

## *Modelado de Base de Datos*

<Titulo del proyecto>

Versión X.X

Programa Nacional de Formación en Informática (PNFI)

Integrantes del equipo:

Cédula de identidad	Apellidos y Nombres	Correo electrónico

Profesor de proyecto: [Nombre y apellido profesor de proyecto]

Docente de la UC: [Docente de la unidad curricular]

Sección: [N° de sección]

## Página de revisiones

Fecha	Versión	Descripción	Autores
XX/XX/XXXX	X.X	Nuevo documento	EGP

## Índice de contenido

1. <u>Identificar las entidades, atributos y claves primarias de cada entidad</u>	4
2. <u>Establecer la cardinalidad entre las entidades</u>	6
3. <u>Realizar el Diagrama Entidad Relación (DER)</u>	6
4. <u>Elaborar el Esquema de Integridad Referencial (EIR) a partir del Diagrama Entidad Relación (DER)</u>	8
5. <u>Elaborar el Diagrama de Entidad Referencial (DIR) a partir del Esquema de Integridad Referencial (EIR)</u>	8
6. <u>Elaborar Diccionario de Datos a partir del Diagrama de Integridad Referencial (DIR) y el Esquema de Integridad Referencial (EIR)</u>	9
7. <u>Implementar el diseño de la base de datos en el Sistema Manejador de Bases de Datos</u>	10
8. <u>Generar los scripts de la Base de Datos (BD, Tablas, Etc)</u>	10
9. <u>Generar los scripts de vistas, procedimientos almacenados, disparadores y funciones</u>	12
10. <u>Plan de mantenimiento de la Base de Datos</u>	14

## 1. Identificar las entidades, atributos y claves primarias de cada entidad

Identifique las entidades, atributos y claves primarias para cada entidad utilizando la tabla indicada.

Tabla	Atributos	Clave Primaria
Usuario	usuarioCedula, usuarioNombre, usuarioApellido, usuarioCorreoElectronico , usuarioClave	ID_USUARIO
Empresa	empresaRazonSocial, empresaRIF, empresaDireccion, empresaTelefono	ID_EMPRESA
Proyecto	proyectoCodigo, proyectoDescripcion, proyectoFecha	ID_PROYECTO

Recomendaciones prácticas:

- A nivel de Base de Datos
  - El charset de las bases de datos debe ser 'UTF-8'.
  - El motor por defecto debe de ser relacional (INNO).
  - Nombre de la base de datos en MAYÚSCULA ej 'MIBASEDEDATOS'.
- A nivel de tablas
  - El nombre de las tablas debe ser en mayúsculas o CamelCase
  - El nombre de las tablas debe ser en singular
  - El nombre de las tablas debe ser descriptivo, no importa que tan largos sean siempre y cuando sean soportados por la base de datos
  - Si la tabla tiene más de 2 palabras estas se deben poner juntas o con un guión bajo, nunca se debe de usar espacios, ej 'APELLIDO\_PATERNO' o en CamelCase 'APaterno'
  - Si la tabla es una tabla muchos a muchos (multivaluada o pivote), se deben utilizar los nombres de las tablas que generan la relación, deben ir con

guiones bajos, ej 'VENDEDOR\_CLIENTE', 'VENDEDOR\_CLIENTE\_PRODUCTO' otro ejemplo 'CONTACTO\_TELEFONO\_TABLA1\_TABLA2\_TABLAN', el guion ayuda a identificar las relaciones

- Evite el uso de tablas temporales, mejor utilice alguna estructura de datos del lenguaje de programación que esté usando
- A nivel de campos
  - Todas las tablas deben de tener un Primary Key (PK) y debe de ser el primer campo de la tabla y debe de ser único e irrepetible
  - Además del PK se sugiere que todas las tablas tengan los siguientes campos: Estatus, Fecha Registro, Fecha Ultima Modificación (FECHA\_MOD), Fecha Eliminación, Usuario Creación, Usuario Modificación y Usuario Eliminación
  - El nombre de los campos/columnas deben ser en singular
  - PK o llave primaria de todas las tablas debe de ser un entero (INT o BIGINT), es único, autoincremental e indexado
  - El nombre del PK empieza con 'ID' en mayúsculas guión bajo nombre de la tabla con mayúsculas, ej 'ID\_NOMBRE\_DE\_LA\_TABLA', 'ID\_APELLIDO\_PATERNO', 'ID\_CONTACTO\_TELEFONO\_TABLA1\_TABLA2'
  - Las llaves foráneas (foreign keys) (FK), deben ser un entero (INT o BIGINT), indexadas, y con la misma nomenclatura que un PK pero agregando 'FK\_' al inicio, ej 'FK\_ID\_NOMBRE\_DE\_LA\_TABLA', 'FK\_ID\_APELLIDO\_PATERNO', 'FK\_ID\_CONTACTO\_TELEFONO\_TABLA1\_TABLA2'
  - El nombre de los campos debe de ser como métodos en JAVA(CamelCase), empiezan en minúsculas, no tienen espacios o guiones bajos, son descriptivos, y las siguientes palabras empiezan con mayúscula, ej 'holaMundo', 'apellidoPaterno', sin embargo también pueden ir en mayúscula usando un guión bajo como espacio, la única condición es que todos sean homogéneos
  - Evita usar el 'NULL' es decir campos en nulo, todos los campos deben de tener una inicialización, esta puede ser vacía y en caso de que tenga otra, debe de estar comentada en la tabla
  - Todos los campos de ESTATUS deben de estar comentados con valores, tratar de usar strings de por lo menos 10 caracteres e indexados

- En ocasiones es muy bueno indexar algún campo de fecha

## 2. Establecer la cardinalidad entre las entidades

Indique la cardinalidad en las relaciones entre las tablas, utilizando “1” para “uno”, y “M,N” para “muchos”.

Usuario	Empresa
1 (a)	1 (a)
M (b)	1 (b)
<b>M</b>	<b>1</b>

De forma descriptiva la relación de cardinalidad se lee así: “En un momento del sistema, un usuario está asociado a una empresa (a); luego, una empresa puede estar asociada a uno o más usuarios (b)”; luego se toma el valor más alto de cada columna y se define la cardinalidad, en este caso es una relación de “uno a muchos” entre empresa y usuarios.


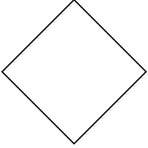

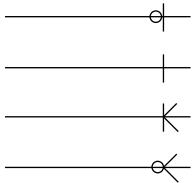
Usuario	Proyecto
1 (a)	N (a)
M (b)	1 (b)
<b>M</b>	<b>N</b>

Para el caso en el que la relación de cardinalidad sea “Muchos a muchos”, se requiere generar una tabla pivote o multivaluada para registrar estas relaciones.

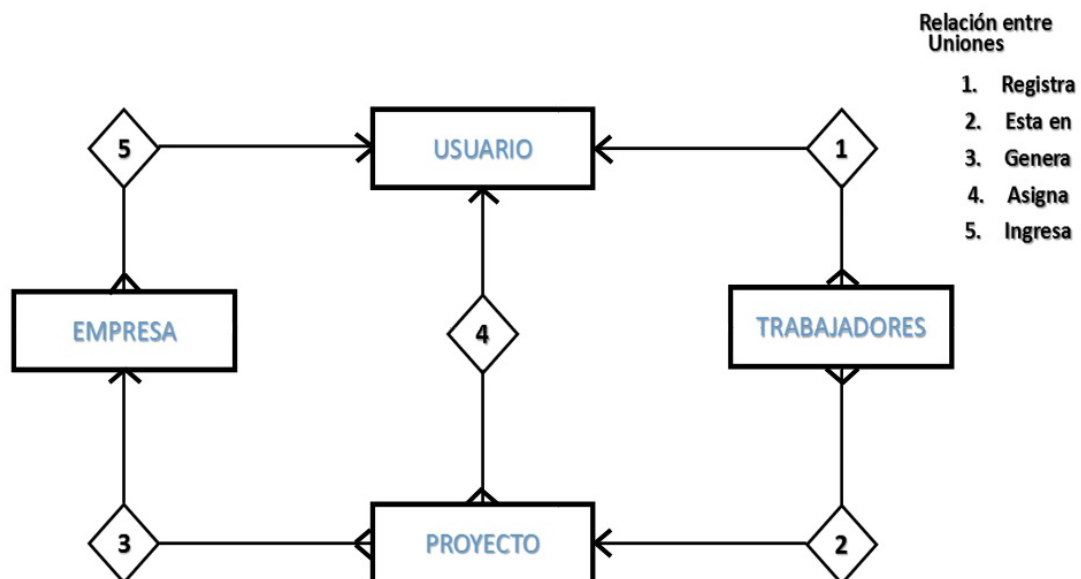
## 3. Realizar el Diagrama Entidad Relación (DER)

En esta sección se debe mostrar las relaciones y cardinalidad entre las entidades definidas en los pasos anteriores. Los estereotipos a utilizar son los siguientes:

Estereotipo	Descripción
Rectángulo	Se utiliza para representar a la entidad. Debe llevar

	dentro de él el nombre de la tabla definido en los pasos anteriores
<p>Rombo</p> 	Se utiliza para representar a la relación entre las entidades. Internamente debe llevar una letra en minúscula. Esta letra debe encontrarse en una leyenda describiendo la relación. La relación debe expresarse claramente y de preferencia en una sola palabra.
<p>Línea</p> 	Se utiliza para conectar los estereotipos de entidades y relaciones. Debe ser una línea sólida
<p>Marcas de cardinalidad</p> 	<p>Ninguno o uno</p> <p>Exactamente uno</p> <p>Uno o más</p> <p>Ninguno o más</p>

Ejemplo:



4. Elaborar el Esquema de Integridad Referencial (EIR) a partir del Diagrama

## Entidad Relación (DER)

En esta sección usted debe definir los campos definitivos de cada tabla, haciendo referencia incluso a cuáles de ellos son llaves primarias y foráneas. Recuerde que en las relaciones 1:N (uno a muchos), la llave foránea debe indicarse en la tabla cuya cardinalidad sea N, y este campo debe definirse exactamente igual que como se define en la tabla en la cual es llave primaria. Ejemplo:

Usuario	Empresa
<b>ID_USUARIO (pk)</b> UsuarioCedula UsuarioNombre UsuarioApellido UsuarioTipo UsuarioAlias UsuarioCorreo UsuarioClave <b>ID_EMPRESA (pk)</b>	<b>ID_EMPRESA (pk)</b> EmpresaRazonSocial EmpresaRIF
Proyecto	Trabajadores
<b>ID_PROYECTO (pk)</b> ProyectoCodigo ProyectoNombre ProyectoTipo ProyectoEstatus ProyectoDireccion ProyectoEstado ProyectoMunicipio ProyectoParroquia ProyectoCiudad <b>IdEmpresa (fk)</b>	<b>ID_TRABAJADOR (pk)</b> TrabajadorCedula TrabajadorNombre TrabajadorApellido TrabajadorEstatus TrabajadorNivelAcademico <b>IdProyecto (fk)</b>

## 5. Elaborar el Diagrama de Integridad Referencial (DIR) a partir del Esquema de Integridad Referencial (EIR)

Mediante el uso de una herramienta de diseño y modelado de bases de datos, genere el Diagrama de Integridad Referencial. Se recomiendan las herramientas: SQL Power Architect Community Edition, Dbeaver, Visual Paradigm.



Si no domina ninguna de estas herramientas, se recomienda al menos realizarlo en alguna aplicación para presentaciones utilizando las formas, pero se invita a invertir tiempo en dominar las herramientas ya que serán útiles en todo el proceso de desarrollo.

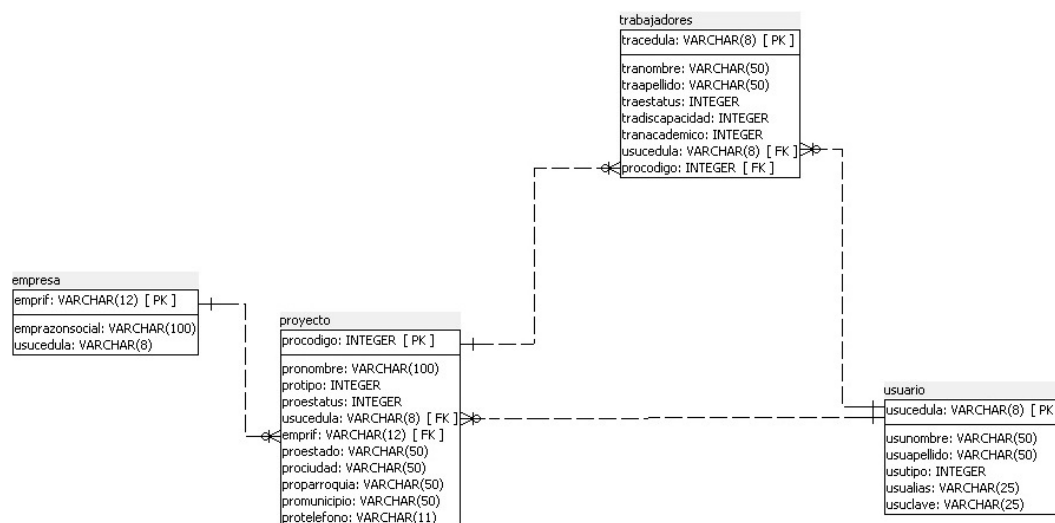
Disponibles en:

[http://www.bestofbi.com/page/architect\\_download\\_os](http://www.bestofbi.com/page/architect_download_os)

<https://dbeaver.io/download/>

<https://www.visual-paradigm.com/download/community.jsp>

Ejemplo:



## 6. Elaborar Diccionario de Datos a partir del Diagrama de Integridad Referencial (DIR) y el Esquema de Integridad Referencial (EIR)

Utilizando la estructura mostrada a continuación, defina los detalles de cada tabla individualmente (una por cada tabla):

Nombre de la tabla (lógico)	Usuario
Nombre de la tabla (físico)	Usuario
Entidades relacionadas	Empresa, Proyecto, Trabajador

Finalidad de la entidad			Registrar la información de los usuarios del sistema de información					
Atributos								
Nombre Columna	Tipo de dato	Tamaño	P K	FK	Único	NotNull	Índice	Descripción
ID_USUARIO	Int		X			X	X	Identificador de la tabla
UsuarioCedula	Varchar	8					X	Cédula del usuario
UsuarioNombre	Varchar	50						Nombre del usuario
...	...	...	...	...	...	...	...	...

## 7. Implementar el diseño de la base de datos en el Sistema Manejador de Bases de Datos

Explique de forma precisa el proceso de implementación de la base de datos en el entorno de desarrollo o producción, indicando también los detalles técnicos del servidor: Sistema Operativo, Sistema de Gestión de Base de Datos utilizado, esquemas propios implementados, etc. De igual forma, explique en esta sección la configuración utilizada, específicamente en aspectos de concurrencia, medidas de seguridad y control de acceso a los datos.

## 8. Generar los scripts de la Base de Datos (BD, Tablas, Etc)

En esta sección debe incluir el script de generación de la base de datos.

Ejemplo:

```
CREATE TABLE public.usuario (
  usucedula VARCHAR(8) NOT NULL,
  usunombre VARCHAR(50),
  usuapellido VARCHAR(50),
  usutipo INTEGER,
  usualias VARCHAR(25),
  usuclave VARCHAR(25),
  CONSTRAINT usucedula PRIMARY KEY (usucedula)
);
```

```
COMMENT ON COLUMN public.usuario.usucedula IS 'Cedula de identidad  
del usuario';  
COMMENT ON COLUMN public.usuario.usunombre IS 'Nombre del usuario';  
COMMENT ON COLUMN public.usuario.usuapellido IS 'Apellido del  
usuario';  
COMMENT ON COLUMN public.usuario.usutipo IS 'Tipo de usuario';
```

```
CREATE TABLE public.empresa (  
    emprif VARCHAR(12) NOT NULL,  
    emprazonsocial VARCHAR(100),  
    usucedula VARCHAR(8) NOT NULL,  
    CONSTRAINT emprif PRIMARY KEY (emprif)  
);  
COMMENT ON COLUMN public.empresa.emprif IS 'Rif de la empresa';  
COMMENT ON COLUMN public.empresa.usucedula IS 'cedula del usuario';
```

```
CREATE TABLE public.proyecto (  
    procodigo INTEGER NOT NULL,  
    pronombre VARCHAR(100),  
    protipo INTEGER,  
    proestatus INTEGER,  
    usucedula VARCHAR(8) NOT NULL,  
    emprif VARCHAR(12) NOT NULL,  
    proestado VARCHAR(50),  
    prociudad VARCHAR(50),  
    proparroquia VARCHAR(50),  
    promunicipio VARCHAR(50),  
    protelefono VARCHAR(11),  
    CONSTRAINT procodigo PRIMARY KEY (procodigo)  
);
```

```
CREATE TABLE public.trabajadores (  
    tracedula VARCHAR(8) NOT NULL,  
    tranombre VARCHAR(50),  
    traapellido VARCHAR(50),  
    traestatus INTEGER,  
    tradiscapacidad INTEGER,  
    tranacademico INTEGER,  
    usucedula VARCHAR(8) NOT NULL,  
    procodigo INTEGER NOT NULL,  
    CONSTRAINT tracedula PRIMARY KEY (tracedula)  
);
```

```
ALTER TABLE public.proyecto ADD CONSTRAINT usucedula  
FOREIGN KEY (usucedula)  
REFERENCES public.usuario (usucedula)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
NOT DEFERRABLE;
```

```
ALTER TABLE public.trabajadores ADD CONSTRAINT usucedula
FOREIGN KEY (usucedula)
REFERENCES public.usuario (usucedula)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

```
ALTER TABLE public.proyecto ADD CONSTRAINT emprif
FOREIGN KEY (emprif)
REFERENCES public.empresa (emprif)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

```
ALTER TABLE public.trabajadores ADD CONSTRAINT procodigo
FOREIGN KEY (procodigo)
REFERENCES public.proyecto (procodigo)
ON DELETE NO ACTION
ON UPDATE NO ACTION
NOT DEFERRABLE;
```

## 9. Generar los scripts de vistas, procedimientos almacenados, disparadores y funciones

En esta sección debe incluir los scripts desarrollados para las consultas vistas, procedimientos almacenados, disparadores y funciones, indicando los aspectos propuestos para cada tipo. Considere lo siguiente:

Vista: Es una alternativa para mostrar datos de varias tablas. Una vista es una tabla virtual que almacena una consulta, y los datos accedidos a través de ella no están almacenados en la base de datos como un objeto.

Procedimiento almacenado: Es un programa almacenado físicamente en la base de datos. Este programa o rutina es ejecutado en el propio motor de base de datos, el cual usualmente corre en un servidor aparte. Opera directamente sobre los datos y retorna un resultado de regreso al usuario, garantizando no generar sobrecarga resultante de comunicar grandes volúmenes de datos salientes y entrantes. A menos que se implemente, los procedimientos almacenados no devuelven resultados.

**Funciones:** Son estructuras que aglutinan una serie de instrucciones con el fin de evitar la repetición de código. Esto como consecuencia, mejora la legibilidad y eficiencia del código, y de igual forma el control de errores. Las funciones normalmente devuelven resultados.

**Disparadores (Triggers):** Es una acción definida en una tabla de la base de datos que se ejecuta automáticamente gracias a una función creada para tal fin. Esta acción se activará según la definición de las reglas y el evento deseado.

Estructura para cada tipo de declaración:

#### Vistas

Nombre de la vista	
Tabla(s) relacionada(s)	
Sentencia	
Resultado	
Descripción	

#### Procedimientos almacenados

Nombre del procedimiento almacenado	
Tabla(s) relacionada(s)	
Sentencia	
Resultado	
Descripción	

#### Funciones

Nombre de la función	
Tabla(s) relacionada(s)	

Sentencia	
Resultado	
Descripción	

Disparadores

Nombre del disparador	
Tabla(s) relacionada(s)	
Evento	
Sentencia	
Descripción	

## 10. Plan de mantenimiento de la Base de Datos

En esta sección explique de forma detallada los mecanismos de mantenimiento de la base de datos que se implementan, incluyendo copias de seguridad, vacuum de tablas, mecanismos para asegurar la persistencia, restauración en casos de emergencia, protocolo de actuación en caso de desastres, etc.