



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

REPORTE EJECUTIVO PROYECTO

Proyecto final. Zoológico de Huitziltepec

ASIGNATURA

Fundamentos de Bases de Datos

PROFESOR

Gerardo Avilés Rosas

AYUDANTES

Gerardo Uriel Soto Miranda
Valeria Fernanda Manjarrez Angeles
Ricardo Badillo Macías
Jerónimo Almeida Rodríguez

ALUMNOS

Castañón Maldonado Carlos Emilio
Chávez Zamora Mauro Emiliano
Gallegos Diego Cristian Ricardo
Navarro Santana Pablo César
Nepomuceno Escarcega Arizdelcy Lizbeth





Índice

1. Decisiones de diseño	3
1.1. Entidades y entidades débiles	3
1.2. Relaciones	5
2. Diccionario	7
2.1. Correo	7
2.2. Teléfono	7
2.3. Cliente	7
2.4. Atender	7
2.5. Notificar	8
2.6. Asistir	8
2.7. Evento	8
2.8. Animal	8
2.9. Jaula	9
2.10. Alimentar y Subministrar	9
2.11. Cuidador	9
2.12. Alimentos	10
2.13. Medicina	11
2.14. Laborar	11
2.15. Veterinario	11
2.16. Proveedor	12
2.17. Servicios	13
2.18. Ofrecer	13
2.19. Ticket	13
2.20. Bioma	14
3. Consultas	14
4. Obtención de datos calculados	27
5. Procedimientos Almacenados	28
6. Funciones	29
7. Disparadores	31
8. Poblamiento de la Base de Datos	32
9. Archivos presentes en este proyecto	33



1. Decisiones de diseño

Diseño del Modelo Entidad-Relación

1.1. Entidades y entidades débiles

Este documento se ha planteado para ir respondiendo qué decisiones tomamos conforme se va siguiendo los requerimientos del documento “Caso de Uso: Zoológico de Huitziltepec”.

Para el diseño de la Base de Datos del Zoológico de Huitziltepec decidimos primero considerar todas las entidades que podrían ser de utilidad según los requerimientos presentados en el caso de uso. Parcialmente la solución se planteó tomando en consideración que debíamos almacenar datos básicos para dar seguimiento a los trabajadores como generalidad (RFC, nombre completo, dirección, etc.), primeramente se estableció una entidad Trabajador que agrupara estas características compartidas. Fue importante tomar en cuenta que varios eran multivaluados o que debían ser segmentados de manera que pudieran tomar valores atómicos (ej. nombre completo se dividió en nombre, apellido paterno, apellido materno). Pero también estas características eran comunes a los clientes, así que para darle claridad al diseño nos apoyamos en dos tipos de herencia, una que divide a una entidad Persona en Trabajador o Cliente, mientras que Trabajador a su vez se divide en Cuidador, Veterinario o Proveedor (con sus respectivos atributos particulares según el tipo de entidad).

Persona tiene todos los elementos comunes a cualquier tipo de persona: género, nombreC (nombre,paterno,materno), teléfono(multivaluado), correo(multivaluado). Para elegir una llave, decidimos crear una llave artificial por dos motivos; esto permite que la herencia en disyunción haga que el tipo de entidad padre pueda heredar a los hijos dicha llave y porque el único otro dato candidato a ser llave era el RFC, pero cuenta con la peculiaridad de que nos sería desconocido el RFC en los clientes (puesto que no es una información que deba manejar el zoológico), además que se han dado casos cada vez más raros (pero posibles) de que dos personas compartan el mismo RFC. Para evitar tener que lidiar como homonimias se eligió llevar un registro aparte de la identidad de sus usuarios con un identificador que sólo funcione dentro de la lógica interna de la base de datos.

Fuentes:

- ✦ Homonimia en servicios financieros:
<https://revista.condusef.gob.mx/2022/09/homonimia/>
- ✦ Corrección de RFC por homonimia:
<https://idconline.mx/fiscal-contable/2018/08/24/correccion-de-rfc-por-homonimia>
- ✦ Prodecon consiguió la corrección de una homonimia en el RFC:
<https://www.elcontribuyente.mx/2021/09/prodecon-consiguio-la-correccion-de-una-homonimia-en-el-rfc/>

La clase Trabajador agrupa a los tres tipos de trabajadores con sus atributos comunes: RFC, Dirección(Calle, Colonia, Estado, NumInterior, NumExterior), FechInicioC, FechFinC y FechNacimiento. Además los tipos de trabajadores tienen sus propios atributos:

- ✦ Veterinario: Salario, Especialidad.
- ✦ Cuidador: Salario, AnimalCargo, DiasTrabajo, Horario.
- ✦ Proveedor: Producto, Frecuencia, Costo.

Notamos que *Salario* no es atributo de la clase trabajador en términos de herencia, puesto que no sería común a Proveedor. Las herencias poseían disyunción en ambos casos, porque no sucede que alguien actúe simultáneamente como cliente y trabajador al mismo tiempo. De la misma manera que no es posible que alguien tenga el trabajo de veterinario/proveedor o cuidador/veterinario por el rol que desempeñan en el zoológico.

El hecho de que un veterinario a lo más pudiera trabajar en dos biomas a la vez que un cuidador pudiera trabajar sólo en un bioma, no era algo posible de expresar en un Modelo Entidad-Relación, así que esa indicación se dejó para la implementación concreta de la base de datos.

En el orden del caso de uso, tocaría decidir cómo representar a los biomas, lo haremos a través de la entidad Bioma que tendrá como llave primaria a *idBioma* (ya que no tenemos ninguna llave natural que pueda ser utilizada para identificar a los biomas), el tipo de bioma será representado por el atributo *tipoBioma* ya que

no poseen cambios significativos en su gestión, por lo informado por el cliente, más allá de cambiar la categoría (desierto, pastizales, franja costera, tundra, aviario, bosque templado, bosque tropical), el número de jaulas (*NumJaulas*), el número de veterinarios (*NumVeterinarios*), el número de cuidadores (*NumCuidadores*) y el número de animales (*NumAnimales*) son todos atributos que pueden ser estimados a través de los registros en la base de datos, así que los designaremos como atributos calculados.

Dentro de Bioma también sabemos que pueden ofrecer servicios, así que crearemos una entidad Servicio la cual especifique en sus atributos el costo unitario (*costoUnitario*) y el tipo de servicio (*tipoServicio*), esto es un atributo en vez de un tipo de entidad porque para su administración no especifican que deban tratarse de manera diferente o tengan atributos aparte según el tipo de servicio. Se generará una llave artificial (*idServicio*) ya que no tenemos ninguna llave que pueda usarse como llave natural.

Ahora aparece animal como una parte importante, elegimos crear la entidad Animal que posee altura (*Altura*), peso (*Peso*), sexo (*Sexo*), alimentación (*Alimentación*), especie (*Especie*) y nombre (*Nombre*). Su identificador será *idAnimal* porque no tenemos ninguna llave natural que podamos utilizar, no existe en esos datos que sean únicos para identificar a los animales de manera sencilla.

También nos pedirán que un animal tenga una única jaula asociada, esta es la segunda vez que nos piden algo relacionado a una jaula a través de su relación con dos entidades distintas (Bioma y Animal), por lo que consideramos oportuno llevar una entidad Jaula que tenga asignado un Animal. Como la existencia de una jaula depende que exista un animal al cual contener, vamos a crearla como una entidad débil de manera que debemos especificar una relación débil entre ambas entidades (la cual llamaremos *Pertenecer*). Una jaula puede contener a varios animales (el problema no dice nada al respecto) pero un animal sólo puede estar asociado a una jaula, la relación tiene participación obligatoria porque no es posible tener un animal sin jaula ni una jaula sin animales. Su identificador será *idJaula* en combinación con el *idAnimal* ya que es una entidad débil.

Un animal sólo puede ser cuidado por un cuidador y atendido por varios veterinarios sólo si son de su mismo bioma, pero eso lo dejamos para la implementación porque no es posible de limitar en esta parte del diseño, más allá de la cardinalidad de las relaciones.

Ahora debemos modelar aquellas partes que nos permitan tener control sobre los insumos, el caso de uso menciona características en común que pondremos en una entidad Insumo, la cual posee nombre (*Nombre*), fecha de caducidad (*FechCaducidad*), si necesita refrigeración (*Refrigeracion*) y la cantidad (*Cantidad*). Además vamos a crear una llave artificial (*idInsumo*) para identificar a los integrantes de esta entidad, puesto que no tenemos ninguna llave natural que nos pueda servir como apoyo.

La entidad Insumo va a heredar a dos entidades diferentes, a Alimento y Medicina, la herencia será con disyunción y de participación obligatoria porque un insumo sólo puede ser de alguno de esos dos tipos pero no se da el caso de que algo sea alimento y medicina simultáneamente. Cosas como "suplementos alimenticios" serán considerados medicina. En Alimento tenemos como atributo el tipo de alimento (*TipoAlim*) y en Medicina tenemos el laboratorio que lo produce (*LabProd*) junto con el lote (*Lote*). Sabemos que un alimento puede ser distribuido en varios biomas, pero eso lo definiremos en la siguiente sección, la de relaciones.

El zoológico necesita una entidad Evento, la cual nos permita tratar a los eventos que realiza unitariamente, como instancias que funcionan por sí mismas. El Evento tendrá tipo de evento (*TipoEvento*) el cual será evaluado según las opciones que nos dicen existen (social, académico, recaudación de fondos, etc.), capacidad (*Capacidad*), lista de asistencia (*Asistencia*) el cual será derivado porque podemos saber la lista de asistencia a través de relacionarlos con las entidades que heredan de persona, y además elegimos agregar el atributo fecha (*Fecha*) porque todos los eventos se realizan en jornadas concretas.

Como vamos a notificar a los clientes sobre los eventos (sus promociones, sus descuentos) haremos una entidad débil Notificación, la cual estará asociada un evento a través de la relación débil Asociar, un evento puede tener muchas notificaciones pero una notificación corresponde a solamente un evento, la participación del lado de Notificar es obligatoria por ser entidad débil (no existen notificaciones si no hay un evento que notificar), pero un evento podría no emitir notificaciones aunque se necesite llevar un registro de que se realizó.

Finalmente y para el registro del estado financiero, vamos a crear una entidad Ticket, la cual será débil, porque para garantizar la exista de uno debemos tener que se ofreció algún tipo de servicio (comida, baño o tiendas). La entidad posee tipo de servicio (el cual conseguiremos a asociarla a la entidad fuerte Servicio), la fecha (*Fecha*) para llevar el registro dentro del fisco, el cliente al que se le generó (*Cliente*) para ver si se realizó un descuento. El costo total (*CostoTotal*) se deriva a través del descuento que se le aplicó al servicio, si el porcentaje de descuento (*PorcentajeDesc*) es cero, entonces el costo total será el costo que tenemos definido en ese momento en el zoológico para el servicio brindado. Obtendremos este descuento por cada exhibición (es decir, cada que se expida un ticket), así que sólo podremos saber si hubo un descuento cuando relacionemos

un servicio a un ticket, a través de la relación débil Inscribir. Por lo que *PorcentajeDesc* será un atributo de la relación débil. La cardinalidad es 1:N porque un servicio puede emitir varios tickets y varios tickets pueden ser emitidos por un servicio, la participación es obligatoria porque todos los tickets se emiten si se ha usado un servicio y todo servicio debe emitir un ticket al usarse.

1.2. Relaciones

Ahora que hemos definido las entidades fuertes y débiles, ya poseemos especificadas cierto tipo de relaciones débiles, pero es necesario relacionar a las demás entidades para que nuestro diseño pueda brindar toda la información que sea de interés para la base de datos, así que vamos a hacer explícito cómo se definieron esas relaciones.

Para relacionar los hijos de la entidad Trabajador debemos hallar una relación que nos dé una idea de su labor en el zoológico, como Proveedor no trabaja directamente en el zoológico, sino que se encarga de distribuir insumos al mismo, vamos a tener que crear relaciones distintas para cada tipo de trabajador.

- ◆ **Cuidar:** Relaciona a Cuidador y a Animal, es de cardinalidad 1:N porque un animal sólo puede ser cuidado por un cuidador según el caso de uso, pero esto no nos limita a que los cuidadores sólo puedan cuidar a un animal. La participación es obligatoria de ambos lados porque todo cuidador debe tener un animal a su cargo y todo animal debe tener un cuidador responsable por el mismo.
- ◆ **Trabajar:** Relaciona a Cuidador y a Bioma, es de cardinalidad N:1 porque un cuidador puede sólo trabajar en un bioma, según el caso de uso, pero no limita a que un bioma no pueda tener varios cuidadores. La participación es parcial del lado del cuidador porque no necesariamente van a trabajar en un bioma, pero un bioma sí debe tener cuidadores que trabajen en él.
- ◆ **Atender:** Relaciona a Veterinario y a Animal, su cardinalidad es N:M porque un veterinario tiene restricciones sobre los biomas en los que trabaja, pero no sobre los animales a los que atiende, así que podemos entender que un animal puede ser atendido por varios veterinarios y los veterinarios por supuesto deben atender a varios animales. La participación es total del lado de Animal porque un animal debe ser atendido por un veterinario, pero parcial del lado de Veterinario porque no necesariamente los veterinarios van a estar atendiendo animales si estos no están enfermos en el momento.
- ◆ **Laborar:** Elegimos el nombre porque "Trabajarza estaba ocupado como una relación entre Cuidador/-Bioma, así que Laborar nos da la relación entre Veterinario y Bioma. El caso de uso menciona que un veterinario puede trabajar/laborar en a lo más 2 biomas, así que la cardinalidad de 1 se descarta aunque tenga una restricción que debamos implementar en el código de la base de datos. Además un bioma puede poseer varios veterinarios que trabajen en él, así que la relación se definió como N:M. La participación es obligatoria del lado de Veterinario porque todo veterinario debe de tener un bioma en el cual laborar y total del lado de Bioma porque todos deben de disponer de un servicio médico de emergencia en caso de que un animal se sienta mal.
- ◆ **Proveer:** Relaciona a Proveedor e Insumo, un insumo sólo puede ser provisto por un proveedor a la vez, pero un mismo proveedor puede proveer varios insumos, así que la relación es 1:N con la unidad del lado de proveedor. La participación es total de ambos lados, porque todo insumo tuvo que haber sido provisto por algún proveedor y a la vez no nos es de utilidad tener proveedores que no provean algún insumo.

Ahora procederemos a evaluar qué relaciones podemos definir a través de la entidad Bioma, siguiendo el orden en que fueron presentadas las entidades en el caso de uso.

- ◆ **Tener:** Una entidad muy evidente es que los biomas deben tener jaula, Tener relaciona a Jaula y Bioma, la participación es N:1 porque un bioma puede tener varias jaulas, pero una jaula sólo puede estar en un bioma a la vez. La participación es obligatoria de ambos lados de la relación, porque todos los biomas deben tener jaulas y a la vez las jaulas necesariamente deben estar en un bioma.
- ◆ **Ofrecer:** Relaciona a Bioma y a Servicio, la cardinalidad es N:M porque un Bioma puede ofrecer varios servicios y un servicio puede ser ofrecido por varios biomas, por ejemplo, un bioma puede tener tienda de souvenir (zoovenirs) y a su vez las tiendas de souvenir pueden estar en varios biomas. La participación es obligatoria de ambos lados, porque si bien no es necesario que todos los biomas ofrezcan los tres

servicios, deben ofrecer al menos uno de ellos (baño) y todos los servicios están localizados en un bioma necesariamente (su lugar físico).

De las relaciones que van a Servicio, tenemos:

- ♦ **Inscribir:** Esta relación débil la definimos desde que creamos a la entidad débil Ticket, acá reafirmamos la decisión. Cardinalidad de 1:N de Servicio a Ticket porque un servicio puede emitir varios tickets, pero un ticket sólo puede ser emitido por un servicio. Participación obligatoria de ambos lados porque todo servicio genera un ticket, y todo ticket es generado por un servicio.

Ahora por secuencia según el caso de uso, debemos ver qué otras relaciones establecer a partir de la entidad Animal:

- ♦ **Pertenecer:** Esta relación débil la definimos desde que creamos a la entidad débil Jaula, acá reafirmamos la decisión. Cardinalidad de N:1 de Animal a Jaula porque una jaula puede tener varios animales, pero un animal sólo puede estar dentro de una jaula. Participación total de ambos lados porque un animal necesariamente debe estar en una jaula y una jaula necesariamente debe contener a algún animal para que tenga sentido su función en el zoológico.
- ♦ **Alimentar:** Todo animal necesita tener un alimento designado, relaciona a Animal y Alimento. Es cardinalidad N:M porque un animal puede tener varios alimentos asignados y un mismo alimento puede darse a múltiples animales. Participación total del lado de Animal porque todo animal debe tener un régimen alimenticio asignado (necesitamos que coman), pero parcial del lado de alimento porque no necesariamente todos los alimentos están siendo dados a los animales (pueden estar en bodega almacenados para el futuro).
- ♦ **Subministrar:** Relaciona a Animal y Medicina, cardinalidad N:M porque un animal puede estar tomando varias medicinas al mismo tiempo, y las medicinas pueden usarse para tratar a varios animales al mismo tiempo (un jarabe, por ejemplo, puede usarse en varios felinos). La participación es parcial del lado de Animal porque no todos los animales tomarán medicina, pero total del lado de medicina porque toda medicina debe ser suministrada a algún animal.

Ahora siguiendo el orden del caso de uso, tenemos las siguiente relaciones para Evento:

- ♦ **Asociar:** Esta relación débil la definimos desde que creamos la entidad débil Notificación, aquí la verificamos. Tiene cardinalidad 1:N desde Evento a Notificación porque un evento puede tener varias notificaciones, pero una notificación sólo puede corresponder a un evento. Es participación total del lado de Notificación porque, además de ser una entidad débil, sólo podemos tener notificaciones si existen eventos. Pero es participación parcial del lado de Evento porque podemos tener eventos que no requieran ser notificados, para gestión interna del zoológico.
- ♦ **Asistir:** Esta relación relaciona a Evento y Persona. Es de cardinalidad N:M porque a un evento pueden asistir varias personas, pero también una persona puede asistir a varios eventos. La participación es parcial de ambos lados, porque podría darse el caso de que a un evento no asista nadie o que una persona no asista a ningún evento.
- ♦ **Notificar:** Relaciona a Cliente y Notificación, es de cardinalidad N:M porque un cliente puede recibir varias notificaciones y una notificación (como una promoción) se puede emitir a varios clientes. La participación es parcial de ambos lados de la relación porque hay notificaciones que no llegaran a ciertos clientes (como un determinado descuento) o hay clientes que no querrán recibir notificaciones, no ser notificados.

Finalmente sólo debemos hacer caso a qué relaciones podrá generar la entidad Ticket (sin contar las previamente descritas), las cual es la siguiente:

- ♦ **Emitir:** Relaciona a Cliente y a Ticket con una cardinalidad 1:N, porque a un cliente se le pueden emitir diversos tickets pero un ticket sólo podrá ser emitido a un cliente. La participación es parcial del lado de cliente porque un cliente puede visitar al zoológico sin que se le emita un ticket por servicios, pero un ticket por servicio necesariamente debe ser emitido a una persona.

Hemos terminado de revisar las decisiones de diseño para el Modelo Entidad/Relación.

2. Diccionario

2.1. Correo

- ◆ De las tablas CorreoCliente, CorreoVeterinario, CorreoCuidador y CorreoProveedor:
Almacena direcciones de correo electrónico asociada a una persona.
 - ◆ Correo : Es el correo electrónico asociado al idPersona y su tipo de datos es una cadena de caracteres de longitud 50. **Rango de dominio:** Cualquier cadena de texto de longitud máxima 50.
 - ◆ IdPersona : Es el id de la persona el cual su tipo de dato es un número entero de 8 bytes. **Rango de dominio:** Números enteros consecutivos.

2.2. Teléfono

- ◆ Tabla Teléfono:
Almacena números de teléfono asociados a una persona (TelefonoCliente, TelefonoVeterinario, TelefonoCuidador y TelefonoProveedor).
 - ◆ IdPersona : Es el id de la persona el cual su tipo de dato es un número entero de 8 bytes. **Rango de dominio:** Números enteros consecutivos.
 - ◆ Telefono : Número entero de 8 bytes que representa el número de teléfono asociado a la persona. **Rango de dominio:** Sucesiones de números de longitud de 10 dígitos.

2.3. Cliente

- ◆ Tabla Cliente:
Almacena información sobre clientes.
 - ◆ IdPersona : Es el id de la persona el cual su tipo de dato es un número entero de 8 bytes. **Rango de dominio:** Números enteros consecutivos.
 - ◆ Nombre : Cadena de caracteres con una longitud máxima de 50, que representa el nombre del cliente. **Rango de dominio:** Cualquier cadena de texto de longitud máxima 50.
 - ◆ Paterno : Cadena de caracteres con una longitud máxima de 50, que representa el apellido paterno del cliente. **Rango de dominio:** Cualquier cadena de texto de longitud máxima 50.
 - ◆ Materno : Cadena de caracteres con una longitud máxima de 50, que representa el apellido materno del cliente. **Rango de dominio:** Cualquier cadena de texto de longitud máxima 50.
 - ◆ Género : Es un carácter de longitud 1 que indica el género del cliente (M/H/X). **Rango de dominio:** Caracteres individuales que sean iguales a 'M', 'H' o 'X'.

2.4. Atender

- ◆ Tabla Atender:
Registra la atención médica brindada por personas a animales.
 - ◆ IdPersona : Número entero de 8 bytes, sirve como identificador único para la persona que brinda la atención médica. **Rango de dominio:** Números enteros consecutivos.
 - ◆ IdAnimal : Número entero de 4 bytes, sirve como identificador único para el animal que recibe la atención médica. **Rango de dominio:** Números enteros consecutivos.
 - ◆ IndMedicas: Tipo texto que permite almacenar información relacionada con la atención médica. **Rango de dominio:** Texto con una longitud máxima de cadena de 2,147,483,647.

2.5. Notificar

◆ Tabla Notificar:

Establece la relación entre personas y notificaciones, registrando quién está asociado con qué notificación.

- ◆ IdPersona : Número entero de 8 bytes, que sirve como identificador único para la persona vinculada con la notificación. **Rango de dominio:** Números enteros consecutivos.
- ◆ IdNotificacion: Número entero de 8 bytes, que sirve como identificador único para la notificación. **Rango de dominio:** Números enteros consecutivos.
- ◆ IdEvento: Número entero de 4 bytes, que sirve como identificador único para el evento vinculado con la notificación. **Rango de dominio:** Números enteros consecutivos.

◆ Tabla Notificacion:

Asocia eventos con notificaciones.

- ◆ IdEvento: Número entero de 4 bytes, que sirve como identificador único para el evento asociado con la notificación. **Rango de dominio:** Números enteros consecutivos.
- ◆ IdNotificacion: Número entero de 8 bytes, que sirve como identificador único para la notificación en cuestión. **Rango de dominio:** Números enteros consecutivos.

2.6. Asistir

◆ De las tablas AsistirCliente, AsistirVeterinario, AsistirProveedor y AsistirCuidador:

Registra la asistencia de las personas a los eventos.

- ◆ IdPersona : Número entero de 8 bytes, que sirve como identificador único para el veterinario que asiste al evento. **Rango de dominio:** Números enteros consecutivos.
- ◆ IdEvento: Número entero de 4 bytes, que sirve como identificador único para el evento al cual el veterinario asiste. **Rango de dominio:** Números enteros consecutivos.

2.7. Evento

◆ Tabla Evento:

Almacena información sobre eventos planificados en el sistema.

- ◆ IdEvento : Número entero de 4 bytes, que sirve como identificador único para cada evento. **Rango de dominio:** Números enteros consecutivos.
- ◆ Capacidad : Número entero de 4 bytes que indica la cantidad máxima de participantes permitidos en el evento. **Rango de dominio:** Valores que den el máximo de la capacidad establecida por protección civil, al ser desconocido, se dará la opción al zoológico de tener de 0 hasta 4,294,967,296 asistentes.
- ◆ Fecha : Tipo fecha que registra la fecha en la que el evento está programado. **Rango de dominio:** Cualquier fecha.
- ◆ TipoEvento: Cadena de caracteres con una longitud máxima de 50, que especifica el tipo o nombre del evento. **Rango de dominio:** Cualquier cadena de longitud máxima 50.

2.8. Animal

◆ Tabla Animal:

Almacena información detallada sobre animales en el sistema.

- ◆ IdAnimal : Número de 4 bytes, que sirve como identificador único para cada animal. **Rango de dominio:** Números enteros consecutivos.
- ◆ Nombre : Cadena de caracteres con una longitud máxima de 50, que representa el nombre del animal. **Rango de dominio:** Cualquier cadena de longitud máxima 50.

- ❖ Especie : Cadena de caracteres con una longitud máxima de 50, que indica la especie del animal. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
- ❖ Alimentación : Tipo texto que describe la dieta o alimentación del animal. **Rango de dominio:** Texto con una longitud máxima de cadena de 2,147,483,647.
- ❖ Sexo : Carácter de longitud 1 que especifica el sexo del animal. **Rango de dominio:** Caracteres individuales que sean iguales a 'M' o 'H'.
- ❖ Peso : Número de punto flotante de 4 bytes que indica el peso del animal. **Rango de dominio:** Cualquier valor de punto flotante.
- ❖ Altura : Número de punto flotante de 4 bytes que indica la altura del animal. **Rango de dominio:** Cualquier valor de punto flotante.
- ❖ IdPersona : Número entero auto incremental de 8 bytes, que sirve como identificador único para la persona responsable del animal. **Rango de dominio:** Números enteros consecutivos.

2.9. Jaula

- ❖ Tabla Jaula:
Relaciona el numero de jaulas, con los animales y sus biomas en el que se encuentran
 - ❖ IdAnimal: Número de 4 bytes, que sirve como identificador único para cada animal. **Rango de dominio:** Números enteros consecutivos.
 - ❖ NumJaula: Número de 4 bytes, que sirve como identificador único para cada jaula. **Rango de dominio:** Números enteros consecutivos.
 - ❖ IdBioma: Número de 4 bytes, que sirve como identificador único para cada bioma. **Rango de dominio:** Números enteros consecutivos.

2.10. Alimentar y Subministrar

- ❖ De las tablas Alimentar y Subministrar:
Relación entre un animal y el insumo que se le administro al animal.
 - ❖ IdAnimal: Número de 4 bytes, que sirve como identificador único para cada animal. **Rango de dominio:** Números enteros consecutivos.
 - ❖ IdInsumo: Número de 4 bytes, que sirve como identificador único para el insumo en cuestión. **Rango de dominio:** Números enteros consecutivos.

2.11. Cuidador

- ❖ Tabla cuidador:
Almacena información sobre los cuidadores en el sistema.
 - ❖ IdPersona : Número entero de 8 bytes, que sirve como identificador único para cada cuidador. **Rango de dominio:** Números enteros consecutivos.
 - ❖ IdBioma: Número entero de 4 bytes, que indica el bioma donde trabaja. **Rango de dominio:** Números enteros consecutivos.
 - ❖ Nombre: Cadena de caracteres con una longitud máxima de 50, que representa el nombre del cuidador. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
 - ❖ Paterno: Cadenas de caracteres con una longitud máxima de 50, que indica el apellido paterno del cuidador. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
 - ❖ Genero: Carácter de longitud 1 que especifica el género del cuidador. **Rango de dominio:** Caracteres individuales que sean iguales a 'M', 'H' o 'X'.

- ❖ Materno: Cadenas de caracteres con una longitud máxima de 50, que indica el apellido materno del cuidador. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
- ❖ RFC : Cadena de caracteres con una longitud fija de 13. **Rango de dominio:** Cualquier cadena de longitud de 13 justa.
- ❖ FechNacimiento : Tipo de fecha que registra la fecha de nacimiento del cuidador. **Rango de dominio:** Cualquier fecha.
- ❖ Calle : Cadena de caracteres con una longitud máxima de 50, que indica la calle del cuidador. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
- ❖ NumInterior : Número interior que representa la residencia del cuidador. **Rango de dominio:** Números enteros.
- ❖ NumExterior : Número exterior que representa la residencia del cuidador. **Rango de dominio:** Números enteros.
- ❖ Colonia : Cadena de caracteres con una longitud máxima de 50, que indica la colonia de residencia del cuidador. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
- ❖ Estado : Cadena de caracteres con una longitud máxima de 50, que indica el estado de residencia del cuidador. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
- ❖ FechInicioC : Indica el tipo de fecha de inicio del contrato del cuidador. **Rango de dominio:** Cualquier fecha.
- ❖ FechFinC : Indica el tipo de fecha del final del contrato del cuidador. **Rango de dominio:** Cualquier fecha.
- ❖ Salario : Número de punto flotante de 4 bytes que representa el salario del cuidador. **Rango de dominio:** Cualquier valor de punto flotante.
- ❖ DiasTrabajo: Cadena de caracteres con una longitud máxima de 50, que indica los días de trabajo del cuidador. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
- ❖ Horario: Número entero que representa el horario de trabajo del cuidador. **Rango de dominio:** Cualquier valor entero.

2.12. Alimentos

- ❖ Tabla Alimento:
Alimentos proveídos por proveedores con todas sus características.
 - ❖ IdInsumo: Número de 8 bytes, que sirve como identificador único para cada insumo. **Rango de dominio:** Números enteros consecutivos.
 - ❖ IdPersona : Número de 8 bytes, que sirve como identificador único para cada persona. **Rango de dominio:** Números enteros consecutivos.
 - ❖ Nombre : Carácter de longitud 50 que especifica el nombre del alimento. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
 - ❖ FechCaducidad : Tipo fecha que registra la fecha de caducidad del alimento. **Rango de dominio:** Cualquier fecha.
 - ❖ Cantidad : Numero entero que indica la cantidad del alimento en cuestión. **Rango de dominio:** Números enteros.
 - ❖ Refrigeracion : Booleano que indica si requiere refrigeración el alimento. **Rango de dominio:** Booleano de valor 'true' o 'false'.
 - ❖ TipoAlim : Carácter de longitud 50 que especifica el tipo de alimento. **Rango de dominio:** Cualquier cadena de longitud máxima 50.

2.13. Medicina

♦ Tabla Medicina:

Medicinas proveídas por proveedores con todas sus características.

- ♦ IdInsumo: Número de 8 bytes, que sirve como identificador único para cada insumo. **Rango de dominio:** Números enteros consecutivos.
- ♦ IdPersona : Número de 8 bytes, que sirve como identificador único para cada persona. **Rango de dominio:** Números enteros consecutivos.
- ♦ Nombre : Carácter de longitud 50 que especifica el nombre de la medicina. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
- ♦ FechCaducidad : Tipo fecha que registra la fecha de caducidad de la medicina. **Rango de dominio:** Cualquier fecha.
- ♦ Cantidad : Numero entero que indica la cantidad de la medicina en cuestión. **Rango de dominio:** Números enteros.
- ♦ Refrigeracion : Booleano que indica si requiere refrigeración la medicina en cuestión.
- ♦ LabProd : Carácter de longitud 50 que especifica el laboratorio de procedencia de la medicina en cuestión. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
- ♦ Lote : Carácter de longitud 50 que especifica el lote de la medicina. **Rango de dominio:** Cualquier cadena de longitud máxima 50.

2.14. Laborar

♦ Tabla Laborar:

Registra qué personas trabajan en qué biomas; indicando en qué áreas específicas están asignadas para trabajar.

- ♦ IdPersona : Número entero de 8 bytes, que sirve como identificador único para la persona asociada al bioma. **Rango de dominio:** Números enteros consecutivos.
- ♦ IdBioma : Número entero de 4 bytes, que indica el bioma al cual la persona está asignada. **Rango de dominio:** Números enteros consecutivos.

2.15. Veterinario

♦ Tabla Veterinario:

Almacena información sobre los veterinarios.

- ♦ IdPersona : Número entero de 8 bytes, que sirve como identificador único para la persona asociada al bioma. **Rango de dominio:** Números enteros consecutivos.
- ♦ Nombre: Carácter de longitud 50 que especifica el nombre del veterinario. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
- ♦ Paterno: Carácter de longitud 50 que especifica el apellido paterno del veterinario. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
- ♦ Materno: Carácter de longitud 50 que especifica el apellido materno del veterinario. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
- ♦ Género : Es un carácter de longitud 1 que indica el género del veterinario (M/H/X). **Rango de dominio:** Caracteres individuales que sean iguales a 'M', 'H' o 'X'.
- ♦ RFC : Cadena de caracteres con una longitud fija de 13, que representa el registro del veterinario.
- ♦ FechNacimiento : Tipo fecha que registra la fecha de nacimiento del veterinario. **Rango de dominio:** Cualquier fecha.

- ❖ Calle : Carácter de longitud 50 que especifica la calle donde vive el veterinario. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
- ❖ NumInterior : Numero entero que especifica el numero interior de la calle donde vive el veterinario. **Rango de dominio:** Números enteros.
- ❖ NumExterior : Numero entero que especifica el numero exterior de la calle donde vive el veterinario. **Rango de dominio:** Números enteros.
- ❖ Colonia : Carácter de longitud 50 que especifica la colonia donde vive el veterinario. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
- ❖ Estado : Carácter de longitud 50 que especifica el estado donde vive el veterinario. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
- ❖ FechInicioC : Tipo fecha que registra la fecha del inicio del contrato del veterinario. **Rango de dominio:** Cualquier fecha.
- ❖ FechFinC : Tipo fecha que registra la fecha del fin del contrato del veterinario. **Rango de dominio:** Cualquier fecha.
- ❖ Salario : Número de punto flotante de 4 bytes que indica el salario del veterinario. **Rango de dominio:** Cualquier valor de punto flotante.
- ❖ Especialidad : Carácter de longitud 50 que especifica la especialidad del veterinario. **Rango de dominio:** Cualquier cadena de longitud máxima 50.

2.16. Proveedor

❖ Tabla Proveedor:

Almacena información sobre proveedores y los productos que suministran.

- ❖ IdPersona : Número entero de 8 bytes, que sirve como identificador único para cada proveedor. **Rango de dominio:** Números enteros consecutivos.
- ❖ Nombre: Cadena de caracteres con una longitud máxima de 50, que representa el nombre del proveedor. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
- ❖ Paterno: Cadena de caracteres con una longitud máxima de 50, que representa el apellido paterno del proveedor. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
- ❖ Materno: Cadena de caracteres con una longitud máxima de 50, que representa el apellido materno del proveedor. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
- ❖ Género : Carácter de longitud 1 que indica el género del proveedor. **Rango de dominio:** Caracteres individuales que sean iguales a 'M', 'H' o 'X'.
- ❖ RFC : Cadena de caracteres con una longitud fija de 13, que representa el registro del proveedor.
- ❖ FechNacimiento : Fecha que registra la fecha del nacimiento del proveedor. **Rango de dominio:** Cualquier fecha.
- ❖ Calle : Cadena de caracteres con una longitud máxima de 50, que indica la calle de residencia del proveedor. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
- ❖ NumInterior : Número interior que representa la residencia del cuidador. **Rango de dominio:** Números enteros.
- ❖ NumExterior : Número exterior que representa la residencia del cuidador. **Rango de dominio:** Números enteros.
- ❖ Colonia : Cadena de caracteres con una longitud máxima de 50, que indica la colonia de residencia del proveedor. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
- ❖ Estado : Cadena de caracteres con una longitud máxima de 50, que indica el estado de residencia del proveedor. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
- ❖ FechInicioC : Indica el tipo de fecha de inicio del contrato del cuidador. **Rango de dominio:** Cualquier fecha.

- ❖ FechFinC : Indica el tipo de fecha del final del contrato del cuidador. **Rango de dominio:** Cualquier fecha.
- ❖ Producto : Cadena de caracteres con una longitud máxima de 50, que indica el producto que el proveedor suministra. **Rango de dominio:** Cualquier cadena de longitud máxima 50.
- ❖ Frecuencia : Número entero que representa la frecuencia con la que se suministra el producto. **Rango de dominio:** Cualquier valor entero no negativo.
- ❖ Costo : Número de punto flotante de 4 bytes que indica el costo del producto suministrado por el proveedor. **Rango de dominio:** Cualquier valor de punto flotante.

2.17. Servicios

- ◆ Tabla Servicio:
Tipos de servicios que se ofrecen en los diferentes biomas.
 - ❖ IdServicio: Número entero de 4 bytes, que sirve como identificador único para el servicio ofrecido. **Rango de dominio:** Números enteros consecutivos.
 - ❖ CostoUnitario: Número de punto flotante de 4 bytes que indica el costo del servicio. **Rango de dominio:** Cualquier valor de punto flotante.
 - ❖ TipoServicio: Cadena de caracteres con una longitud máxima de 50 que indica el tipo de servicio. **Rango de dominio:** Cualquier cadena de longitud máxima 50.

2.18. Ofrecer

- ◆ Tabla Ofrecer:
Relaciona biomas con servicios que se ofrecen en esos biomas.
 - ❖ IdBioma: Número entero de 4 bytes, que sirve como identificador único para el bioma. **Rango de dominio:** Números enteros consecutivos.
 - ❖ IdServicio: Número entero de 4 bytes, que sirve como identificador único para el servicio ofrecido. **Rango de dominio:** Números enteros consecutivos.

2.19. Ticket

- ◆ Tabla Ticket:
Registra información sobre tickets generados por servicios para personas.
 - ❖ IdTicket: Número entero de 8 bytes, que sirve como identificador único para cada ticket. **Rango de dominio:** Números enteros consecutivos.
 - ❖ IdServicio: Número entero de 4 bytes, que identifica el servicio asociado al ticket. **Rango de dominio:** Números enteros consecutivos.
 - ❖ IdPersona : Número entero de 4 bytes, que identifica a la persona relacionada con el ticket. **Rango de dominio:** Números enteros consecutivos.
 - ❖ Fecha : Tipo fecha que registra la fecha en que se generó el ticket. **Rango de dominio:** Cualquier fecha.
 - ❖ PorcentajeDesc : Número de punto flotante de 4 bytes, que indica el porcentaje de descuento aplicado al servicio. **Rango de dominio:** Cualquier valor de punto flotante.

2.20. Bioma

◆ Tabla Bioma:

Registra información sobre tickets generados por servicios para personas.

- ◆ IdBioma: Número entero de 4 bytes, que identifica el bioma. **Rango de dominio:** Números enteros consecutivos.
- ◆ TipoBioma: Cadena de caracteres con una longitud máxima de 50 que indica el tipo del bioma. **Rango de dominio:** Cualquier cadena de longitud máxima 50.

3. Consultas

- I) Consulta para obtener la información suficiente de todos los veterinarios y cuidadores que laboren en los biomas 1, 3, 5 ó 7, y que el nombre de los veterinarios empiece con la letra R y que su apellido paterno contenga la letra e.

```
SELECT
    v.idpersona, nombre, paterno, materno, genero, rfc
FROM
    (SELECT * FROM veterinario WHERE(nombre LIKE 'R%' AND LOWER(paterno) LIKE
    'e%')) AS v,
    (SELECT * FROM laborar WHERE (idbioma IN (1, 3, 5, 7))) AS l
WHERE (
    (v.idpersona = l.idpersona)
)
UNION
SELECT
    idpersona, nombre, paterno, materno, genero, rfc
FROM cuidador
WHERE (nombre LIKE 'R%' AND LOWER(paterno) LIKE 'e%' AND idbioma IN (1, 3, 5, 7));
```

	123 idpersona	ABC nombre	ABC paterno	ABC materno	ABC genero	ABC rfc	
1	48	Rosa	Gonzalez	Navarro	H	MTRT27081362G	
2	104	Roberto	Perez	Hernandez	H	TPSS7701518XV	
3	50	Ricardo	Flores	Hernandez	M	YPSN047196SDU	
4	4	Ricardo	Flores	Campos	H	WKPE5555866N4	
5	93	Raul	Rivera	Cortez	X	EXJX926235ZK1	
6	30	Rosa	Moreno	Campos	H	UOIQ231771RE4	
7	104	Rosa	Rodriguez	Perez	H	GAXZ218901488	
8	68	Rosa	Jimenez	Ramos	M	IISW6646484VR	
9	24	Rafael	Mendoza	Lopez	X	MYXY417851XBD	
10	82	Roberto	Mendoza	Rojas	H	PGUD129504Z3A	
11	45	Rosa	Romero	Gutierrez	H	SREQ857125PAJ	

- II) Consulta que regresa los nombres completos de todos los trabajadores asociados a los biomas (veterinarios y cuidadores) ordenados por el tipo de Bioma.

```
select * from
(
  select
    c.nombre,
    c.paterno,
    c.materno,
    b.tipoBioma
  from cuidador as c
  join bioma as b
  on c.idBioma = b.idBioma) as a
union
select * from
(
  select
    v.nombre,
    v.paterno,
    v.materno,
    lb.tipoBioma
  from veterinario as v
  join
    (
      select *
      from bioma
      join laborar
      on bioma.idbioma = laborar.idbioma) as lb
  on lb.idPersona = v.idPersona) as b
order by tipoBioma;
```

	ABC nombre	ABC paterno	ABC materno	ABC tipobioma	
1	Miguel	Martinez	Delgado	Aviario	
2	Luis	Torres	Jimenez	Aviario	
3	Daniel	Reyes	Sanchez	Aviario	
4	Victor	Morales	Pena	Aviario	
5	Maria	Moreno	Suarez	Aviario	
6	Antonio	Martinez	Hernandez	Aviario	
7	Gloria	Mendoza	Delgado	Aviario	
8	Daniel	Rodriguez	Reyes	Aviario	
9	Alejandro	Cruz	Gomez	Aviario	
10	Julio	Garcia	Flores	Aviario	
11	Guadalupe	Castillo	Castillo	Aviario	

- III) Alimentos de tipo vegetales y carnes, que son distribuidos en todos los biomas y ordenados por el tipo de alim e idinsumo.

```
SELECT
    tipoalim,
    idinsumo,
    idpersona,
    nombre,
    fechacaducidad,
    cantidad,
    refrigeracion,
    b.tipobioma
FROM (SELECT
    s.*,
    j.idbioma
    FROM (SELECT
        r.*
        FROM (SELECT
            al.*,
            alr.idanimal
            FROM (SELECT *
                FROM alimento
                WHERE tipoalim IN ('vegetales', 'carnes')
            ) AS al,
            alimentar AS alr
            WHERE (al.idinsumo = alr.idinsumo)
        ) AS r,
        animal AS a
        WHERE (r.idanimal = a.idanimal)
    ) AS s,
    jaula AS j
    WHERE (s.idanimal = j.idanimal)
) AS t,
(SELECT *
    FROM bioma
) AS b
WHERE (t.idbioma = b.idbioma)
ORDER BY tipoalim, idinsumo;
```

	asc tipoalim ▼	123 idinsumo ▼	123 idpersona ▼	asc nombre ▼	🕒 fechacaducidad ▼	123 cantidad ▼
1	carnes	1	1	Carne de res	2023-11-20	500
2	carnes	2	2	Pescado	2023-10-15	300
3	carnes	3	3	Pollo	2023-12-05	400
4	carnes	20	20	Carne de res	2023-12-31	500
5	carnes	24	24	Cerdos	2023-10-05	600
6	carnes	38	38	Pollo	2023-10-30	400
7	carnes	41	41	Pescado	2023-11-15	300
8	vegetales	5	5	Calabazas	2023-11-10	800
9	vegetales	21	21	Calabazas de Alt	2023-10-01	800
10	vegetales	35	35	Apio	2023-12-01	800
11	vegetales	52	52	Brocoli	2023-12-05	800

- IV) Consulta para saber cuales son los cuidadores y veterinarios que tienen el mismo estado y que el cuidador, cuide a un animal carnívoro macho en los biomas 1,3 o 5 y que la información salga ordenada.

```
SELECT
  c.nombre AS cuidador,
  v.nombre AS veterinario,
  c.estado AS cuidadorEstado,
  v.estado AS veterinarioEstado,
  a.nombre AS animal,
  a.sexo AS sexo,
  b.tipobioma AS bioma
FROM
  cuidador AS c
  INNER JOIN veterinario AS v ON c.estado = v.estado
  INNER JOIN animal AS a ON c.idpersona = a.idpersona
  INNER JOIN jaula AS j ON a.idanimal = j.idanimal
  INNER JOIN bioma AS b ON j.idbioma = b.idbioma
WHERE
  a.sexo = 'M' AND
  a.alimentacion = 'Carnivoro' AND
  b.idbioma IN (1,3,5)
ORDER BY cuidador, veterinario;
```

	ABC cuidador ▼	ABC veterinario ▼	ABC cuidadoreestado ▼	ABC veterinarioestado ▼	ABC animal ▼	ABC sexo ▼	ABC bioma ▼
1	Guadalupe	Arturo	Guanajuato	Guanajuato	Pulpo de Anillos Azules	M	Franja costera
2	Guadalupe	Hector	Guanajuato	Guanajuato	Pulpo de Anillos Azules	M	Franja costera
3	Guadalupe	Norma	Guanajuato	Guanajuato	Pulpo de Anillos Azules	M	Franja costera
4	Hector	Alma	Durango	Durango	Buho real	M	Aviario
5	Hector	Enrique	Durango	Durango	Buho real	M	Aviario
6	Hector	Gerardo	Durango	Durango	Buho real	M	Aviario
7	Hector	Juana	Durango	Durango	Buho real	M	Aviario
8	Hector	Maria	Durango	Durango	Buho real	M	Aviario
9	Hector	Maria	Durango	Durango	Buho real	M	Aviario
10	Hector	Rosa	Durango	Durango	Buho real	M	Aviario
11	Hector	Rosa	Durango	Durango	Buho real	M	Aviario


- V) Consulta que nos regresa el nombre completo del veterinario, las indicaciones médicas propuestas para el animal, junto con el nombre de la medicina que se le administra, la fecha de caducidad, laboratorio y el nombre de dicho animal.

```
select
avs.nombre,
avs.paterno,
avs.materno,
avs.indicacionesmedicas,
m.nombre as nombre_medicina,
m.fechCaducidad,
m.labProd as laboratorio,
avs.nombre_animal
from medicina as m
join
    (select *
    from subministrar as s
    join
        (select
            a.idAnimal,
            atenv.nombre,
            atenv.paterno,
            atenv.materno,
            atenv.indicacionesmedicas,
            a.nombre as nombre_animal
            from animal as a
            join (select *
                from atender as ate
                join veterinario as v
                on v.idpersona = ate.idpersona) as atenv
            on a.idAnimal = atenv.idAnimal) as avet
        on s.idAnimal = avet.idAnimal) as avs
on avs.idInsumo = m.idInsumo;
```

	ABC nombre	ABC paterno	ABC materno	ABC indicacionesmedicas	ABC nombre_medicina	fechcaducidad	ABC laboratorio	ABC nombre_animal
1	Rosa	Gonzalez	Navarro	Control de peso	Mirtazapina	2012-02-17	Lab15	Canguro
2	Maximo	Jimenez	Gutierrez	Vacuna	Cefpodoxima	2008-11-23	Lab1	Gorila
3	Patricia	Castillo	Torres	Herida	Cefpodoxima	2008-11-23	Lab1	Orangutan
4	Alma	Gonzalez	Lopez	Examen medico	Cefpodoxima	2008-11-23	Lab1	Orangutan
5	Gloria	Torres	Pena	Vacunas	Ranitidina	2014-10-30	Lab18	Koala
6	Patricia	Castillo	Torres	Herida	Atorvastatina	2008-01-12	Lab10	Orangutan
7	Alma	Gonzalez	Lopez	Examen medico	Atorvastatina	2008-01-12	Lab10	Orangutan
8	Jose	Romero	Acosta	Servicio dental	Letrozol	2016-01-08	Lab2	Panda
9	Gloria	Torres	Pena	Revision anual	Letrozol	2016-01-08	Lab2	Panda
10	Roberto	Sanchez	Navarro	Examen medico	Carvedilol	2011-04-22	Lab16	Mono
11	Arturo	Romero	Gomez	Ojo inflamado	Carvedilol	2011-04-22	Lab16	Mono

- VI) Consulta que regresa una tabla con la información de los biomas, los ids de los trabajadores que trabajan en ellos y sus teléfonos ordenado por bioma y después por el identificador de los trabajadores

```
select
aux.idBioma,
aux.tipoBioma,
tc.idPersona,
tc.telefono
from telefonoCuidador as tc
join
(select b.tipoBioma,
b.idBioma,
c.idPersona
from Cuidador as c
join Bioma as b
on c.idBioma = b.IdBioma) as aux
on tc.idPersona = aux.idPersona
order by idBioma, idPersona;
```

	 123 idbioma ▼	ABC tipobioma ▼	123 idpersona ▼	123 telefono ▼	
1	1	Desierto	2	3,735,647,611	
2	1	Desierto	20	6,818,435,088	
3	1	Desierto	30	7,287,196,361	
4	1	Desierto	42	4,555,522,660	
5	1	Desierto	45	4,491,629,119	
6	1	Desierto	54	1,127,462,902	
7	1	Desierto	64	4,748,165,890	
8	1	Desierto	71	8,658,478,841	
9	1	Desierto	72	8,096,948,116	
10	1	Desierto	77	6,649,408,177	
11	1	Desierto	80	8,102,263,102	
12	1	Desierto	85	3,518,219,960	

- VII) Consulta que regresa el número de proveedores que tenemos por estado siempre y cuando su fecha de nacimiento haya sido previa al año 2000. Debemos regresar la consulta ordenada por estado.

```
(select estado, count(idPersona)
from
(select * from proveedor
where fechnacimiento <= '01-01-2000') as a
group by estado
order by estado);
```

	estado	count	
1	Aguascalientes	7	
2	Baja California	6	
3	Baja California Sur	8	
4	Campeche	4	
5	Chiapas	6	
6	Chihuahua	4	
7	Coahuila	3	
8	Colima	6	
9	Durango	11	
10	Guanajuato	4	
11	Guerrero	6	
12	Hidalgo	9	

- VIII) Consulta que regresa toda la información de los veterinarios que trabajan en el zoológico pero además agrega el salario promedio, mínimo y máximo que se les paga según la especialidad que poseen.

```
select *
from veterinario as v
join(
select especialidad,
avg(salario) as salario_promedio,
max(salario) as salario_maximo,
min(salario) as salario_minimo
from veterinario
group by especialidad) as aux
on v.especialidad = aux.especialidad;
```

	idpersona	nombre	paterno	materno	genero	rfc	fechnacimiento	calle	numinterior	
1	1	Juan	Reyes	Rivera	X	CHHU992264OC6	1962-11-01	Juarez	74	
2	2	Martin	Jimenez	Rodriguez	X	VYTK924050GH8	2001-06-26	Reforma	32	
3	3	Alma	Castillo	Rivera	M	VKDO7168586CW	1981-08-07	Hidalgo	58	
4	4	Ricardo	Flores	Campos	H	WKPE5555866N4	1965-12-27	Hidalgo	18	
5	5	Arturo	Morales	Mendoza	X	RHOQ555874WZ7	1967-12-04	Paseo de la Reforma	17	
6	6	Claudia	Sanchez	Campos	M	ORAR103560LBM	1962-12-04	Reforma	44	
7	7	Pedro	Perez	Martinez	X	GELP9572689A7	2000-03-07	Madero	21	
8	8	Rafael	Morales	Gomez	X	ULVJ933303XHD	1979-11-27	Tacuba	69	
9	9	Claudia	Lopez	Garcia	X	AUGV656248L3P	1984-11-29	Juarez	67	
10	10	Rafael	Rodriguez	Santiago	X	DMSN459132853	1969-02-20	Insurgentes	25	
11	11	Daniel	Santos	Flores	H	IRVR705350WUT	1961-05-21	Paseo de la Reforma	22	


- IX) Consulta que nos regresa el nombre completo de los cuidadores y el animal que tiene asignado cada persona.

```
select c.nombre,
c.paterno,
c.materno,
a.nombre as animal_asignado
from animal as a
join cuidador as c
on c.idPersona = a.idPersona;
```

	nombre	paterno	materno	animal_asignado
1	Daniel	Santos	Diaz	Coyote
2	Rosa	Romero	Gutierrez	Camello
3	Alejandro	Flores	Vazquez	Serpiente de Cascabel
4	Mario	Vargas	Campos	Jabali del Desierto
5	Jorge	Perez	Morales	Guepardo
6	Guadalupe	Diaz	Rios	Ave del Roadrunner
7	Rosa	Moreno	Campos	Iguana del Desierto
8	Rosa	Ortiz	Castro	Fenec
9	Jorge	Diaz	Medina	aguila Real
10	Patricia	Diaz	Herrera	Escarabajo del Desierto
11	Silvia	Ortiz	Martinez	Leon
12	Daniel	Torres	Alvarez	Elefante

- X) Consulta que nos regresa un tipo de evento y el nombre de las personas que asistieron a ese tipo de evento.

```
select tipoevento,  
nombre,  
paterno,  
materno  
from  
  (select *  
   from cliente as c  
   join  
     (select ae.*,  
      e.tipoEvento  
     from asistirCliente as ae  
     join evento as e  
     on ae.idEvento = e.idEvento) as r  
   on c.idPersona = r.idPersona) as b  
order by tipoevento;
```

		ASC tipoevento ▼	ASC nombre ▼	ASC paterno ▼	ASC materno ▼	
1		academico	Jose	Sanchez	Ortiz	
2		academico	Ricardo	Rivera	Ortiz	
3		academico	David	Gonzalez	Santiago	
4		academico	Julio	Garcia	Cortez	
5		academico	Norma	Diaz	Cruz	
6		academico	Hector	Diaz	Hernandez	
7		academico	Hector	Martinez	Suarez	
8		academico	Raul	Reyes	Ortiz	
9		academico	Silvia	Rivera	Lopez	
10		academico	Jorge	Sanchez	Reyes	
11		academico	David	Hernandez	Ramirez	
12		academico	Eduardo	Mendoza	Sanchez	

- XI) Todos los veterinarios que hayan atendido a un animal y sean de la Ciudad de Mexico ó de Baja California Sur.

```
select *
from (SELECT veterinario.idpersona
      FROM atender, veterinario
      where veterinario.idpersona = atender.idpersona
) as a, veterinario as v
where v.estado in ('Mexico', 'Baja California Sur') and v.idpersona = a.idpersona;
```

	123 idpersona	123 idpersona	ABC nombre	ABC paterno	ABC materno	ABC genero	ABC rfc	fechnacimiento	ABC calle	123
1	13	13	Juan	Cruz	Morales	H	EISA991334V9V	1994-12-03	Insurgentes	12
2	25	25	Veronica	Martinez	Castillo	H	KFMS5021277P9	1992-05-25	Av. Chapultepec	
3	28	28	Antonio	Gonzalez	Alvarez	H	HWIA123901VK0	1969-06-05	Tacuba	
4	45	45	Rafael	Sanchez	Ramos	X	THEH6813586UK	1961-11-09	Madero	
5	65	65	Ana	Ortiz	Castillo	X	MJUAU1365991PP	1983-06-29	Balderas	
6	81	81	Roberto	Sanchez	Navarro	H	KLGO642551ER0	1983-11-02	Hidalgo	
7	84	84	Gloria	Torres	Pena	X	YBVD660757LQI	1973-12-24	Madero	
8	84	84	Gloria	Torres	Pena	X	YBVD660757LQI	1973-12-24	Madero	
9	100	100	Javier	Santos	Jimenez	X	HZLS1733628CV	1965-01-25	Balderas	
10	103	103	Mario	Gonzalez	Castro	X	APVO239741D80	1969-09-09	Paseo de la Reforr	

- XII) Obtener el nombre completo de los proveedores que provean carnes junto con el precio ordenado y ordenados de menor a mayor el precio.

```
SELECT nombre, paterno, materno, costo
FROM (
  SELECT idpersona
  FROM alimento
  WHERE tipoalim = 'carnes'
) AS a, proveedor AS b
WHERE a.idpersona = b.idpersona
order by costo;
```

	ABC nombre	ABC paterno	ABC materno	123 costo	
1	Rosa	Torres	Herrera	550.94000244	
2	Rafael	Martinez	Medina	594.78997803	
3	Alejandro	Santos	Cruz	790.45001221	
4	Manuel	Rivera	Castillo	893.96002197	
5	Manuel	Rivera	Castillo	893.96002197	
6	Teresa	Vargas	Chavez	918.7199707	
7	Leticia	Hernandez	Moreno	1,276.41003418	
8	Hector	Torres	Diaz	1,476.80004883	
9	Silvia	Reyes	Silva	1,507.81005859	
10	Daniel	Reyes	Ramirez	1,552.9699707	
11	Veronica	Ramirez	Sanchez	1,648.48999023	
12	Sergio	Sanchez	Torres	1,784.67004395	

XIII) Obtener todos los animales hembras del aviario

```
select *  
from (  
  SELECT a.idanimal  
  FROM animal AS a,  
        (SELECT idanimal  
         FROM jaula  
         where idbioma = 5) AS b  
  where a.sexo = 'H' AND a.idanimal = b.idanimal  
) AS a,  
animal AS b  
WHERE a.idanimal = b.idanimal;
```

	123 idanimal	123 idanimal	ABC nombre	ABC especie	ABC alimentacion	ABC sexo	123 p
1	22	22	Tucan	Ramphastidae	Frugivoro	H	
2	24	24	Colibri de garganta rubi	Archilochus colubris	Nectarivoro	H	
3	26	26	aguila calva	Haliaeetus leucocephalus	Carnivoro	H	6
4	28	28	Flamenco rosa	Phoenicopterus roseus	Omnivoro	H	
5	30	30	Albatros errante	Diomedea exulans	Piscivoro	H	
6	32	32	Gavilan	Accipiter gentilis	Carnivoro	H	
7	34	34	Ave del Paraíso	Paradisaeidae	Frugivoro	H	
8	36	36	Cardenal norteno	Cardinalis cardinalis	Granivoro	H	
9	38	38	Gorrion	Passeridae	Omnivoro	H	
10	40	40	Halcon peregrino	Falco peregrinus	Carnivoro	H	

- XIV) Obtener toda la información del cliente menos el genero con sus correos, donde se tiene dos o mas correos electrónicos ordenados por id y de forma ascendente.

```
select c.*, cc.correo
from
    (select
        idpersona,
        nombre,
        paterno,
        materno
    from cliente
    ) as c,
    correocliente as cc,
    (select idpersona
    from correocliente
    group by idpersona
    having count(distinct correo) >= 2
    ) as ccn
where
    (ccn.idpersona = cc.idpersona and
    cc.idpersona = c.idpersona)
order by idpersona asc;
```

	123 idpersona ▼	ABC nombre ▼	ABC paterno ▼	ABC materno ▼	ABC correo ▼	
1	1	Pedro	Castillo	Flores	cliente1@gmail.com	
2	1	Pedro	Castillo	Flores	cliente11@gmail.com	
3	2	Sergio	Jimenez	Morales	cliente22@hotmail.com	
4	2	Sergio	Jimenez	Morales	cliente2@hotmail.com	
5	3	Daniel	Gonzalez	Torres	cliente333@yahoo.com	
6	3	Daniel	Gonzalez	Torres	cliente33@yahoo.com	
7	3	Daniel	Gonzalez	Torres	cliente3@yahoo.com	
8	4	Jorge	Sanchez	Reyes	cliente4@outlook.com	
9	4	Jorge	Sanchez	Reyes	cliente44@outlook.com	
10	4	Jorge	Sanchez	Reyes	cliente4444@outlook.com	
11	4	Jorge	Sanchez	Reyes	cliente444@outlook.com	
12	5	Manuel	Romero	Jimenez	cliente5@gmail.com	

- XV) Obtener toda la información de los alimentos que van a caducar a partir de esta día en transcurso al fin del año 2023, además obtener la información del idpersona, su nombre completo y rfc que entrego ese alimento.

```
select *
from
    (select
        idpersona,
        nombre,
        paterno,
        materno,
        rfc
    from
        proveedor
    ) as pp,
    (select *
    from alimento
    where fechacaducidad >= current_date and
    fechacaducidad <= '2023-12-30'
    ) as a
where pp.idpersona = a.idpersona;
```

	123 idpersona	ABC nombre	ABC paterno	ABC materno	ABC rfc	123 idinsumo	123 idpersona	ABC nombre	fechacaducidad	123
1	3	Sergio	Sanchez	Torres	WBCR34647559Q	103	3	Pollo	2023-12-05	
2	3	Sergio	Sanchez	Torres	WBCR34647559Q	3	3	Pollo	2023-12-05	
3	6	Luis	Santos	Lopez	SHXX3442595TB	106	6	Platano	2023-12-25	
4	6	Luis	Santos	Lopez	SHXX3442595TB	6	6	Sandia	2023-12-25	
5	10	Gerardo	Vargas	Morales	UBRL96262154Q	110	10	Cañamo	2023-12-10	
6	10	Gerardo	Vargas	Morales	UBRL96262154Q	10	10	Alpiste	2023-12-10	
7	11	Francisco	Rodriguez	Alvarez	LKVZ1468877ZV	11	11	Pescado	2023-12-05	
8	23	Javier	Ramirez	Cabrera	INH550937085	23	23	Heno	2023-12-20	
9	26	Jose	Santos	Jimenez	DUAD299013M56	26	26	Cañamo	2023-12-15	
10	27	Mario	Ramirez	Flores	ZMLD245526362	27	27	Larvas	2023-12-25	
11	28	Silvia	Reyes	Alvarez	FKYW8214803PJ	28	28	Pienso	2023-12-10	

4. Obtención de datos calculados

Con la siguiente, podemos obtener el numero de jaulas, cuidadores, veterinarios o animales presentes en el zoológico

```
/**
 * Funcion para poder obtener el numero de jaulas, cuidadores, veterinarios o animales.
 * @param consulta := 'jaulas' || 'cuidadores' || 'veterinarios' || 'animales'.
 */
CREATE OR REPLACE FUNCTION obtenerCalculados(consulta TEXT)
RETURNS TEXT AS $$
DECLARE
    contador INTEGER;
BEGIN
    IF consulta = 'jaulas' THEN
        SELECT COUNT(*) INTO contador from jaula;
    ELSIF consulta = 'cuidadores' THEN
        SELECT COUNT(*) INTO contador from cuidador;
    ELSIF consulta = 'veterinarios' THEN
        SELECT COUNT(*) INTO contador from veterinario;
    ELSIF consulta = 'animales' THEN
        SELECT COUNT(*) INTO contador from animal;
    END IF;
    RETURN 'Numero de ' || consulta || ' = ' || contador::TEXT;
END;
$$ LANGUAGE plpgsql;

-- Sirve para llamar y obtener el numero de jaulas.
SELECT obtenerCalculados('jaulas');

-- Sirve para llamar y obtener el numero de cuidadores.
SELECT obtenerCalculados('cuidadores');

-- Sirve para llamar y obtener el numero de veterinarios.
SELECT obtenerCalculados('veterinarios');

-- Sirve para llamar y obtener el numero de animales.
SELECT obtenerCalculados('animales');
```

5. Procedimientos Almacenados

- ◆ El siguiente SP se encarga de registrar un cliente, para hacer uso de este SP es necesario introducir la información del cliente en cuestión para que el SP se encargue de insertarlo en la tabla correspondiente, es importante decir que el SP no permite la inserción de números o símbolos cuando sean campos relacionados a nombres.

```
CREATE OR REPLACE FUNCTION registrar_cliente(  
    p_idPersona BIGINT,  
    p_nombre VARCHAR(50),  
    p_paterno VARCHAR(50),  
    p_materno VARCHAR(50),  
    p_genero CHAR(1)  
) RETURNS VOID AS $$  
BEGIN  
    -- Verificar que nombre, paterno y materno no contengan números o símbolos  
    IF p_nombre ~ '[^a-zA-Z]' OR p_paterno ~ '[^a-zA-Z]' OR p_materno ~ '[^a-zA-Z]'  
    THEN RAISE EXCEPTION 'Los campos nombre, paterno y materno no deben  
        contener números o símb';  
    END IF;  
  
    -- Insertar los datos en la tabla cliente  
    INSERT INTO cliente(idPersona, nombre, paterno, materno, genero)  
    VALUES (p_idPersona, p_nombre, p_paterno, p_materno, p_genero);  
END;  
$$ LANGUAGE plpgsql;
```

- ◆ El siguiente SP se encarga de registrar un nuevo alimento de algún proveedor.

```
CREATE OR REPLACE FUNCTION registrar_alimento(  
    p_idInsumo bigint,  
    p_idPersona bigint,  
    p_nombre varchar(50),  
    p_fechaCaducidad date,  
    p_cantidad int4,  
    p_refrigeracion bool,  
    p_tipoAlim varchar(50)  
) RETURNS VOID AS $$  
BEGIN  
    if (SELECT NOT EXISTS (select p_idPersona from proveedor  
        where idPersona = p_idPersona)) THEN  
        RAISE EXCEPTION 'No existe el idpersona en la tabla proveedor';  
    END IF;  
  
    -- Insertar los datos en la tabla alimento  
    INSERT INTO alimento(idInsumo, idPersona, nombre, fechaCaducidad, cantidad,  
        refrigeracion, tipoAlim)  
    VALUES (p_idinsumo, p_idpersona, p_nombre, p_fechaCaducidad, p_cantidad,  
        p_refrigeracion, p_tipoalim);  
END;  
$$ LANGUAGE plpgsql;
```

- ◆ El siguiente SP se encarga de eliminar un animal.

```
CREATE OR REPLACE FUNCTION eliminar_animal(  
    p_idanimal bigint  
) RETURNS VOID AS $$  
BEGIN  
    if (SELECT NOT EXISTS (select p_idanimal from animal  
        where animal.idanimal = idanimal)) THEN  
        RAISE EXCEPTION 'No existe el animal';  
    END IF;  
  
    -- Insertar los datos en la tabla alimento  
    DELETE FROM animal where idanimal = animal;  
END;  
$$ LANGUAGE plpgsql;
```

6. Funciones

- ◆ Función que regresa la información de un cliente en base a su id.

```
CREATE OR REPLACE FUNCTION obtener_cliente(p_idPersona BIGINT)  
RETURNS TABLE(  
    idPersona BIGINT,  
    nombre VARCHAR(50),  
    paterno VARCHAR(50),  
    materno VARCHAR(50),  
    genero CHAR(1)  
) AS $$  
BEGIN  
    RETURN QUERY  
    SELECT cliente.idPersona, cliente.nombre, cliente.paterno,  
        cliente.materno, cliente.genero  
    FROM cliente  
    WHERE cliente.idPersona = p_idPersona;  
END;  
$$ LANGUAGE plpgsql;
```

- ◆ Función que verifica que un veterinario puede trabajar en a lo mas dos biomas.

```
create or replace function noBiomasTrabajados()  
returns trigger as $$  
declare  
    contador INTEGER;  
begin  
    select count(*) into contador from laborar where (idpersona = new.idpersona);  
    if contador >= 2 then  
        if TG_OP = 'INSERT' then  
            raise exception 'El veterinario/a a INSERTAR no puede trabajar  
                en mas de 2 biomas.  
                Ha alcanzado el limite de biomas trabajados.';  
        elsif TG_OP = 'UPDATE' then
```



```
        raise exception 'El veterinario/a a ACTUALIZAR ya tiene
asignado dos biomas.
No se pueden actualizar los valores.';
    end if;
    return null;
end if;
return new;
end;
$$ language plpgsql;
```

- ◆ Función para verificar que un animal solo puede estar en una jaula. Nota: Ya se cumple con que pertenezca a un solo bioma, además se cumple con que tanto el cuidador debe de ser de laborar en el mismo bioma que el animal.

```
create or replace function checarAnimalEnJaula()
returns trigger as $$
declare
    contador INTEGER;
    idCuidadorBioma INTEGER;
    idAnimalBioma INTEGER;
begin
    select c.idbioma into idCuidadorBioma
    from
        animal as a,
        cuidador as c
    where (
        new.idanimal = a.idanimal and
        a.idpersona = c.idpersona
    );
    idAnimalBioma := new.idbioma;
    select count(*) into contador from jaula where (idanimal = new.idanimal);
    if contador >= 1 then
        if TG_OP = 'INSERT' then
            raise exception 'El animal en la jaula a INSERTAR ya esta en
una jaula, solo esta permitido que un animal este en una jaula
y en un Bioma.';
        elsif TG_OP = 'UPDATE' then
            raise exception 'El animal de la jaula a ACTUALIZAR ya cuenta
con una jaula, no esta permitido que un animal este en
mas de una Jaula';
        end if;
        return null;
    end if;
    if idCuidadorBioma <> idAnimalBioma then
        raise exception 'No se tiene permitido que un cuidador que
es de otro bioma, este a cargo de este animal.';
        return null;
    end if;
    return new;
end;
$$ language plpgsql;
```

- ◆ Función para verificar que un animal solo puede ser atendido por veterinarios del mismo bioma.

```
create or replace function checarVeterinarioAtender()
returns trigger as $$
declare
    idAnimalBioma INTEGER;
begin
    select jaula.idbioma into idAnimalBioma from jaula
    where (idanimal = new.idanimal);
    if (SELECT NOT EXISTS
        (select l.idbioma
         from laborar as l
         where new.idpersona = l.idpersona AND idAnimalBioma = l.idbioma)) THEN
        raise exception 'No se tiene permitido que un veterinario atienda a un
        animal de otro bioma al asignado.';
        return null;
    end if;
    return new;
end;
$$ language plpgsql;
```

7. Disparadores

- ◆ Trigger para verificar que un veterinario a lo más puede trabajar en dos biomas.

```
create trigger verificarVeterinarios
before insert or update on laborar
for each row
execute function noBiomassTrabajados();
```

- ◆ Trigger para verificar que un animal solo puede estar en una jaula y un bioma.

```
create trigger verificarAnimal
before insert or update on jaula
for each row
execute function checarAnimalEnJaula();
```

- ◆ Trigger para verificar que un veterinario solo puede atender a animales del mismo bioma.

```
create trigger verificarAtender
before insert or update on atender
for each row
execute function checarVeterinarioAtender();
```

8. Poblamiento de la Base de Datos

Para generar los datos necesarios para nuestro proyecto, empleamos una eficiente y gratuita herramienta en línea llamada Mockaroo, esta pagina nos ayudo a poder generar los datos de manera rápida y sencilla, de tal forma que podamos poblar nuestra base de datos.

El proceso de utilización de Mockaroo se llevó a cabo de la siguiente manera: primero, creamos un nuevo proyecto en la plataforma, después, procedimos a definir los campos que deseábamos generar para nuestra base de datos, para cada campo, tuvimos la flexibilidad de especificar el tipo de datos que queríamos, el formato en el que debían presentarse, la longitud que debían tener y los valores que debían ser generados en cada entrada.

Una vez que agregamos todos los campos con sus respectivas configuraciones, Mockaroo nos permitió generar el archivo de datos completo. Este archivo, que contenía toda la información que habíamos diseñado, estaba listo para su descarga.

En resumen, Mockaroo se reveló como una herramienta esencial que nos facilitó la generación de datos personalizados para nuestra aplicación. Gracias a su versatilidad y facilidad de uso, logramos obtener información sintética de alta calidad que enriqueció nuestra base de datos.



9. Archivos presentes en este proyecto

