIKE(hora\_desde,'^([0-9]{2,2}):([0-9]{2,2})$') AND

REGEXP\_LIKE(hora\_hasta,'^([0-9]{2,2}):([0-9]{2,2})$')

),

CONSTRAINT bloque\_hora\_ck3 CHECK (

TO\_NUMBER(SUBSTR(hora\_desde, 1, 2)) < 24 AND

TO\_NUMBER(SUBSTR(hora\_desde, 4, 2)) < 60

),

CONSTRAINT bloque\_hora\_ck4 CHECK (

TO\_NUMBER(SUBSTR(hora\_hasta, 0, 2)) < 24 AND

TO\_NUMBER(SUBSTR(hora\_hasta, 4, 2)) < 60

),

CONSTRAINT bloque\_hora\_ck5 CHECK (

TO\_NUMBER(REPLACE(hora\_desde,':', '')) <=

TO\_NUMBER(REPLACE(hora\_hasta,':', ''))

)

);

/\* Trigger de bloque\_hora para que el dia se almacene en mayusculas \*/

CREATE OR REPLACE TRIGGER touppercasebloquehora

BEFORE INSERT OR UPDATE ON bloque\_hora

FOR EACH ROW

BEGIN

:NEW.dia := UPPER(:NEW.dia);

END;

/

CREATE TABLE disponibilidad(

profesores\_cedula CHAR(11) NOT NULL,

bloque\_hora\_desde CHAR(5) NOT NULL,

bloque\_hora\_hasta CHAR(5) NOT NULL,

bloque\_dia CHAR(10) NOT NULL,

CONSTRAINT disponibilidad\_profesor\_fk

FOREIGN KEY (profesores\_cedula) REFERENCES profesores(cedula)

ON DELETE CASCADE,

CONSTRAINT disponibilidad\_hora\_fk

FOREIGN KEY (bloque\_hora\_desde, bloque\_hora\_hasta, bloque\_dia)

REFERENCES bloque\_hora(hora\_desde, hora\_hasta, dia) ON DELETE CASCADE,

CONSTRAINT disponibilidad\_uk

UNIQUE (profesores\_cedula, bloque\_hora\_desde, bloque\_dia),

CONSTRAINT disponibilidad\_hora\_pk

PRIMARY KEY

(profesores\_cedula, bloque\_hora\_desde, bloque\_hora\_hasta, bloque\_dia)

);

/\* Trigger de disponibilidad para que cedula y dia se almacenen en mayusculas \*/

CREATE OR REPLACE TRIGGER touppercasedisp

BEFORE INSERT OR UPDATE ON disponibilidad

FOR EACH ROW

BEGIN

:NEW.bloque\_dia := UPPER(:NEW.bloque\_dia);

:NEW.profesores\_cedula := UPPER(:NEW.profesores\_cedula);

END;

/

CREATE TABLE hor\_clases(

profesores\_cedula CHAR(11) NOT NULL,

secciones\_id INT NOT NULL,

aulas\_id INT NOT NULL,

bloque\_hora\_desde CHAR(5) NOT NULL,

bloque\_hora\_hasta CHAR(5) NOT NULL,

bloque\_dia CHAR(10) NOT NULL,

tipo\_clase CHAR(1) NOT NULL,

CONSTRAINT hor\_clases\_aulas\_fk

FOREIGN KEY (aulas\_id) REFERENCES aulas(id)

ON DELETE SET NULL,

CONSTRAINT hor\_clases\_secciones\_fk

FOREIGN KEY (secciones\_id, profesores\_cedula)

REFERENCES secciones(id, profesores\_cedula)

ON DELETE CASCADE,

CONSTRAINT hor\_clases\_bloque\_hora\_fk

FOREIGN KEY (bloque\_hora\_desde, bloque\_hora\_hasta, bloque\_dia)

REFERENCES bloque\_hora(hora\_desde, hora\_hasta, dia) ON DELETE CASCADE,

CONSTRAINT hor\_clases\_uk1

UNIQUE (profesores\_cedula, bloque\_hora\_desde, bloque\_dia),

CONSTRAINT hor\_clases\_uk2 /\* en teoria no deberia hacer falta \*/

UNIQUE (secciones\_id, bloque\_hora\_desde, bloque\_dia),

CONSTRAINT hor\_clases\_uk3

UNIQUE (aulas\_id, bloque\_hora\_desde, bloque\_dia),

CONSTRAINT hor\_clases\_uk4 CHECK (tipo\_clase IN ('T', 'L')),

CONSTRAINT hor\_clases\_pk

PRIMARY KEY

(secciones\_id, aulas\_id, bloque\_hora\_desde, bloque\_hora\_hasta, bloque\_dia)

);

/\* Trigger de hor\_clases para que cedula, dia y tipo\_clase dia se almacenen

en mayusculas \*/

CREATE OR REPLACE TRIGGER touppercasehor\_clases

BEFORE INSERT OR UPDATE ON hor\_clases

FOR EACH ROW

BEGIN

:NEW.bloque\_dia := UPPER(:NEW.bloque\_dia);

:NEW.profesores\_cedula := UPPER(:NEW.profesores\_cedula);

:NEW.tipo\_clase := UPPER(:NEW.tipo\_clase);

END;

/

-- Falta hacer checks de que profesor tenga 16 horas o menos de clase a la semana

/\* AQUI EMPIEZAN LOS INSERT \*/

-- tipos\_dptos

INSERT INTO tipos\_dptos (nombre)

VALUES ('CARRERA');

INSERT INTO tipos\_dptos (nombre)

VALUES ('APOYO');

--ERROR

INSERT INTO tipos\_dptos (nombre)

VALUES ('');

INSERT INTO tipos\_dptos (nombre)

VALUES ('CIENCIAS');

INSERT INTO tipos\_dptos (nombre)

VALUES ('apoyo');

SELECT \* FROM TIPOS\_DPTOS;

-- edificios

INSERT INTO edificios (nombre)

VALUES ('A');

INSERT INTO edificios (nombre)

VALUES ('B');

--ERROR

INSERT INTO edificios (nombre)

VALUES ('');

SELECT \* FROM EDIFICIOS;

-- departamentos-------------------

INSERT INTO departamentos(codigo,carrera,tipos\_dptos\_id,edificios\_id)

VALUES(123,'matematica',1,1); --carrera edi\_A

INSERT INTO departamentos(codigo,carrera,tipos\_dptos\_id,edificios\_id)

VALUES(1234,'informatica',1,2);--carrera edi\_B

--ERROR

INSERT INTO departamentos(codigo,carrera,tipos\_dptos\_id,edificios\_id)

VALUES(123,'electronica',1,1); --carrera edi\_A

INSERT INTO departamentos(codigo,carrera,tipos\_dptos\_id,edificios\_id)

VALUES(1,'matematica',1,1); --carrera edi\_A

INSERT INTO departamentos(codigo,carrera,tipos\_dptos\_id,edificios\_id)

VALUES(123,1,1); --carrera edi\_A

SELECT \* FROM DEPARTAMENTOS;

-- aulas

INSERT INTO aulas (nro\_aula,edificios\_id)

VALUES ('1',1); -- nro 1 edificio A

INSERT INTO aulas (nro\_aula,edificios\_id)

VALUES ('2',1); -- nro 2 edificio A

INSERT INTO aulas (nro\_aula,edificios\_id)

VALUES ('1',2); -- nro 1 edificio B

INSERT INTO aulas (nro\_aula,edificios\_id)

VALUES ('2',2); -- nro 2 edificio B

-- error

INSERT INTO aulas (nro\_aula,edificios\_id)

VALUES ('1',1); -- ya existe

INSERT INTO aulas (nro\_aula,edificios\_id)

VALUES ('1',3); -- edificio de ID 3 no existe

SELECT \* FROM AULAS;

-- areas\_saber

INSERT INTO areas\_saber (nombre,descripcion,dpto\_id)

VALUES ('humanidades','Humanidad1',1);

INSERT INTO areas\_saber (nombre,descripcion,dpto\_id)

VALUES ('arte','Arte.',2);

-- error

INSERT INTO areas\_saber (nombre,descripcion,dpto\_id)

VALUES ('humanidades','humanidad2',1);

SELECT \* FROM AREAS\_SABER;

-- asignaturas

INSERT INTO asignaturas (codigo,nombre,hora\_lab,hora\_teoria,estado)

VALUES ('123','matematica','3','2','A');

INSERT INTO asignaturas (codigo,nombre,hora\_lab,hora\_teoria,estado)

VALUES ('124','fisica','3','2','I');

INSERT INTO asignaturas (codigo,nombre,hora\_teoria,estado)

VALUES ('125','historia','4','I');

INSERT INTO asignaturas (codigo,nombre,hora\_lab,estado)

VALUES ('126','lab fisica','4','I');

-- error

INSERT INTO asignaturas (codigo,nombre,hora\_lab,hora\_teoria,estado)

VALUES ('123','calculo','3','2','A'); -- codigo

INSERT INTO asignaturas (codigo,nombre,hora\_lab,hora\_teoria,estado)

VALUES ('127','algebra lineal','3','2','P'); -- estado no valido

INSERT INTO asignaturas (codigo,nombre,hora\_lab,hora\_teoria,estado)

VALUES ('128','matematica','1','1','A'); -- nombre

INSERT INTO asignaturas (codigo,nombre,estado)

VALUES ('129','matematica I','A'); -- ambas horas nulas

INSERT INTO asignaturas (codigo,nombre,hora\_lab,hora\_teoria,estado)

VALUES ('130','matemati','','','A'); -- codigo

SELECT \* FROM ASIGNATURAS;

-- PROFESORES

INSERT INTO profesores

VALUES ('v12123123',1,'jose barrera','121231231231','jose@correo.com','Sc edo tachira','docente',1,1);

INSERT INTO profesores

VALUES ('e12123123',2,'maria sandoval','121231231230','mariajo@correo.com','pueblo nuevo','docente',1,1);

INSERT INTO profesores

VALUES ('v23545444',3,'jose barrera','121231231231','jose3@correo.com','Sc edo tachira','docente',1,1);

-- ERROR

INSERT INTO profesores

VALUES ('v12123123',4,'jose barrera','121231','jose@correo','Sc edo tachira','docente',1,1); -- cedula

INSERT INTO profesores

VALUES ('V0',5,'jose barrera','121231','jose@correo.com','Sc edo tachira','docente',1,1); -- cedula

INSERT INTO profesores

VALUES ('V',5,'jose barrera','121231','jose@correo.com','Sc edo tachira','docente',1,1); -- cedula

INSERT INTO profesores

VALUES ('V32123123',1,'jose piedra','121231','jose4@correo.com','Sc edo tachira','docente',1,1); -- nro carnet

INSERT INTO profesores

VALUES ('v1212312323',12,'jose barrera'); -- faltan datos

INSERT INTO profesores

VALUES ('v23545444',5,'jose piedra','121231231231','jose@correo.com','Sc edo tachira','docente',1,1); -- correo

INSERT INTO profesores

VALUES ('v23545444',5,'jose piedra','121231231231','jose @correo.com','Sc edo tachira','docente',1,1); -- correo

INSERT INTO profesores

VALUES ('v23545444',5,'jose piedra','121231231231','jose@ correo.com','Sc edo tachira','docente',1,1); -- correo

INSERT INTO profesores

VALUES ('v23545444',5,'jose piedra','121231231231','jose@correocom','Sc edo tachira','docente',1,1); -- correo

INSERT INTO profesores

VALUES ('v23545444',5,'jose piedra','121231231231','josecorreo.com','Sc edo tachira','docente',1,1); -- correo

SELECT \* FROM PROFESORES;

-- secciones

INSERT INTO secciones (nro\_seccion,asignaturas\_id,profesores\_cedula)

VALUES(1,1,'v12123123');

INSERT INTO secciones (nro\_seccion,asignaturas\_id,profesores\_cedula)

VALUES(1,2,'v23545444');

INSERT INTO secciones (nro\_seccion,asignaturas\_id,profesores\_cedula)

VALUES(2,1,'e12123123');

INSERT INTO secciones (nro\_seccion,asignaturas\_id,profesores\_cedula)

VALUES(2,2,'v12123123');

-- error

INSERT INTO secciones (nro\_seccion,asignaturas\_id,profesores\_cedula)

VALUES(1,1,'v12123123'); -- combinacion

INSERT INTO secciones (nro\_seccion,asignaturas\_id,profesores\_cedula)

VALUES(100,1,'v12123111'); -- nro seccion

INSERT INTO secciones (nro\_seccion,asignaturas\_id,profesores\_cedula)

VALUES(-1,1,'v12123123'); -- nro seccion

INSERT INTO secciones (nro\_seccion,asignaturas\_id,profesores\_cedula)

VALUES(0,1,'v12123123'); -- nro seccion

SELECT \* FROM SECCIONES;

-- bloque\_hora

INSERT INTO bloque\_hora

VALUES ('08:00','10:00','lunes');

INSERT INTO bloque\_hora

VALUES ('02:00','17:00','lunes');

INSERT INTO bloque\_hora

VALUES ('00:50','00:50','lunes');

INSERT INTO bloque\_hora

VALUES ('08:00','15:00','lunes');

-- ERROR

INSERT INTO bloque\_hora

VALUES ('20:00am','10:00am','lunes'); --datos mal, debe ser hora militar

INSERT INTO bloque\_hora

VALUES ('24:50','00:50','lunes'); --hora desde

INSERT INTO bloque\_hora

VALUES ('00:60','00:50','lunes'); -- hora desde

INSERT INTO bloque\_hora

VALUES ('00:50','24:50','lunes'); -- hora hasta

INSERT INTO bloque\_hora

VALUES ('00:50','00:60','lunes'); -- hora hasta

INSERT INTO bloque\_hora

VALUES ('08:00','10:00','algo'); -- dia

INSERT INTO bloque\_hora

VALUES ('08:00','10:00','lunes'); -- repetido

INSERT INTO bloque\_hora

VALUES ('00:50','00:00','lunes'); -- hora hasta menor a hora desde

SELECT \* FROM BLOQUE\_HORA;

-- disponibilidad

INSERT INTO disponibilidad

VALUES ('V12123123','08:00','10:00','lunes');

INSERT INTO disponibilidad

VALUES ('V12123123','02:00','17:00','lunes');

INSERT INTO disponibilidad

VALUES ('V23545444','02:00','17:00','lunes');

-- ERROR

INSERT INTO disponibilidad

VALUES ('V12123123 ','02:00','17:00','lunes'); --PK

INSERT INTO disponibilidad

VALUES ('V23545444','02:00','17:00','martes'); --bloque\_hora

INSERT INTO disponibilidad

VALUES ('V23545444','03:00','17:00','lunes'); --bloque\_hora

INSERT INTO disponibilidad

VALUES ('V12123123','08:00','15:00','lunes'); --hora\_desde y dia repetidos para mismo prof

SELECT \* FROM DISPONIBILIDAD;

-- hor\_clases

-- (cedula, secciones\_id, aula\_id, bloque\_desde bloque\_hasta bloque\_dia, tipo\_clase)

INSERT INTO hor\_clases

VALUES ('v12123123',1, 1,'08:00','10:00','LUNES', 'T');

INSERT INTO hor\_clases

VALUES ('V23545444',2, 2,'08:00','10:00','LUNES', 'l');

INSERT INTO hor\_clases

VALUES ('V23545444',2, 1,'02:00','17:00','LUNES', 't');

-- ERROR

INSERT INTO hor\_clases

VALUES ('v12123123',1, 1,'08:00','10:00','LUNES', 'T'); --aula ocupada

INSERT INTO hor\_clases

VALUES ('V23545444',1, 1,'08:00','10:00','LUNES', 'T'); --prof no corresponde a seccion

INSERT INTO hor\_clases

VALUES ('v12123123',1, 1,'02:00','17:00','LUNES', 'l'); -- profesor ocupado

INSERT INTO hor\_clases

VALUES ('V23545444',2, 2,'08:00','10:00','LUNES', 'D'); --tipo\_clase

INSERT INTO hor\_clases

VALUES ('v23545444',1, 2,'02:00','17:00','martes', 'T'); --bloque\_hora no existe

INSERT INTO hor\_clases

VALUES ('V23545444',2, 1,'08:00','10:00','LUNES', 'l'); -- error seccion

SELECT \* FROM HOR\_CLASES;