

**UNIVERSIDAD AUTÓNOMA DE NUEVO LEON**

**FACULTAD DE CIENCIAS FISICO-MATEMÁTICAS**

**LIC. CIENCIAS COMPUTACIONALES**

**Base de Datos**

**PIA**

**MAESTRO:**

EDNA MARIA MEDINA MORÓN

**ALUMNOS:**

**DE LA FUENTE AGUILAR DESTINY YARELI** 1855184

**RODIGUEZ PUENTE JANYAN ARIDAÍ** 1941544

**SAUCEDO ALEMAN IAN MAURICIO** 1868954

**HERNANDEZ VILLASANA EDUARDO ALAN** 1941416

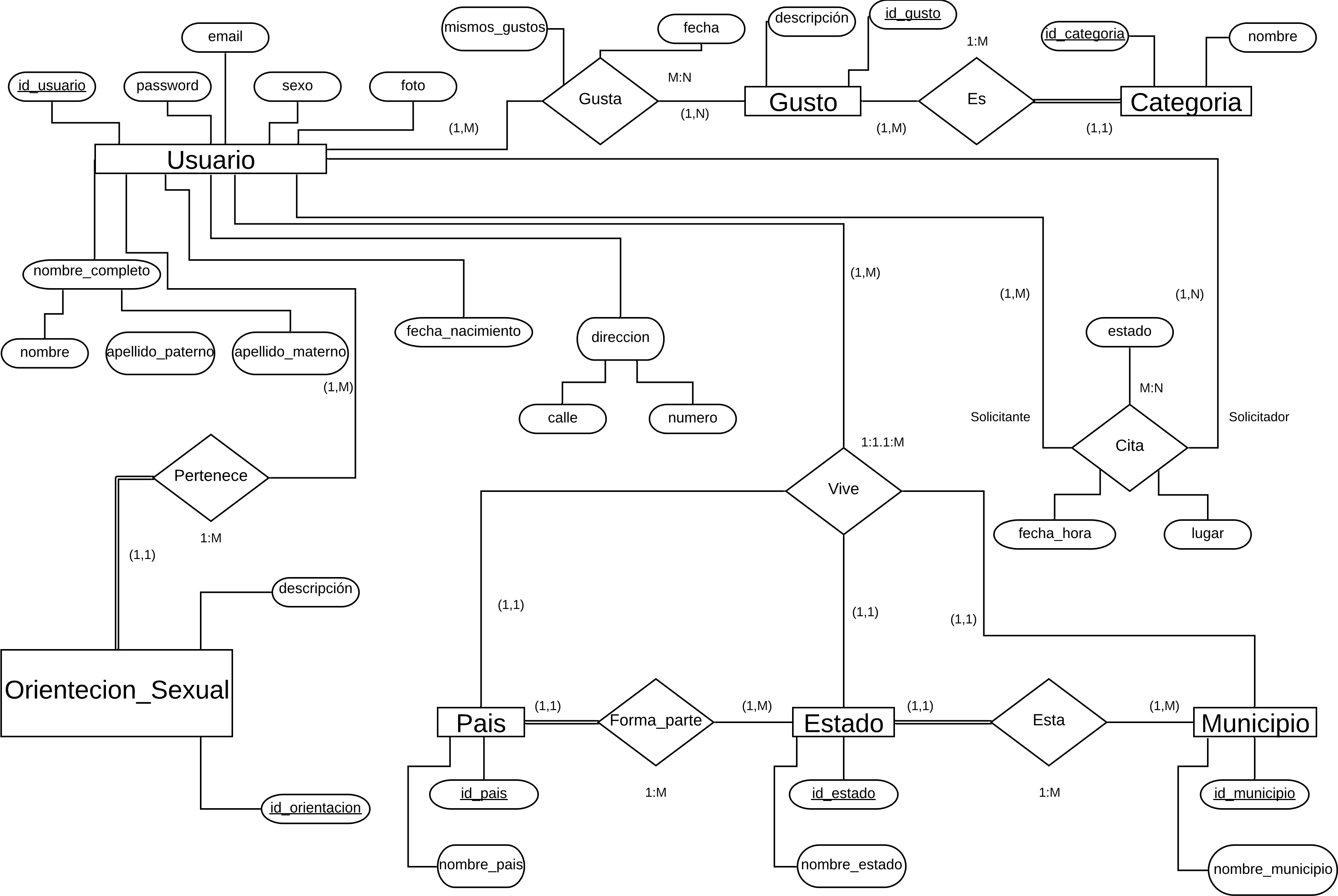
**Dado el siguiente proyecto realizaremos crearemos nuestra base de datos:**

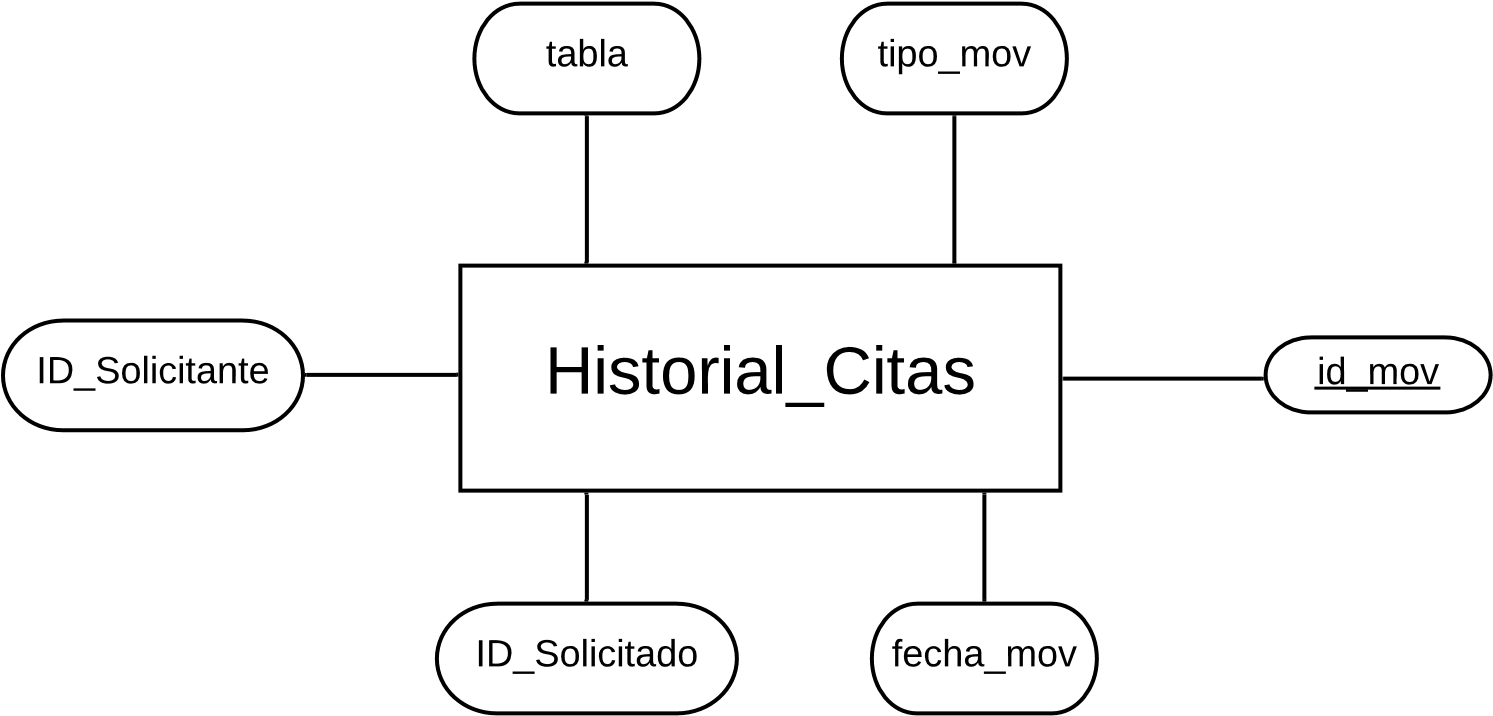
Se va a crear una página web de contactos en el dominio de BuscoPareja.net.

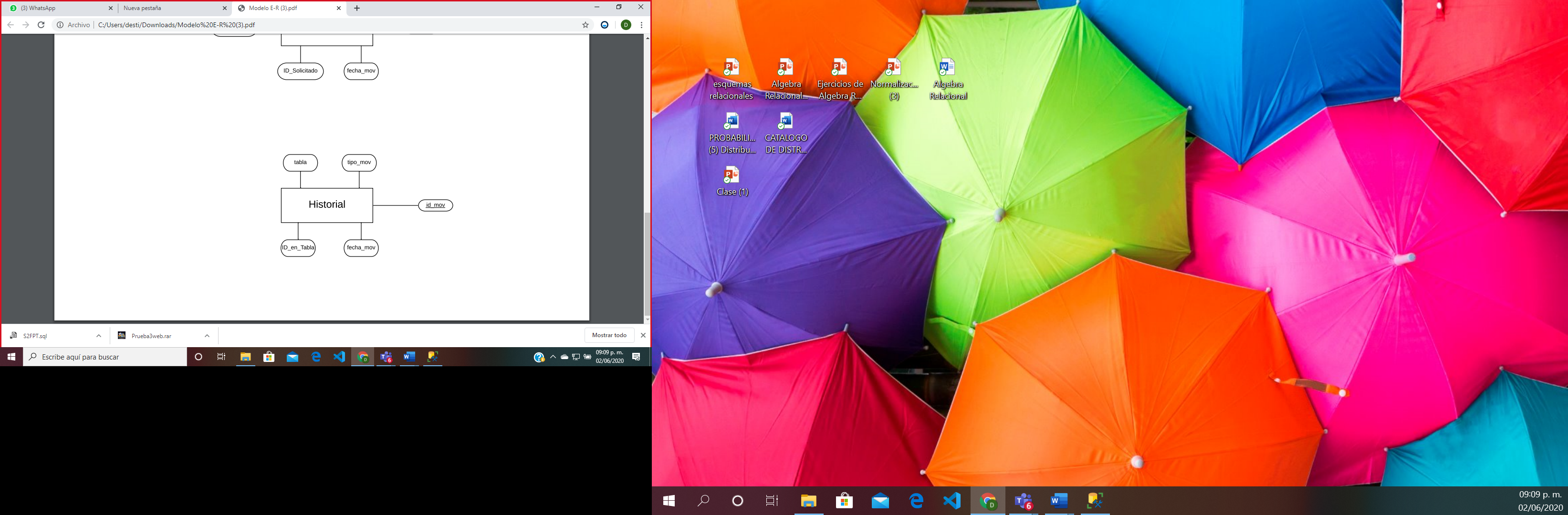
Cuando un usuario se registra en el sistema, se almacenan sus daros personales, concretamente su email, su nombre, dirección, ciudad, país, su sexo y orientación sexual, su foto y una contraseña que utilizara junto con su email para acceder al sistema. El usuario a continuación rellena una lista de preferencias o gustos. De cada gusto o preferencia se almacenará (Deporte, Música, Evento Social), la fecha de la última vez que hizo una actividad de ese estilo y si le gustaría o no que su pareja tuviera la misma preferencia. A partir de esta información se organizan citas entre los contactos en distintas ubicaciones. Se desea registrar las citas entre los contactos almacenando quién se cita con quién, en qué lugar y a qué hora, y si la cita fracasó.

**Definición técnica de la base de datos**

* Diagrama entidad – relación







* Esquema relacional de la base de datos

Esquemas relacionales originales provenientes del DER.

**Entidades**

usuario = (id\_usuario,email, password, sexo, foto, nombre, apellido\_paterno, apellido\_materno, id\_pais, id\_estado, id\_municipio, calle, numero, fecha\_nacimiento, id\_orientacion)

gusto = (id\_gusto, id\_categoria, descripcion)

categoría = (id\_categoria, nombre)

orientacion\_sexual = (id\_orientacion, descripcion)

pais = (id\_pais, nombre\_pais)

estado = (id\_estado, id\_pais, nombre\_estado)

municipio = (id\_municipio, id\_estado, nombre\_municipio)

**Relaciones**

gusta = (email, id\_gusto, fecha, mismos\_gustos)

cita = (email\_solicitante, email\_solicitado, fecha\_hora, lugar, estado)

Esquema relacional y el resultado obtenido en cada forma normal (1NF, 2NF, 3NF).

Teniendo los esquemas relacionales resultantes del apartado anterior:

**usuario = (id\_usuario, email, password, sexo, foto, nombre, apellido\_paterno, apellido\_materno, id\_pais, id\_estado, id\_municipio, calle, numero, fecha\_nacimiento, id\_orientacion)**

**gusto = (id\_gusto, id\_categoria, descripcion)**

**categoría = (id\_categoria, nombre)**

**orientacion\_sexual = (id\_orientacion, descripcion)**

**pais = (id\_pais, nombre\_pais)**

**estado = (id\_estado, id\_pais, nombre\_estado)**

**municipio = (id\_municipio, id\_estado, nombre\_municipio)**

**gusta = (id\_usuario, id\_gusto, fecha, mismos\_gustos)**

**cita = (id\_usuario\_solicitante,id\_usuario\_solicitado, fecha\_hora, lugar,estado)**

Podemos asegurar que no es necesaria la normalización en ninguno de dichos esquemas debido a que ya han llegado al máximo estado de normalización según lo estipulado en la Primera, Segunda y Tercera Forma Normal. La Primera Forma dice que todos los atributos deben de ser atómicos/indivisibles, y en los esquemas no se presentan atributos que sean compuestos o multivariados, al pasar el esquema entidad-relación a esquemas relacionales se elimina este problema ya que hay una regla que nos indica que hacer con estos tipos de atributos, que bien se descomponen en la misma tabla o crean una tabla por si mismos y dejan un valor representativo en la relación original para que puedan estar relacionadas por cierto dato. De igual manera, los esquemas cumplen la Segunda Forma debido a que cada uno de los datos presentes en cada uno de los esquemas dependen directamente de la clave primaria. Finalmente, la Tercera Forma nos dice que no deben de existir dependencias transitivas y en los esquemas ninguno de los atributos que no sean la clave primaria completa dependen de otro que tampoco sea clave. En caso de que cualquiera de los esquemas no hubiera cumplido con las formas de normalización hubiera sido necesario añadir más esquemas, y hubiéramos podido concluir que la base de datos diseñada en el esquema entidad-relación no se encuentra en su mejor diseño y todavía puede ser reducida o dividida aún más.

* Especificación técnica de las relaciones

**usuario**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Tipo de dato | Llave primaria | Dominio | Índice |
| id\_usuaio | int |  |  |  |
| email | varchar(30) |  |  |  |
| password | varchar(10) |  |  |  |
| sexo | char(1) |  | M o F |  |
| foto | image |  |  |  |
| nombre | varchar(100) |  |  |  |
| apellido\_paterno | varchar(200) |  |  |  |
| apellido\_materno | varchar(200) |  |  |  |
| id\_pais | tinyint |  |  |  |
| id\_estado | int |  |  |  |
| id\_municipio | int |  |  |  |
| calle | varchar(200) |  |  |  |
| numero | smallint |  |  |  |
| fecha\_nacimiento | date |  |  |  |
| id\_orientacion | tinyint |  |  |  |

**gusto**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Tipo de dato | Llave primaria | Dominio | Índice |
| id\_gusto | int |  |  |  |
| id\_categoria | tinyint |  |  |  |
| descripcion | varchar(20) |  |  |  |

**categoría**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Tipo de dato | Llave primaria | Dominio | Índice |
| id\_categoria | tinyint |  |  |  |
| nombre | varchar(20) |  | Evento Social, Deporte o música |  |

**orientacion\_sexual**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Tipo de dato | Llave primaria | Dominio | Índice |
| id\_orientacion | tinyint |  |  |  |
| descripcion | varchar(20) |  | Heterosexual,  Homosexual,  Bisexual,  Asexual,  Pansexual, |  |

**pais**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Tipo de dato | Llave primaria | Dominio | Índice |
| id\_pais | tinyint |  |  |  |
| nombre\_pais | varchar(50) |  |  |  |

**estado**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Tipo de dato | Llave primaria | Dominio | Índice |
| id\_estado | Int IDENTITY |  |  |  |
| id\_pais | tinyint |  |  |  |
| nombre\_estado | varchar(50) |  |  |  |

**municipio**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Tipo de dato | Llave primaria | Dominio | Índice |
| id\_municipio | Int IDENTITY |  |  |  |
| id\_estado | Int |  |  |  |
| id\_pais | tinyint |  |  |  |
| nombre\_municipio | varchar(50) |  |  |  |

**gusta**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Tipo de dato | Llave primaria | Dominio | Índice |
| id\_usuario | int |  |  |  |
| id\_gusto | int |  |  |  |
| fecha | date |  |  |  |
| mismos\_gustos | bit |  | 1=éxito  0=fracaso |  |

**cita**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atributo | Tipo de dato | Llave primaria | Dominio | Índice |
| id\_usuario\_solicitante | int |  |  |  |
| id\_usuario\_solicitado | int |  |  |  |
| fecha\_hora | datetime |  |  |  |
| lugar | varchar(50) |  |  |  |
| estado | bit |  | 1=éxito  0=fracaso |  |

* Definición de integridad y método de implementación.

**Usuario**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Llave foránea | Not null | Unique | Ckeck | Enunciado de validación | Mecanismo de implementación |
| id\_usuaio |  |  |  |  |  |  |
| email |  |  |  |  |  |  |
| password |  |  |  |  |  |  |
| sexo |  |  |  |  |  |  |
| foto |  |  |  |  |  |  |
| nombre |  |  |  |  |  |  |
| apellido\_paterno |  |  |  |  |  |  |
| apellido\_materno |  |  |  |  |  |  |
| id\_pais |  |  |  |  |  | id\_pais int FK |
| id\_estado |  |  |  |  |  | id\_estado int FK |
| id\_municipio |  |  |  |  |  | id\_municipio int FK |
| calle |  |  |  |  | Debe de ser no nulo. | calle varchar(50) not null |
| numero |  |  |  |  | Debe de ser no nulo. | numero int not null |
| fecha\_nacimiento |  |  |  |  | Debe de ser no nulo. | fecha\_nacimiento date not null |
| id\_orientacion |  |  |  |  |  | id\_orientacion tinyint FK |

**gusto**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Llave foránea | Not null | Unique | Check | Enunciado de validación | Mecanismo de implementación |
| id\_gusto |  |  |  |  |  |  |
| id\_categoria |  |  |  |  | Es la llave foranea | id\_categoria tinyint FK |
| descripcion |  |  |  |  |  |  |

**categoría**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Llave foránea | Not null | Unique | Ckeck | Enunciado de validación | Mecanismo de implementación |
| id\_categoria |  |  |  |  |  |  |
| nombre |  |  |  |  | El nombre puede ser Evento Social, Deporte o Musica. | check(nombre in ( ‘Evento Social’,  ’Deporte’,  ‘Musica’)) |

**orientacion\_sexual**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Llave foranea | Not null | Unique | Check | Enunciado de validación | Mecanismo de implementación |
| id\_orientacion |  |  |  |  |  |  |
| descripcion |  |  |  |  | La descripcion puede ser Heterosexual,  Homosexual,  Bisexual,  Asexual,  Pansexual | Check(descripción in (‘Heterosexual’,  ‘Homosexual’,  ‘Bisexual’,  ‘Asexual’,  ‘Pansexual’)) |

**pais**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Atributo | Llave foranea | Not null | Unique | Enunciado de validación | Mecanismo de implementación |
| id\_pais |  |  |  |  |  |
| nombre\_pais |  |  |  |  |  |

**estado**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Llave foránea | Not null | Unique | check | Enunciado de validación | Mecanismo de implementación |
| id\_estado |  |  |  |  |  |  |
| id\_pais |  |  |  |  | Es la llave foránea | foreign key (id\_pais) |
| nombre\_estado |  |  |  |  |  |  |

**municipio**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Llave foránea | Not null | Unique | Check | Enunciado de validación | Mecanismo de implementación |
| id\_municipio |  |  |  |  |  |  |
| id\_estado |  |  |  |  | Es llave foránea | foreign key (id\_estado) |
| id\_pais |  |  |  |  | Es llave foránea | foreign key  (id\_pais) |
| nombre\_municipio |  |  |  |  |  |  |

**gusta**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Llave foránea | Not null | Unique | Check | Enunciado de validación | Mecanismo de implementación |
| id\_usuario |  |  |  |  |  |  |
| id\_gusto |  |  |  |  |  |  |
| fecha |  |  |  |  |  |  |
| mismos\_gustos |  |  |  |  | Mismos\_gustos no puede ser nulo | mismos\_gustos bit not null |

**cita**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Llave foránea | Not null | Unique | Check | Enunciado de validación | Mecanismo  de implementa-cion |
| id\_usuario\_solicitante |  |  |  |  |  |  |
| id\_usuario\_solicitado |  |  |  |  |  |  |
| fecha\_hora |  |  |  |  |  |  |
| lugar |  |  |  |  |  |  |
| estado |  |  |  |  | El estado no puede ser nulo | estado bit not null |

* Definición de integridad referencial

Relación “Usuario” y atributos referenciantes id\_ país, id\_ estado,

id\_municipio y id\_ orientación

En realidad, si queremos eliminar alguno de estos atributos significa que queremos borrarlo del registro de usuario así que la eliminación de estos solo se tratará de asignarle el valor nulo, en cambio si queremos actualizarlos debemos de asegurarnos que el nuevo dato exista en la relación original a la que pertenece, por lo que será una actualización restringida.

Relación “Usuario” y atributo referenciado id\_usuario

Id\_usuario es la clave principal de la relación Usuario, si queremos actualizar su id se debe de actualizar en cualquier tabla que use también ese atributo por lo que usaremos cascada. Si queremos borrarlo significa que deseamos borrar el usuario, por lo que en el momento en que se quiera borrar el id se debe de eliminar todo el registro, pero no la información que dejo en las demás relaciones que usan el id por lo que a las otras relaciones le pondremos nulo.

Relación “gusto” y atributo referenciado id \_ gusto

Si queremos actualizar el atributo id\_ gusto es necesario que se actualice también en el resto de las relaciones en que se usa por lo que deberá ser actualización en cascada. En caso de que se quiera eliminar todo registro de ese gusto de la base de datos se deberá eliminar en cascada y cualquier registro de él en las otras relaciones.

Relación “gusto” y atributo referenciante id \_ categoría

Debido a que los valores que toma id\_ categoría están predeterminados en la relación gusto, si queremos actualizar debemos de asegurarnos que el nuevo valor esté presente en gusto, por lo que será una actualización restringida. Tampoco se puede eliminar del registro ya que es un dato predeterminado con el que la base de datos funciona, no un dato que algún usuario ingresa.

Relación “gusta” y atributos referenciados y referenciantes id\_usuario y id \_ gusto

Juntos son la clave primaria de la relación “gusta” pero son foráneos, debemos de permitir la actualización de ambos atributos, pero la operación se lleva a cabo en las relaciones gusto y usuario ya que no podemos actualizar ese registro con el id de otro usuario ya que cada usuario tiene su propio registro. En cuanto a la eliminación junto con los id’s se irá el resto del registro en la misma relación, sin embargo, no se eliminan de otras relaciones.

Relación “cita” y atributos referenciados y referenciantes id \_ usuario\_ solicitante y id\_usuario \_ solicitado

No se permite la actualización o eliminación de ninguno de los dos atributos, la única manera en que se actualiza es desde la relación Usuario y la eliminación no es posible porque debemos de mantener un registro de las citas de los usuarios. La única manera en que desaparecerán será cuando se elimine el usuario de la base de datos y en caso de que pase solo se les asignará el valor nulo.

Relación “categoría” y atributo referenciado id \_ categoría

Los valores que toma id\_ categoría están contados por lo tanto es un dato que no podemos eliminar ya que causaría problemas en las tablas que toman prestado el id. Podemos actualizarlo, pero por la misma razón explicada para eliminar, no queremos que nos afecte las demás tablas por lo que la actualización será en cascada.

Relación “orientación” y atributo referenciado id \_ orientación

Al igual que con categoría el id de orientación no puede ser eliminado, pero puede ser actualizado y debido a que determina la orientación sexual de todos los usuarios es necesario que la actualización sea en cascada.

Relación “país” y atributo referenciado id \_ país

No podemos eliminar el id\_ país de la base de datos. Si se desea actualizar deberá de ser por un valor que no se vaya a repetir y la actualización será en cascada.

Relación “estado” y atributo referenciado id \_ estado

La eliminación de id\_ estado no es posible. Si se desea actualizar deberá ser por un valor único y la actualización será en cascada.

Relación “estado” y atributo referenciante id \_ país

La actualización de id\_ país dependerá de la existencia de dicho país en la relación país, por lo tanto, es restringida. Para la eliminación del atributo pondremos el valor nulo.

Relación “municipio” y atributo referenciado id \_ municipio

Si se desea actualizar deberá ser en cascada para que no se pierda la conexión con las otras tablas que la utilizan. La eliminación no es posible, cualquier país, estado o municipio no necesita del usuario para existir en la base de datos.

Relación “municipio” y atributo referenciante id \_ estado

Si necesitamos actualizar id\_ estado debemos de asegurarnos de que el nuevo id exista en la relación estado, de modo que la actualización necesita usar restrict. Para la eliminación solo sustituiremos el valor por un nulo.





