

FORMACION PEDAGÓGICA DE LOS DOCENTES

UNIVERSIDAD NACIONAL DE COLOMBIA

BASES DE DATOS

JUAN CARLOS TORRES PARDO

NICOLAS EDUARDO PARDO ARIAS – SEBASTIAN GARNICA QUIROZ – ANDRES FELIPE CASTILLO SOPÓ

npardoa@unal.edu.co – sgarnicaq@unal.edu.co – acastillos@unal.edu.co

PASO 0: SUBDIVISIÓN DEL PROBLEMA

- El problema se abarcará con una sola base de datos, es decir, no se subdividirá el problema. Lo anterior debido a que tanto los docentes con cursos de educación continua, como los docentes con formación enfocada a pedagogía de inclusión social se puede administrar desde un mismo modelo, especificando los temas y enfoques que un programa puede llegar a tener.

“

Unidad de Educación Continua y permanente

Facultad de Ingeniería

Universidad Nacional de Colombia

Comprende el desarrollo de programas y planes de entrenamiento que contribuyen a la actualización y calificación permanente en áreas de ingeniería, dirigida a actualizar, profundizar y desarrollar nuevos conocimientos y competencias en disciplinas académicas para transmitir la experiencia de la Universidad en la formulación de estrategias pedagógicas a las instituciones públicas o privadas de los distintos niveles, y prestarles ayuda. Entre otros ejemplos están las acciones que se realizan para el Ministerio de Educación.

Dentro de esta Unidad las actividades de capacitación se desarrollan por medio de cursos abiertos, cursos cerrados, diplomados, congresos y eventos.

Nuestra metodología puede ser teórica, práctica o una combinación de ambas de educación presencial, semi-presencial o virtual, descritos en las siguientes modalidades:

- **Cursos de Extensión**

Programas educativos en los que se adquieren o actualizan conocimientos e información sobre una temática específica. Pueden ser teóricos, prácticos o una combinación de ambos.

- **Cursos de actualización o de profundización**

Programas académicos de corta o media duración cuyo objetivo es actualizar o profundizar conocimientos e información, producto del trabajo investigativo, docente y de extensión de la Facultad a la que está adscrito el programa respectivo. Se dirigen a estudiantes y profesionales. Pueden ser teóricos, prácticos o una combinación de ambos.

- **Programas de Formación Docente**

Programas académicos dirigidos a la actualización y el mejoramiento profesional de los educadores vinculados al sistema de educación pública. En estos programas se otorgan créditos que se exigen como requisito de capacitación para el ingreso y al ascenso en el Escalafón Docente. Son propuestos para complementariedad pedagógica, investigativa y disciplinaria dirigido a maestras y maestros.

- **Diplomados**

Programas educativos cuyo propósito es profundizar o actualizar los conocimientos o desarrollar competencias y

”

habilidades específicas para el desempeño profesional.

PASOS 1: POSIBLES ENTIDADES

E1: PAIS

E2: CIUDAD

E3: INSTITUCION

E4: TIPO_INSTITUCION

E5: PROGRAMA

E6: TIPO_PROGRAMA

E7: ENFOQUE

E8: DOCENTE

E9: TIPO_DOCENTE

E10: DEPARTAMENTO

PASO 2 Y 3: MATRIZ ENTIDAD-ENTIDAD Y SU CARDINALIDAD

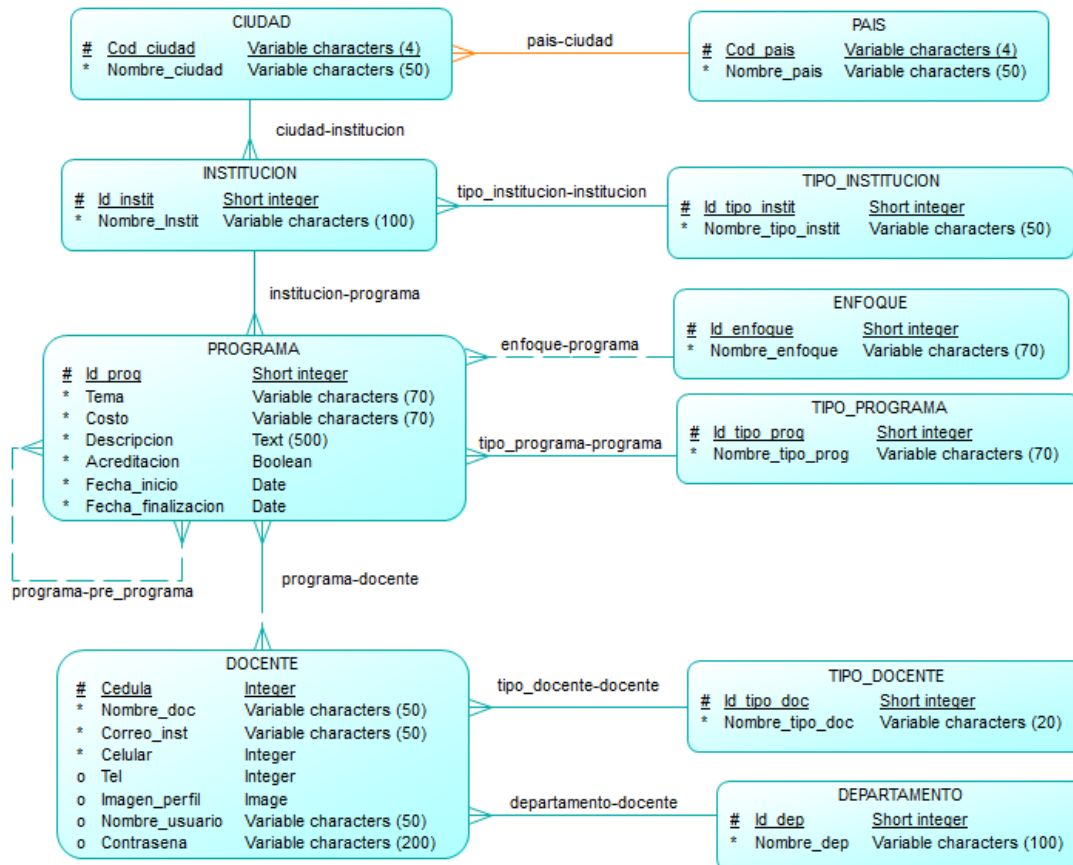
[illegible]

PASO 4: ATRIBUTOS DE LAS ENTIDADES

- PAIS = {Cod_pais, Nombre_pais}
- CIUDAD = {Cod_ciudad, Nombre_ciduad}
- INSTITUCION = { Id_instit, Nombre_Inst}
- TIPO_INSTITUCION = { Id_tipo_instit, Nombre_tipo_instit}
- PROGRAMA = {Id_prog, Tema, Costo, Descripcion, Acreditacion, Fecha_inicio, Fecha_finalización}
- ENFOQUE = {Id_enfoque, Nombre_enfoque}
- TIPO_PROGRAMA = { Id_tipo_prog, Nombre_tipo_prog}
- DOCENTE = {Cedula, Nombre_doc, Correo_Inst, Celular, Tel, Imagen_perfil, Nombre_usuario, Contraseña}
- TIPO_DOCENTE = { Id_tipo_doc, Nombre_tipo_doc}
- DEPARTAMENTO = { Id_dep, Nombre_dep}

PASO 5: MODELO ENTIDAD-RELACION INICIAL

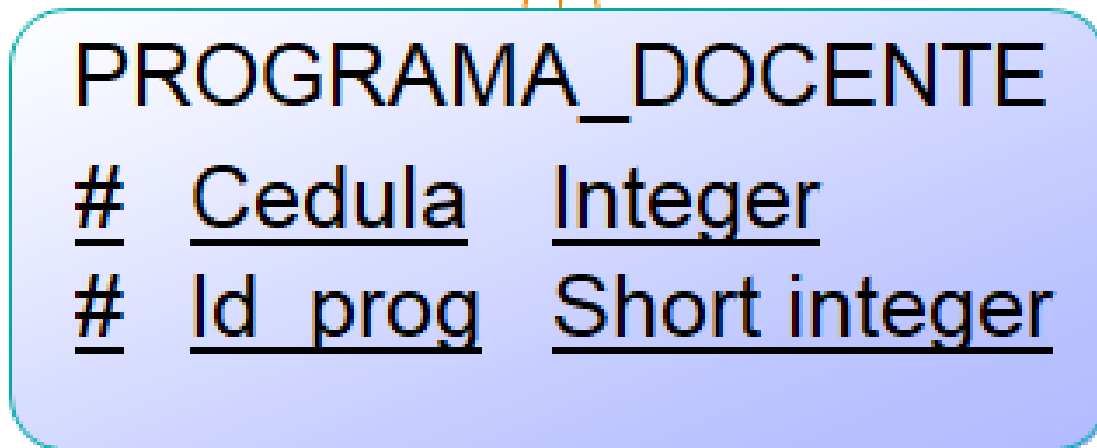
FORMACION DOCENTE EN LA FACULTAD DE INGENIERÍA



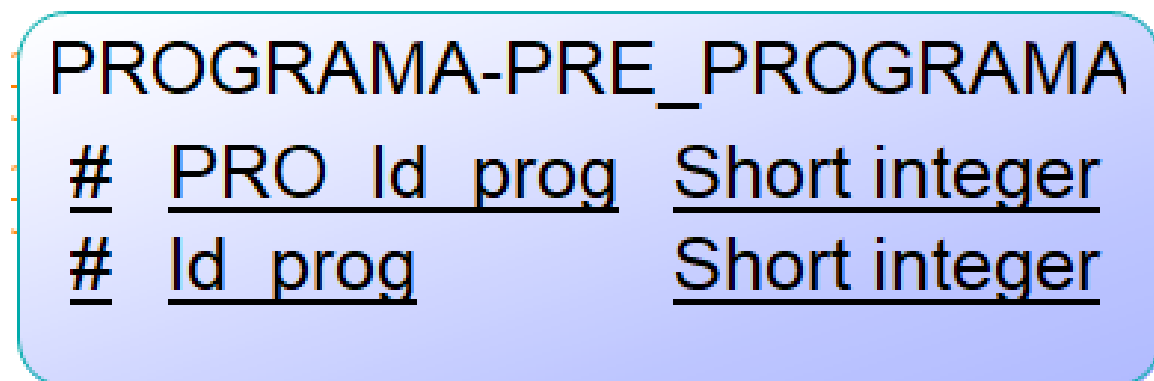
Conceptual Data Model	
Model: Formacion Docente En La Facultad De Ingenieria	
Package:	
Diagram: Formacion Docente En La Facultad De Ingenieria	
Author: Andres Castillo, Sebastian Garnica, Nicolas Pardo	Date: 4/06/2018
Version:	

PASO 6: TRANSFORMAR RELACIONES N-M Y FORMALIZAR ENTIDADES DÉBILES

E11:



E12:

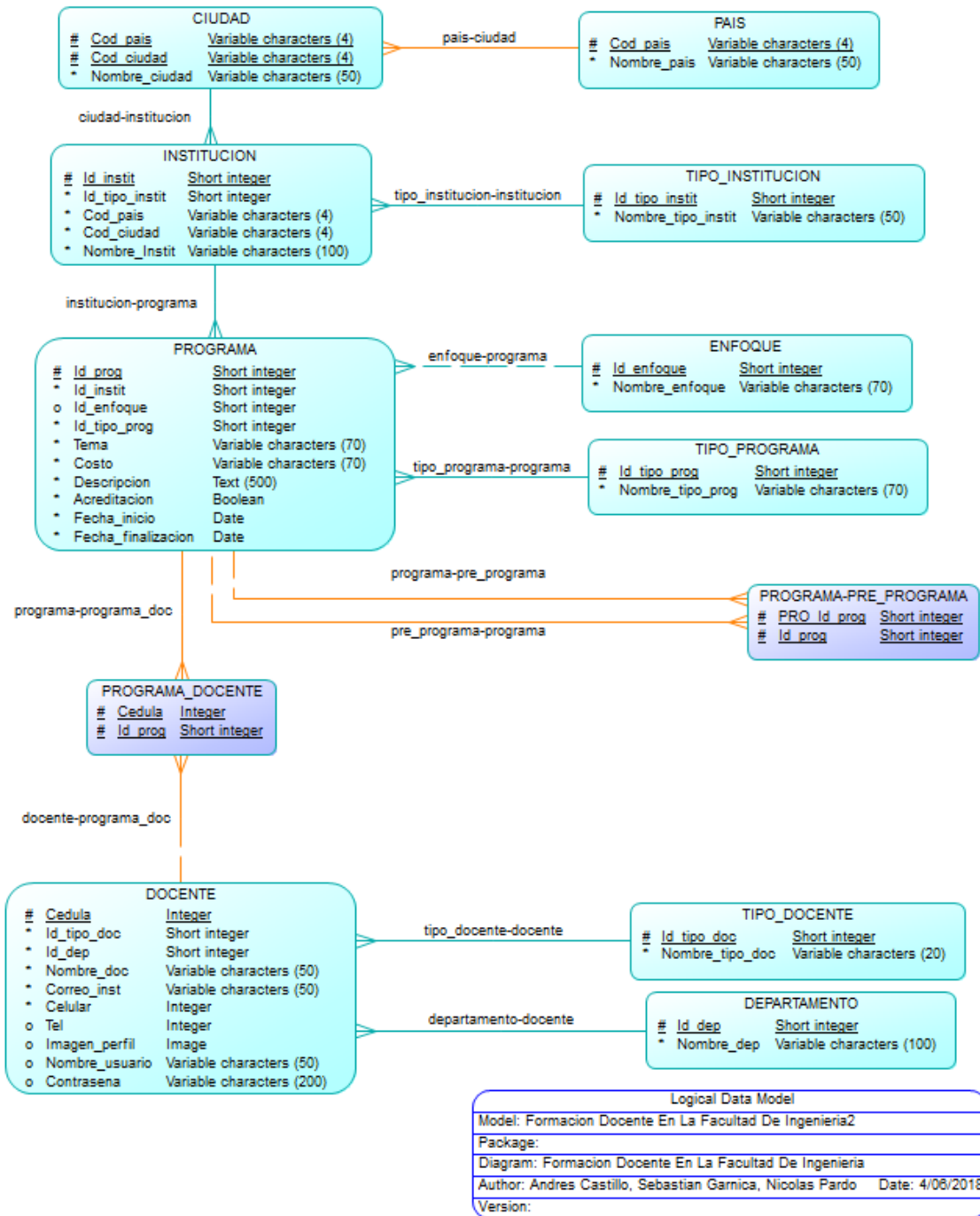


PASO 7: MATRIZ ENTIDAD-ENTIDAD EXTENDIDA

[illegible]

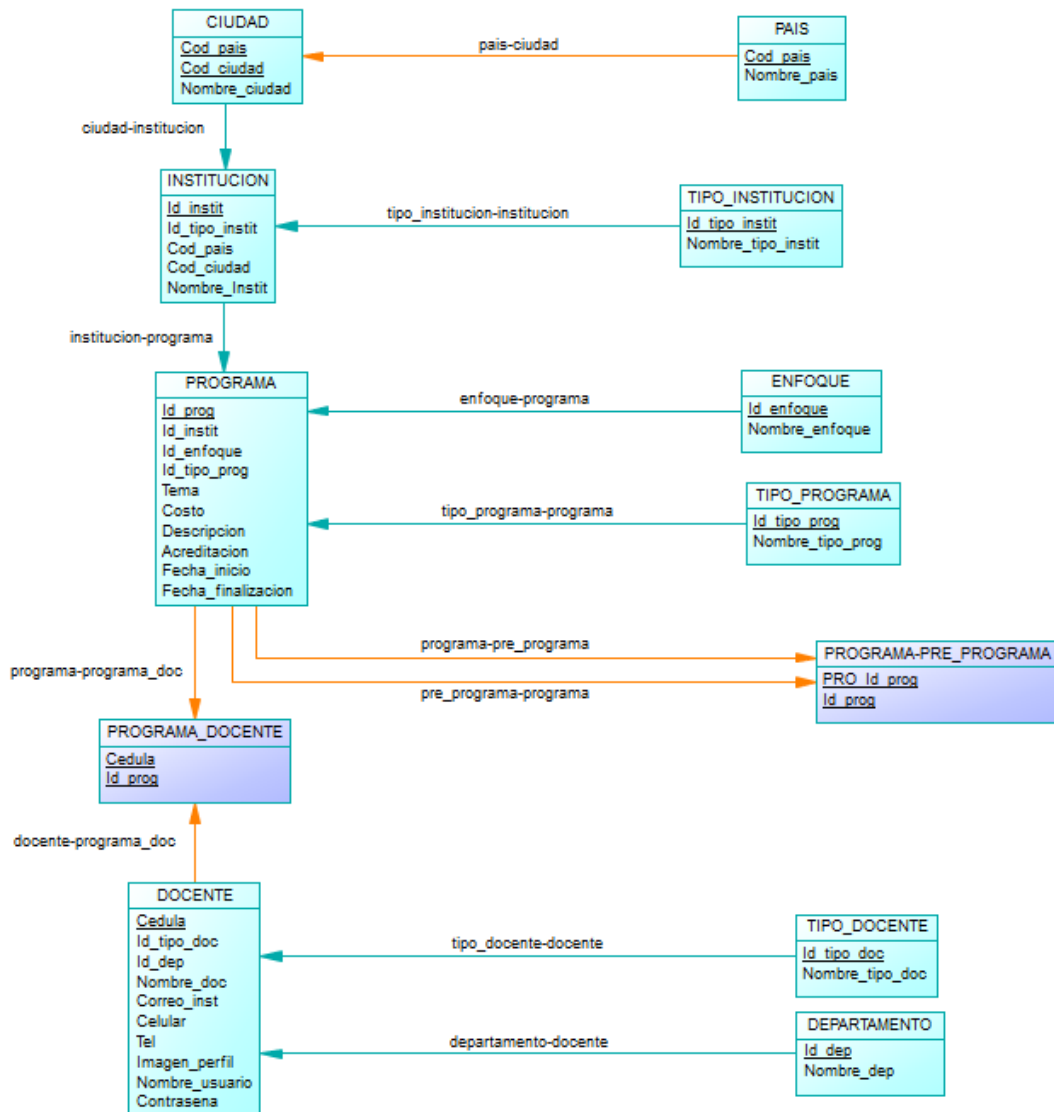
PASO 8: MODELO RELACION-ENTIDAD FINAL

FORMACION DOCENTE EN LA FACULTAD DE INGENIERÍA



PASO 9: MODELO ESTRUCTURA DATO

FORMACION DOCENTE EN LA FACULTAD DE INGENIERÍA



Physical Data Model

Model: Formacion Docente En La Facultad De Ingenieria2

Package:

Diagram: Formacion Docente En La Facultad De Ingenieria

Author: Andres Castillo, Sebastian Garnica, Nicolas Pardo Date: 4/06/2018

Version:

PASO 10: FORMATO TABLA CAMPO**PAIS**

Name	Code	Data Type	Length	Pr	P	F	M	Referencia
Cod_pais	COD_PAIS	Varchar(4)	4		x		x	
Nombre_pais	NOMBRE_PAIS	Varchar(50)	50				x	

CIUDAD

Name	Code	Data Type	Length	Pr	P	F	M	Referencia
Cod_pais	COD_PAIS	Varchar(4)	4		x	x	x	PAIS(Cod_pais)
Cod_ciudad	COD_CIUDAD	Varchar(4)	4		x		x	
Nombre_ciudad	NOMBRE_CIUDAD	Varchar(50)	50				x	

INSTITUCION

Name	Code	Data Type	Length	Pr	P	F	M	Referencia
Id_instit	ID_INSTIT	smallint			x		x	
Id_tipo_instit	ID_TIPO_INSTIT	smallint				x	x	TIPO_INSTITUCION(Id_tipo_instit)
Cod_pais	COD_PAIS	Varchar(4)	4			x	x	PAIS(Cod_pais)
Cod_ciudad	COD_CIUDAD	Varchar(4)	4			x	x	CIUDAD(Cod_ciudad)
Nombre_Instit	NOMBRE_INSTIT	Varchar(100)	100				x	

TIPO_INSTITUCION

Name	Code	Data Type	Length	Pr	P	F	M	Referencia
Id_tipo_instit	ID_TIPO_INSTIT	smallint			x		x	
Nombre_tipo_instit	NOMBRE_TIPO_INSTIT	Varchar(50)					x	

PROGRAMA

Name	Code	Data Type	Length	Pr	P	F	M	Referencia
Id_prog	ID_PROG	smallint			x		x	
Id_instit	ID_INSTIT	smallint				x	x	INSTITUCIÓN(Id_instit)
Id_enfoque	ID_ENFOQUE	smallint				x		ENFOQUE(Id_enfoque)
Id_tipo_prog	ID_TIPO_PROG	smallint				x	x	TIPO_PROGRAMA(Id_tipo_prog)
Tema	TEMA	Varchar(70)	70				x	

Costo	COSTO	Varchar(70)	70				x	
Descripcion	DESCRIPCION	Text					x	
Acreditacion	ACREDITACION	Bool					x	
Fecha_inicio	FECHA_INICIO	Date					x	
Fecha_finalización	FECHA_FINALIZACION	Date					x	

ENFOQUE

Name	Code	Data Type	Length	Pr	P	F	M	Referencia
Id_enfoque	ID_ENFOQUE	smallint			x		x	
Id_instit	ID_INSTIT	smallint					x	

TIPO_PROGRAMA

Name	Code	Data Type	Length	Pr	P	F	M	Referencia
Id_tipo_prog	ID_TIPO_PROG	smallint			x		x	
Nombre_tipo_prog	NOMBRE_TIPO_PROG	Varchar(70)	70				x	

PROGRAMA-PRE_PROGRAMA

Name	Code	Data Type	Length	Pr	P	F	M	Referencia
PRO_Id_prog	PRO_ID_PROG	smallint			x	x	x	PROGRAMA(Id_prog)
Id_prog	ID_PROG	smallint			x	x	x	PROGRAMA(Id_prog)

PROGRAMA_DOCENTE

Name	Code	Data Type	Length	Pr	P	F	M	Referencia
Cedula	CEDULA	int			x	x	x	DOCENTE(Cedula)
Id_prog	ID_PROG	smallint			x	x	x	PROGRAMA(Id_prog)

DOCENTE

Name	Code	Data Type	Length	Pr	P	F	M	Referencia
Cedula	CEDULA	int			x		x	
Id_tipo_doc	ID_TIPO_DOC	smallint				x	x	TIPO_DOCENTE(Id_tipo_doc)
Id_dep	ID_DEP	smallint				x	x	DEPARTAMENTO(Id_dep)

Nombre_doc	NOMBRE_DOC	Varchar(50)	50				x	
Correo_inst	CORREO_INST	Varchar(50)	50				x	
Celular	CELULAR	int					x	
Tel	TEL	int						
Imagen_perfil	IMAGEN_PERFIL	longblob						
Nombre_Usuario	NOMBRE_USUARIO	Varchar(50)	50					
Contrasena	CONTRASENA	Varchar(200)	200					

TIPO_DOCENTE

Name	Code	Data Type	Length	Pr	P	F	M	Referencia
Id_tipo_doc	ID_TIPO_DOC	smallint			x		x	
Id_instit	ID_INSTIT	smallint				x	x	

DEPARTAMENTO

Name	Code	Data Type	Length	Pr	P	F	M	Referencia
Id_dep	ID_DEP	smallint			x		x	
Nombre_dep	NOMBRE_DEP	Varchar(100)	100				x	

PASO 11: GENERAR SCRIPT

```
/*=====*/
```

```
/* DBMS name:    MySQL 5.0           */
```

```
/* Created on:   4/06/2018 7:25:28 p.m.    */
```

```
/*=====*/
```

drop table if exists CIUDAD;

drop table if exists DEPARTAMENTO;

drop table if exists DOCENTE;

drop table if exists ENFOQUE;

drop table if exists INSTITUCION;

drop table if exists PAIS;

drop table if exists PROGRAMA;

drop table if exists PROGRAMA_DOCENTE;

drop table if exists PROGRAMA_PROGRAMA;

drop table if exists TIPO_DOCENTE;

drop table if exists TIPO_INSTITUCION;

drop table if exists TIPO_PROGRAMA;

```
/*=====*/
```

```
/* Table: CIUDAD           */
```

```
/*=====*/
```

create table CIUDAD

```
(  
    COD_PAIS      varchar(4) not null,  
    COD_CIUDAD    varchar(4) not null,  
    NOMBRE_CIUDAD varchar(50) not null,  
    primary key (COD_PAIS, COD_CIUDAD)  
);
```

/*=====*/

/* Table: DEPARTAMENTO */

/*=====*/

create table DEPARTAMENTO

```
(  
    ID_DEP      smallint not null,  
    NOMBRE_DEP  varchar(100) not null,  
    primary key (ID_DEP)  
);
```

/*=====*/

/* Table: DOCENTE */

/*=====*/

create table DOCENTE

```
(  
    CEDULA      int not null,  
    ID_TIPO_DOC smallint not null,  
    ID_DEP      smallint not null,  
    NOMBRE_DOC   varchar(50) not null,  
    CORREO_INST  varchar(50) not null,  
    CELULAR      int not null,  
    TEL          int,  
    IMAGEN_PERFIL longblob,  
    NOMBRE_USUARIO varchar(50),  
    CONTRASENA   varchar(200),  
    primary key (CEDULA)
```

);

/*=====*/

/* Table: ENFOQUE */

/*=====*/

create table ENFOQUE

(

ID_ENFOQUE smallint not null,

NOMBRE_ENFOQUE varchar(70) not null,

primary key (ID_ENFOQUE)

);

/*=====*/

/* Table: INSTITUCION */

/*=====*/

create table INSTITUCION

(

ID_INSTIT smallint not null,

ID_TIPO_INSTIT smallint not null,

COD_PAIS varchar(4) not null,

COD_CIUDAD varchar(4) not null,

NOMBRE_INSTIT varchar(100) not null,

primary key (ID_INSTIT)

);

/*=====*/

/* Table: PAIS */

/*=====*/

create table PAIS

(

COD_PAIS varchar(4) not null,

NOMBRE_PAIS varchar(50) not null,

primary key (COD_PAIS)

);

/*=====*/

/* Table: PROGRAMA */

/*=====*/

create table PROGRAMA

(
 ID_PROG **smallint not null,**
 ID_INSTIT **smallint not null,**
 ID_ENFOQUE **smallint,**
 ID_TIPO_PROG **smallint not null,**
 TEMA **varchar(70) not null,**
 COSTO **varchar(70) not null,**
 DESCRIPCION **text not null,**
 ACREDITACION **bool not null,**
 FECHA_INICIO **date not null,**
 FECHA_FINALIZACION **date not null,**
 primary key (ID_PROG)
);

/*=====*/

/* Table: PROGRAMA_DOCENTE */

/*=====*/

create table PROGRAMA_DOCENTE

(
 CEDULA **int not null,**
 ID_PROG **smallint not null,**
 primary key (CEDULA, ID_PROG)
);

/*=====*/

/* Table: PROGRAMA_PROGRAMA */

/*=====*/

create table PROGRAMA_PROGRAMA

(


```
    PRO_ID_PROG      smallint not null,
    ID_PROG          smallint not null,
    primary key (PRO_ID_PROG, ID_PROG)
);
```

```
/*=====*/
```

```
/* Table: TIPO_DOCENTE */
```

```
/*=====*/
```

```
create table TIPO_DOCENTE
```

```
(
    ID_TIPO_DOC      smallint not null,
    NOMBRE_TIPO_DOC  varchar(20) not null,
    primary key (ID_TIPO_DOC)
);
```

```
/*=====*/
```

```
/* Table: TIPO_INSTITUCION */
```

```
/*=====*/
```

```
create table TIPO_INSTITUCION
```

```
(
    ID_TIPO_INSTIT   smallint not null,
    NOMBRE_TIPO_INSTIT varchar(50) not null,
    primary key (ID_TIPO_INSTIT)
);
```

```
/*=====*/
```

```
/* Table: TIPO_PROGRAMA */
```

```
/*=====*/
```

```
create table TIPO_PROGRAMA
```

```
(
    ID_TIPO_PROG     smallint not null,
    NOMBRE_TIPO_PROG varchar(70) not null,
    primary key (ID_TIPO_PROG)
);
```

**alter table CIUDAD add constraint FK_PAIS_CIUDAD foreign key (COD_PAIS)
references PAIS (COD_PAIS) on delete restrict on update restrict;**

**alter table DOCENTE add constraint FK_DEPARTAMENTO_DOCENTE foreign key (ID_DEP)
references DEPARTAMENTO (ID_DEP) on delete restrict on update restrict;**

**alter table DOCENTE add constraint FK_TIPO_DOCENTE_DOCENTE foreign key (ID_TIPO_DOC)
references TIPO_DOCENTE (ID_TIPO_DOC) on delete restrict on update restrict;**

**alter table INSTITUCION add constraint FK_CIUDAD_INSTITUCION foreign key (COD_PAIS,
COD_CIUDAD)
references CIUDAD (COD_PAIS, COD_CIUDAD) on delete restrict on update restrict;**

**alter table INSTITUCION add constraint FK_TIPO_INSTITUCION_INSTITUCION foreign key
(ID_TIPO_INSTIT)
references TIPO_INSTITUCION (ID_TIPO_INSTIT) on delete restrict on update restrict;**

**alter table PROGRAMA add constraint FK_ENFOQUE_PROGRAMA foreign key (ID_ENFOQUE)
references ENFOQUE (ID_ENFOQUE) on delete restrict on update restrict;**

**alter table PROGRAMA add constraint FK_INSTITUCION_PROGRAMA foreign key (ID_INSTIT)
references INSTITUCION (ID_INSTIT) on delete restrict on update restrict;**

**alter table PROGRAMA add constraint FK_TIPO_PROGRAMA_PROGRAMA foreign key (ID_TIPO_PROG)
references TIPO_PROGRAMA (ID_TIPO_PROG) on delete restrict on update restrict;**

**alter table PROGRAMA_DOCENTE add constraint FK_DOCENTE_PROGRAMA_DOC foreign key (CEDULA)
references DOCENTE (CEDULA) on delete restrict on update restrict;**

**alter table PROGRAMA_DOCENTE add constraint FK_PROGRAMA_DOCENTE2 foreign key (ID_PROG)
references PROGRAMA (ID_PROG) on delete restrict on update restrict;**

**alter table PROGRAMA_PROGRAMA add constraint FK_PROGRAMA_PROGRAMA foreign key
(PRO_ID_PROG)**

references PROGRAMA (ID_PROG) on delete restrict on update restrict;

```
alter table PROGRAMA_PROGRAMA add constraint FK_PROGRAMA_PROGRAMA2 foreign key  
(ID_PROG)
```

references PROGRAMA (ID_PROG) on delete restrict on update restrict;

PASO 12: RUN

El script es ejecutado mediante Sublime Text 3 y visualizado en la aplicación web creada.