



# Trabajo tutelado: Diseño de una base de datos.

---

## INTRODUCCIÓN A LAS BASES DE DATOS.

Agustín García Doñate | Pedro Sánchez García.

MÁSTER UNIVERSITARIO EN BIOINFORMÁTICA PARA CIENCIAS DE LA SALUD.  
PROFESORA: DRA. NIEVES RODRIGUEZ BRISABOA.

## ÍNDICE DE CONTENIDO

ENUNCIADO.....	2
OBJETIVOS.....	2
DISCUSIÓN DEL ENUNCIADO .....	3
MODELO E-R .....	4
MAPEADO E-R A RELACIONAL .....	5
NORMALIZACIÓN DEL DISEÑO DEL ESQUEMA RELACIONAL .....	5
MODELO RELACIONAL DE TRABAJO.....	14
CONSULTAS EN SQL.....	14
CONCLUSIONES.....	18
BIBLIOGRAFÍA.....	18
ANEXO I.....	19

## ENUNCIADO

Supongamos que un antiguo alumno del MUBICS ha fundado la empresa BioinFarm, que consiste en una iniciativa dedicada a la genómica y bioinformática en el sector agroalimentario. Recientemente, la Xunta de Galicia convoca unas ayudas económicas para impulsar los proyectos de este tipo de iniciativas empresariales. Sin embargo, la burocracia implica exponer la estructura de la empresa, organización en horas de trabajo y justificar una serie de gastos en material realizados. De esta forma, el fundador de BioinFarm opta por contactar con un amigo informático para que diseñe una base de datos que se ajuste a los aspectos mencionados y se pueda presentar ante el comité encargado de adjudicar las ayudas.

BioinFarm es una empresa de ámbito nacional, de forma que presenta laboratorios en numerosas ciudades de España asignados a una determinada sede correspondiente. Cada laboratorio cuenta con un equipamiento que comprende material informático y equipos de secuenciación, empleados en los servicios ofrecidos, así como en proyectos de investigación. Con respecto al personal, cabe destacar la existencia de diferentes rangos, los cuales condicionan el salario asignado a cada empleado y empleada.

## OBJETIVOS

El presente proyecto tiene como objetivos principales:

- I. Plantear un diseño de base de datos que satisfaga las demandas del cliente.
- II. Emplear la bibliografía recomendada en la asignatura para complementar las explicaciones ofrecidas en clase y justificar las decisiones ante los problemas que se plantean en el proceso de diseño.
- III. Formular una serie de consultas aplicadas a la base de datos con lenguaje SQL mediante PostgreSQL.

## DISCUSIÓN DEL ENUNCIADO

A partir del problema planteado, en primer lugar, es preciso tener en cuenta una serie de aspectos que serán de utilidad para el planteamiento de las entidades y del modelo E-R inicial para la base de datos:

- Para las **sedes**, se desea conocer el polígono industrial en el que se ubica, el número de parcela y área que comprende, precio pagado por la sede e impuestos anuales abonados.
- Dado que es necesario mostrar el **equipamiento** de cada laboratorio, se debe tener constancia de la marca, modelo, precio y generación del material informático y equipos de secuenciación.
- Con el fin de valorar la evolución de la empresa, el comité solicita conocer, con respecto a los **laboratorios**, su fecha de alta. En lo que se refiere a los **servicios**, se debe indicar un nombre, el tipo de muestras de trabajo, fecha de inicio y fin, así como los gastos que han implicado a la empresa.
- Por su parte, de los **proyectos** se deberá reflejar el presupuesto acordado por el fundador de la empresa con los empleados y empleadas implicados en estos.
- El otro aspecto que evaluará el comité es la organización del **personal** en la empresa, con información asociada como DNI, dirección, provincia, Código Postal, rango y salario asignado.

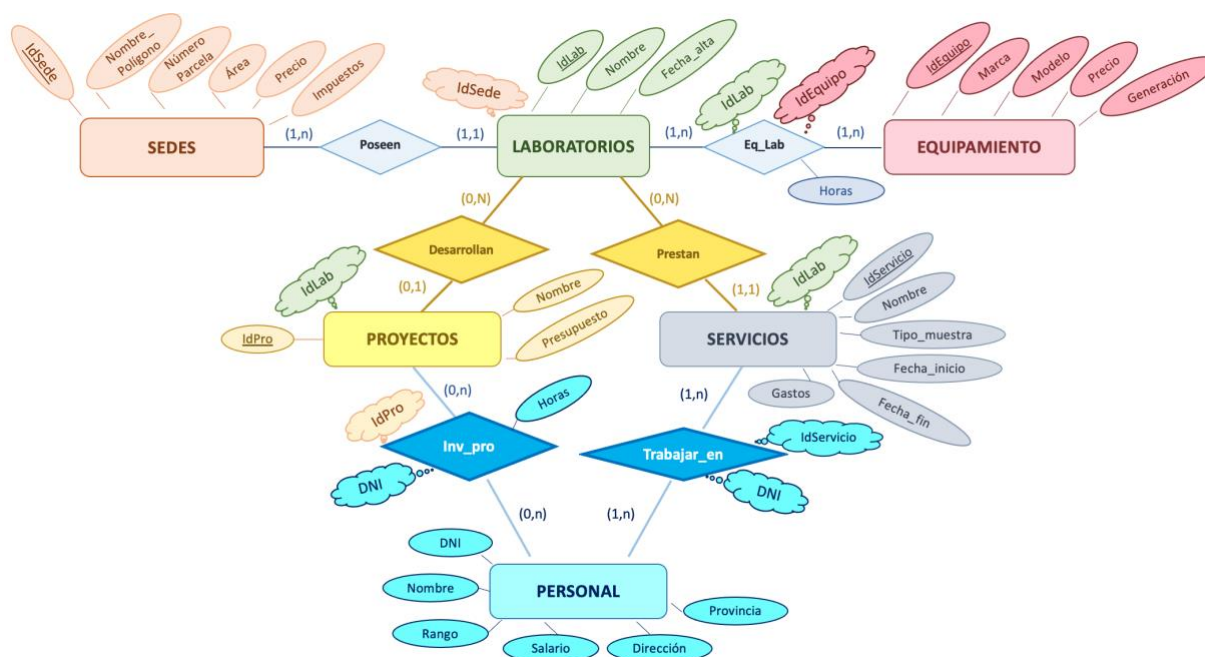
Por otra parte, se deben tener en cuenta las siguientes cuestiones, que serán relevantes en la generación del modelo entidad-relación (E-R):

- Cada sede posee uno o varios laboratorios, mientras que cada laboratorio está asignado a una única sede.
- Cada laboratorio puede tener varios equipos. Además, estos pueden compartirse entre varios laboratorios.
- En un determinado laboratorio se pueden llevar a cabo varios proyectos de investigación, pero estos pertenecen a un único laboratorio.
- Los proyectos pueden implicar a varios empleados y empleadas, de forma que, además, un empleado o empleada puede estar trabajando para varios proyectos. Esta interrelación es igual para personal y servicios ofrecidos por la empresa.
- Los laboratorios pueden realizar varios servicios, pero estos se asignan a un único laboratorio.

## MODELO E-R

Un diagrama entidad-relación (E-R) se describe como una forma de modelo de objeto similar en muchas formas a un modelo de objeto semántico. Presenta entidades, las cuales representan una instancia específica, que puede ser una cosa física o una abstracción. Las entidades similares se agruparán en conjuntos de entidades. Al igual que un objeto semántico, las entidades poseen atributos que describen a los objetos que estas representan. El otro componente en este diagrama consiste son las relaciones existentes entre las entidades, que constan de una cardinalidad, es decir, el número máximo de instancias de relación en las que una entidad puede participar. Por ejemplo, en una relación binaria con una razón de cardinalidad de 1.N significaría en una entidad se relaciona con cualquier número de entidades (N). Debemos tener en cuenta que las posibles razones de cardinalidad para los tipos de relación binaria son 1.1, 1.N, N.1 y N.N (Stephens, 2009).

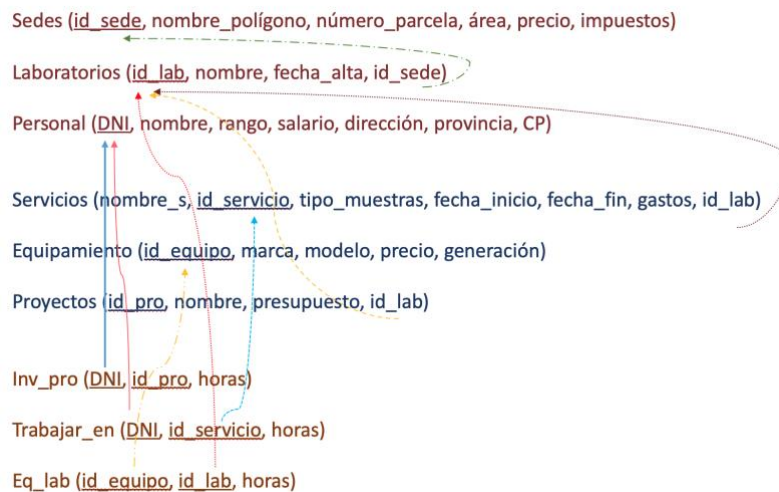
Para la construcción progresiva de este modelo, se sigue una metodología en la que se valoran las decisiones sobre las tablas que se precisan y los atributos que han de estar presentes en cada una de ellas. Para el presente proyecto, se distinguen las entidades, cada una de ellas con atributos mencionados en el apartado anterior y las relaciones existentes mostradas en la siguiente figura:



Cabe destacar de las **interrelaciones** existentes entre las entidades, que son de tipo 1.N en el caso de laboratorios-sedes, laboratorios-servicios y laboratorios-proyectos. Esto se debe a que una sede puede tener varios laboratorios, pero estos se corresponden con una única sede. Además, del mismo modo, los laboratorios pueden tener varios proyectos y servicios, pero en ambos casos se corresponden con un único laboratorio. En todos estos casos, se debe tener presente que la llave primaria de la entidad en lado n pasa al lado 1 como llave foránea. Por otra parte, se distinguen interrelaciones que son N.N entre las entidades de laboratorios y equipamiento, proyectos y personal, así como entre personal y servicios, que dan lugar a las tablas Eqp\_Lab, Inv\_pro y Trabajar\_en respectivamente. En este caso, las tablas presentarán las llaves primarias de ambas entidades, junto con otro/s atributo/s.

## MAPEADO E-R A RELACIONAL

El análisis de las relaciones mostrado en el apartado anterior nos conduce al planteamiento del esquema de la base de datos relacional mostrado en la siguiente figura, donde se reflejan las tablas de las entidades con sus correspondientes atributos, llaves primarias y llaves foráneas:



## NORMALIZACIÓN DEL DISEÑO DEL ESQUEMA RELACIONAL

### DEPENDENCIAS FUNCIONALES Y LLAVES CANDIDATAS

Una vez planteado el esquema relacional, el diseñador de la base de datos ha de evaluar una serie de aspectos que se definirán formalmente. En primer lugar, en cada esquema de las entidades de la base de datos, se analizarán las **dependencias funcionales**. Estas, denotadas por  $X \rightarrow Y$ , entre dos conjuntos de atributos  $X$  e  $Y$  que son subconjuntos de  $R$ , especifica una restricción entre las posibles tuplas que pueden formar un estado de relación  $r$  de  $R$ . Por tanto, la restricción dice que dos tuplas  $t_1$  y  $t_2$  en  $r$  cumplen que  $t_1[X] = t_2[X]$ , deben cumplir también que  $t_1[Y] = t_2[Y]$  (Elmasri & Navathe, 2008). Además, se debe tener en cuenta que si una restricción de  $R$  indica que no puede haber más de una tupla con un valor  $x$  concreto en cualquier instancia de relación  $r(R)$ , se dice que  $X$  es una **llave candidata** de  $R$ , cumpliendo que  $X \rightarrow Y$  para cualquier subconjunto de atributos  $Y$  de  $R$ . Por tanto, desde el punto de vista de la semántica de los atributos, se mantienen unas dependencias funcionales que conformarán el **conjunto  $F$  de dependencias funcionales**. Mediante las siguientes reglas, denominadas axiomas de Armstrong, se pueden inferir dependencias funcionales adicionales posteriormente:

1. Reflexividad:  $y \subseteq x \Rightarrow x \rightarrow y$
2. Aumentatividad:  $x \rightarrow y \Rightarrow xz \rightarrow yz$
3. Transitividad:  $x \rightarrow y, y \rightarrow z \Rightarrow x \rightarrow z$
4. Unión:  $x \rightarrow y \cap x \rightarrow z \Rightarrow x \rightarrow yz$
5. Pseudotransitividad:  $x \rightarrow y \cap wy \rightarrow z \Rightarrow wx \rightarrow z$
6. Descomposición:  $x \rightarrow y, z \subseteq y \Rightarrow x \rightarrow z$

En base a los axiomas de Armstrong y al conjunto  $F$  de dependencias funcionales, se calcular el **cierre transitivo de cada descriptor**, es decir, el conjunto total de atributos que viene determinado por el descriptor en cuestión, de forma que se analizan las llaves candidatas. De forma arbitraria, se elegirá una de ellas como **llave principal**.

## DESCOMPOSICIÓN DE LOS ESQUEMAS DE RELACIÓN

Una vez estudiadas las dependencias funcionales y ciertas propiedades, se procede al proceso de **normalización del diseño del esquema relacional**. La esencia de este proceso es evaluar las relaciones de idoneidad y descomponerlas todo lo necesario para alcanzar formas normales elevadas. Se define la normalización de datos como un análisis del esquema de relación, basado en las dependencias funcionales y llaves principales para **minimizar redundancia y anomalías en inserción, borrado y actualización** existentes, tal y como se explicará más adelante.

De esta forma, aquellos esquemas de relación que no cumplan ciertas condiciones se descomponen en esquemas de relación más pequeños que cumplen esas condiciones y propiedades. Para un buen diseño de base de datos, además, se debe **mantener la propiedad de reunión sin pérdida**, donde la unión de los atributos de todas las tablas es igual a la de partida. Además, para que **no se produzca pérdida en información**, la descomposición de  $R$  en  $R_1$  y  $R_2$  ha de tener la llave de  $R_1$  o de  $R_2$  como atributos en común.

Para determinar la forma normal en la que se encuentra una relación  $R$ , es preciso tener en cuenta el concepto de **atributo primo**, el cual es miembro de alguna de las llaves candidatas de  $R$ . Por su parte, el **atributo no primo** se trata de aquel que no es miembro de ninguna llave candidata.

Si todos los atributos de una relación  $R$  son mono valuados, esta relación se encuentra en **primera forma normal (1FN)**. Si está en primera forma normal y además, todos los atributos no primos dependen de la totalidad de las llaves candidatas, entonces la relación está en **segunda forma normal (2FN)**. En lo que respecta a la **tercera forma normal (3FN)**, esta se cumple cuando está en 2FN y no se producen dependencias transitivas de ningún atributo no primo con respecto a alguna llave candidata.

Además, se distingue una forma normal superior, denominada **forma normal de Boyce-Codd (FNBC)**, más simple que la 3FN, pero más estricta. Una tabla estará en FNBC si se encuentra en 3FN y todos los determinantes presentes en el conjunto  $F$  de dependencias funcionales son llaves candidatas (Stephens, 2009).

A continuación, se muestra el proceso de normalización en cada una de las tablas del caso propuesto en este trabajo, donde se recoge una breve descripción de la información contenida, el conjunto de dependencias funcionales, las llaves candidatas y la decisión que tomaría el diseñador de la base de datos.

**Descripción:** Esta tabla recoge la información relativa a las sedes, con su identificador, nombre del polígono industrial, número de parcela correspondiente, área que comprende la ubicación, precio pagado por la sede y los impuestos abonados.

Sedes (id\_sede, nombre\_polígono, número\_parcela, área, precio, impuestos)

$F = \{id\_sede \rightarrow nombre\_polígono, número\_parcela, área, precio, impuestos\}$  (1)

$nombre\_polígono, número\_parcela \rightarrow id\_sede, área, precio, impuestos$  (2)

$nombre\_polígono \rightarrow impuestos$  (3)

$área \rightarrow precio$  (4)}

Llaves candidatas: id\_sede, (nombre\_polígono, número\_parcela)

Atributos primos: id\_sede, (nombre\_polígono, número\_parcela)

Atributos no primos: área, precio, impuestos

**Explicación:** Se encuentra en 1FN, ya que debido a la df (2) se viola la definición general de 2FN, donde impuestos, atributo no primo, es parcialmente dependiente de la llave candidata (nombre\_polígono, número\_parcela).

**Descomposición de Sedes:** Se toma como punto de partida la df (3) siguiendo la heurística de que impuestos no aparece a la izquierda de ninguna otra df.

Sedes2 (nombre\_polígono, impuestos)

$F = \{nombre\_polígono \rightarrow impuestos\}$

Llaves candidatas: nombre\_polígono

Atributos primos: nombre\_polígono

Atributos no primos: impuestos

**Explicación:** Se encuentra en FNBC, ya que impuestos (atributo no primo) depende de la totalidad de las llaves candidatas y lo hace de forma directa, no transitivamente. Además, todos los determinantes (en este caso nombre\_polígono) cumplen que son llaves candidatas.

Sedes1 (id\_sede, nombre\_polígono, número\_parcela, área, precio)

$F = \{id\_sede \rightarrow nombre\_polígono, número\_parcela, área, precio\}$  (1)

$nombre\_polígono, número\_parcela \rightarrow id\_sede, área, precio$  (2)

$área \rightarrow precio$  (3)}

Llaves candidatas: id\_sede, (nombre\_polígono, número\_parcela)

Atributos primos: id\_sede, (nombre\_polígono, número\_parcela)

Atributos no primos: área, precio

**Explicación:** Se encuentra en 2FN, ya que se produce dependencia transitiva de precio (atributo no primo) con cada una de las llaves candidatas a través de área, otro atributo no primo.



Descomposición de Sedes1: Se construye Sedes1A eliminando el atributo precios de Sedes1, que viola la 3FN y se coloca en Sedes1B junto con área (lado izquierdo de la df (3)), que provoca la dependencia transitiva.

Sedes1A (id\_sede, nombre\_polígono, número\_parcela, área)

$F = \{id\_sede \rightarrow nombre\_polígono, número\_parcela, área\}$  (1)

$nombre\_polígono, número\_parcela \rightarrow id\_sede, área$  (2)

Llaves candidatas: id\_sede, (nombre\_polígono, número\_parcela)

Atributos primos: id\_sede, (nombre\_polígono, número\_parcela)

Atributos no primos: área

**Explicación:** Se encuentra en FNBC, ya que todos los determinantes son llaves candidatas y el área (atributo no primo) depende directamente de la llaves candidatas.

Sedes1B (área, precio)

$F = \{área \rightarrow precio\}$

Llaves candidatas: área

Atributos primos: área

Atributos no primos: precio

**Explicación:** Se encuentra en FNBC, donde todos los determinantes son llaves candidatas y el precio (atributo no primo) depende directamente de la llave candidata.

## PERSONAL

---

Descripción: Esta tabla recoge toda la información relativa a los trabajadores y trabajadoras de BioinFarm, con el DNI, nombre, rango que presenta en la empresa, salario asignado, dirección, provincia y código postal.

Personal (DNI, nombre, rango, salario, dirección, provincia, CP)

$F = \{DNI \rightarrow nombre, rango, dirección, CP\}$  (1)

$dirección \rightarrow CP$  (2)

$rango \rightarrow salario$  (3)

$CP \rightarrow provincia$  (4)

Llaves candidatas: DNI

Atributos primos: DNI

Atributos no primos: nombre, rango, salario, dirección, provincia, CP

**Explicación:** Se encuentra en 2FN, ya que los atributos no primos dependen de la totalidad de la llave candidata. Sin embargo, se producen dependencias transitivas entre los atributos no primos

de salario y provincia con la llave candidata a través de los atributos no primos de rango y CP respectivamente, lo que impide la 3FN.

**Descomposición de Personal:** Se construyen otras tablas con las df (3) y df(4) que violan la 3FN.

Salario (rango, salario)

$F = \{\text{rango} \rightarrow \text{salario}\}$

Llaves candidatas: rango

Atributos primos: rango

Atributos no primos: salario

Explicación: Se encuentra en FNBC, con el salario (atributo no primo) que depende de la totalidad de las llaves candidatas de manera directa. Además, todos los determinantes (en este caso único) cumple la condición de ser llave candidata en el conjunto  $F$ .

CP (CP, provincia)

$F = \{\text{CP} \rightarrow \text{provincia}\}$

Llaves candidatas: CP

Atributos primos: CP

Atributos no primos: provincia

Explicación: Se encuentra en FNBC, con provincia (atributo no primo) que depende de la totalidad de las llaves candidatas de forma directa. Además, todos los determinantes (en este caso único) también cumple la condición de ser llave candidata en el conjunto  $F$ .

**Mantenimiento de Personal en 2FN:** Cabe destacar que, en este caso, el diseñador de la base de datos debe valorar si realizar o no la descomposición. Es cierto que, en el caso del salario, para hacer cambios con el salario del rango de trabajadores y trabajadoras, estos se harían en la tabla Salario sin producirse problemas de actualización como sucedería con la de Personal. En el caso de eliminar a un único trabajador/ra de un rango, tampoco se generarían problemas de borrado de toda la información asociada como en el caso de Personal.

A pesar de estos aspectos, en lo que respecta al código postal (CP), si estuviésemos interesados en realizar una consulta sobre provincias, sería preciso plantear una operación de join entre las tablas de Personal y CP. Esto implica una reducción en el rendimiento del sistema realmente que se puede evitar debido a que hay una probabilidad muy reducida, en realidad, de cambios en el código postal. En consecuencia, se opta por mantener la tabla de Personal en 2FN.

## SERVICIOS

---

**Descripción:** En esta tabla se almacena información relativa a los servicios ofrecidos por BioinFarm con un identificador, nombre, tipo de muestras de trabajo, fechas de inicio y fin del servicio, gastos que ha implicado a la empresa y, además, el identificador del laboratorio donde se ha llevado a cabo el servicio, que se trata de una llave foránea que referencia a la tabla de laboratorios.

Servicios (nombre\_s, id\_servicio, tipo\_muestras, fecha\_inicio, fecha\_fin, gastos, id\_lab)

$F = \{id\_servicio \rightarrow nombre\_s, tipo\_muestras, fecha\_inicio, id\_lab\} \quad (1)$

$fecha\_inicio \rightarrow fecha\_fin, gastos, id\_lab \quad (2)$

Llaves candidatas: id\_servicio

Atributos primos: id\_servicio

Atributos no primos: nombre\_s, tipo\_muestras, fecha\_inicio, fecha\_fin, gastos, id\_lab

**Explicación:** Se aprecia que fecha\_fin y gastos vienen determinados transitivamente por la llave candidata a través de fecha\_inicio (atributo no primo), de tal forma que se produce una violación de 3FN. Para pasar de 2FN a la 3FN y en consecuencia, a la FNBC, es preciso descomponer Servicios.

**Descomposición de Servicios:** Se construyen dos tablas S1 y S2 con las df (1) y df (2).

S1 (nombre\_s, id\_servicio, tipo\_muestras, fecha\_inicio, id\_lab)

$F = \{id\_servicio \rightarrow nombre\_s, tipo\_muestras, fecha\_inicio, id\_lab\}$

Llaves candidatas: id\_servicio

Atributos primos: id\_servicio

Atributos no primos: nombre\_s, tipo\_muestras, fecha\_inicio, id\_lab

**Explicación:** Se logra la 3FN y por tanto, FNBC, pues se cumple que todos los determinantes (en este caso único) es llave candidata en el conjunto  $F$ .

S2 (fecha\_inicio, fecha\_fin, gastos, id\_lab)

$F = \{fecha\_inicio \rightarrow fecha\_fin, gastos, id\_lab\}$

Llaves candidatas: fecha\_inicio

Atributos primos: fecha\_inicio

Atributos no primos: fecha\_fin, gastos, id\_lab

**Explicación:** Se alcanza la 3FN y por tanto, FNBC, donde todos los determinantes (en este caso único) es llave candidata en el conjunto  $F$ .

## EQUIPAMIENTO

---

**Descripción:** En esta tabla se almacena información relacionada con el equipamiento que posee BioinFarm en sus laboratorios. Contiene un identificador del equipo, marca, modelo, precio y la generación del modelo.

Equipamiento (id\_equipo, marca, modelo, precio, generación)

$F = \{ \text{id\_equipo} \rightarrow \text{modelo, precio, generación (1)}$   
 $\text{modelo} \rightarrow \text{marca, precio (2)} \}$

Llaves candidatas: id\_equipo

Atributos primos: id\_equipo

Atributos no primos: marca, modelo, precio, generación

**Explicación:** La tabla está en 2FN, pues los atributos no primos dependen de la totalidad de la llave candidata. Sin embargo, marca (atributo no primo) depende transitivamente de la llave candidata, por lo que se viola la 3FN.

**Descomposición de Equipamiento:** Se construyen dos tablas E1 y E2 con las df (1) y df (2).

E1 (id\_equipo, modelo, generación)

$F = \{ \text{id\_equipo} \rightarrow \text{modelo, generación} \}$

Llaves candidatas: id\_equipo

Atributos primos: id\_equipo

Atributos no primos: modelo, generación

**Explicación:** Se logra la 3FN y por tanto, FNBC, con todos los determinantes (en este caso único) como llave candidata en el conjunto  $F$ .

E2 (marca, modelo, precio)

$F = \{ \text{modelo} \rightarrow \text{marca, precio} \}$

Llaves candidatas: modelo

Atributos primos: modelo

Atributos no primos: marca, precio

**Explicación:** En este caso, también logra la 3FN y por tanto, FNBC, con todos los determinantes (en este caso único) como llave candidata en el conjunto  $F$ .

## LABORATORIOS

---

**Descripción:** Esta tabla contiene la información relacionada con los laboratorios de BioinFarm. Contiene un identificador del laboratorio, nombre identificativo, la fecha en la que se dio de alta y el identificador de la sede en la que se encuentra, que es llave foránea que referencia a la tabla de Sedes.

Equipamiento (id\_lab, nombre, fecha\_alta, id\_sede)

$F = \{id\_lab \rightarrow nombre, fecha\_alta, id\_sede\}$

Llaves candidatas: id\_lab

Atributos primos: id\_lab

Atributos no primos: nombre, fecha\_alta, id\_sede

**Explicación:** La tabla se encuentra en 3FN y por tanto, en FNBC, con todos los determinantes (en este caso único) como llave candidata en el conjunto  $F$ .

## PROYECTOS

---

**Descripción:** Esta tabla contiene la información relacionada con los proyectos en desarrollo de BioinFarm. Contiene un identificador del proyecto, nombre identificativo, el presupuesto estimado y acordado, así como el identificador del laboratorio en el que se lleva a cabo, que es llave foránea que referencia a la tabla de laboratorios.

Proyectos (id\_pro, nombre, presupuesto, id\_lab)

$F = \{id\_pro \rightarrow nombre, presupuesto, id\_lab\}$

Llaves candidatas: id\_pro

Atributos primos: id\_pro

Atributos no primos: nombre, presupuesto, id\_lab

**Explicación:** Está en 3FN y por tanto, en FNBC, pues todos los determinantes (en este caso único) son llave candidata en el conjunto  $F$ .

## INV\_PRO.

---

**Descripción:** Esta tabla almacena información relacionada con los trabajadores y trabajadoras de BioinFarm que se encuentran implicados en los proyectos en desarrollo. Contiene el DNI e identificador del proyecto como llaves foráneas que refieren a las tablas Personal y Proyectos respectivamente, así como las horas dedicadas al proyecto en cuestión de forma semanal para cada empleado.

Personal (DNI, id\_pro, horas)

$F = \{DNI, id\_pro \rightarrow horas\}$

Llaves candidatas: DNI, id\_pro

Atributos primos: DNI, id\_pro

Atributos no primos: horas

**Explicación:** Se encuentra en FNBC, ya que está en 3FN y todos los determinantes (en este caso único) son llave candidata en el conjunto  $F$ .

## TRABAJAR\_EN

---

**Descripción:** En esta tabla se almacena información relativa a los trabajadores y trabajadoras que llevan a cabo los servicios ofrecidos por BioinFarm. Contiene el DNI e identificador del servicio como llaves foráneas que referencian a las tablas Personal y Servicios respectivamente, así como las horas dedicadas al correspondiente servicio de forma semanal para cada empleado.

Trabajar\_en (DNI, id\_servicio, horas)

$F = \{ \text{DNI, id_servicio} \rightarrow \text{horas} \}$

Llaves candidatas: DNI, id\_servicio

Atributos primos: DNI, id\_servicio

Atributos no primos: horas

**Explicación:** Está en FNBC, ya que cumple estar en 3FN y todos los determinantes (en este caso único) son llave candidata en el conjunto  $F$ .

## EQ\_LAB

---

**Descripción:** Esta tabla contiene la información relacionada con los equipos de los laboratorios de BioinFarm, un identificador del laboratorio donde se ubican y las horas semanales de empleo.

Eq\_lab (id\_equipo, id\_lab, horas)

$F = \{ \text{id_equipo, id_lab} \rightarrow \text{horas} \}$

Llaves candidatas: id\_equipo, id\_lab

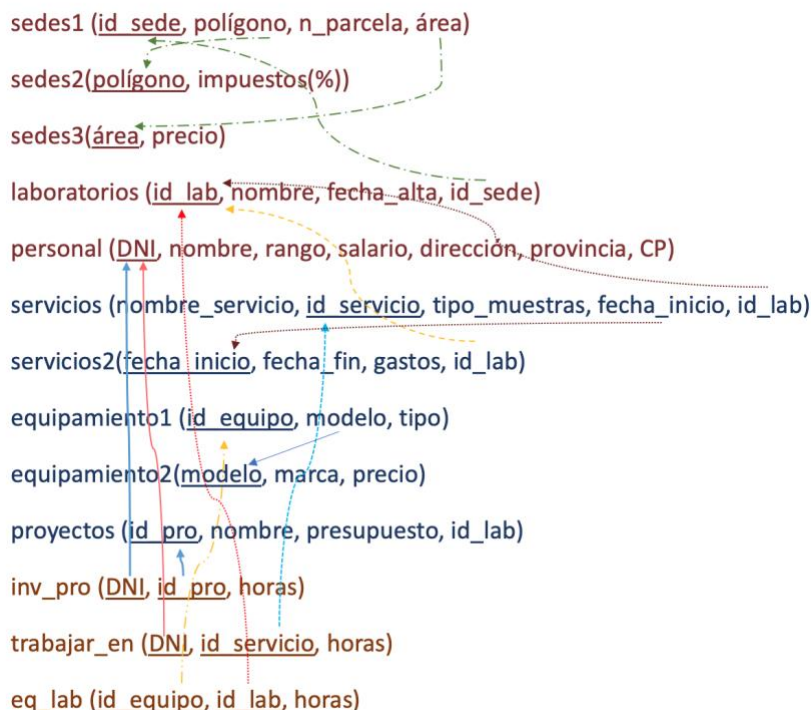
Atributos primos: id\_equipo, id\_lab

Atributos no primos: horas

**Explicación:** Se encuentra en FNBC, pues se logra la 3FN y todos los determinantes (en este caso único) son llave candidata en el conjunto  $F$ .

## MODELO RELACIONAL DE TRABAJO

El modelo relacional resultante de los análisis efectuados en el apartado anterior se refleja en la siguiente figura:



## CONSULTAS EN SQL

Se exponen las respectivas consultas efectuadas en contexto y las capturas que muestran el resultado en pgAdmin 4.

1. ¿Cuántas personas participan en cada proyecto? Muestra el nombre de proyecto.

```
select p.id_pro, t.nombre, count(a.DNI)
from personal a join inv_pro p on a.DNI=p.DNI join proyectos t on p.id_pro=t.id_pro
group by p.id_pro, t.nombre
```

	id_pro numeric (2)	nombre character varying (60)	count bigint
1	48	RECOGIDA DIVERSAS Y POTENCIALES ESPECIES	3
2	39	GENOTIPACIÓN NUEVAS ESPECIES	3
3	10	FORMAR SONDAS PCR 2	3
4	55	DISEÑO DE SONDAS PCR Y ANÁLISIS DE DATOS	3
5	9	SECUENCIAR VARIANTES DE ESPECIES DE INTERÉS	4
6	61	GENOTIPACIÓN ESPECIES DE INTERÉS	3
7	22	FORMAR SONDAS PCR 1	3
8	90	SECUENCIAR TODAS ESPECIES	4

2. Muestra los proyectos que serán viables a un año vista teniendo en cuenta el salario de los empleados y los gastos de cada proyecto sabiendo que el presupuesto para cada proyecto se incrementará en un 80% a lo largo del año.

```
select o.id_pro, p.nombre, s.gastos, p.presupuesto, sum(salario) as salario, (sum(salario)*12 +
s.gastos) as "gastos 1 año", (p.presupuesto*1.8 - (sum(salario)*12 + s.gastos)) as "dinero restante"
from servicios2 s join proyectos p on s.id_lab=p.id_lab join inv_pro o on p.id_pro=o.id_pro
join personal a on a.DNI=o.DNI
group by o.id_pro, p.nombre, s.gastos, p.presupuesto
```

	id_pro numeric (2)	nombre character varying (60)	gastos numeric (7,2)	presupuesto numeric (10,2)	salario numeric	gastos 1 año numeric	dinero restante numeric
1	9	SECUENCIAR VARIANTES DE ESPECIES DE INTERÉS	230.50	75000.00	9600.00	115430.50	19569.500
2	10	FORMAR SONDAS PCR 2	9000.00	70000.00	6600.00	88200.00	37800.000
3	22	FORMAR SONDAS PCR 1	5000.20	40000.00	7300.00	92600.20	-20600.200
4	39	GENOTIPARCIÓN NUEVAS ESPECIES	300.00	100000.00	6600.00	79500.00	100500.000
5	48	RECOGIDA DIVERSAS Y POTENCIALES ESPECIES	499.99	15000.00	6200.00	74899.99	-47899.990
6	55	DISEÑO DE SONDAS PCR Y ANÁLISIS DE DATOS	600.00	10000.00	8300.00	100200.00	-82200.000
7	61	GENOTIPACIÓN ESPECIES DE INTERÉS	2300.00	100000.00	6600.00	81500.00	98500.000
8	90	SECUENCIAR TODAS ESPECIES	3000.80	800000.00	9600.00	118200.80	1321799.200

3. ¿Cuántas personas viven en las avenidas más transitadas de Coruña? (Avda. Juan Flórez y Rda. Nelle)

```
select count(DNI)
from personal
where dirección LIKE 'Avda. Juan Flórez%' OR dirección like 'Rda. Nelle%'
```

	count bigint
1	7

4. ¿Qué materiales tiene cada laboratorio? Muestra también el modelo y el tipo de material.

```
select l.nombre as laboratorio, l.id_sede, l.id_lab, e.id_equipo, j.modelo, j.tipo
from laboratorios l join eq_lab e on l.id_lab=e.id_lab join equipamiento1 j on
j.id_equipo=e.id_equipo
group by l.id_lab, e.id_lab, e.id_equipo, j.modelo, j.tipo
order by l.id_lab, e.id_lab
```

	laboratorio character varying (14)	id_sede numeric (2)	id_lab numeric (2)	id_equipo numeric (3)	modelo character varying (14)	tipo character varying (30)
1	GENOTIPADO1	10	1	101	AX47	SECUENCIADOR
2	GENOTIPADO1	10	1	404	PIP44	MICROPIPETA
3	GENOTIPADO1	10	1	606	AG100	KIT EXTRACCIÓN DNA
4	GENOTIPADO1	10	1	707	PCR007	REACTIVOS PARA PCR
5	GENOTIPADO1	10	1	808	TT88	GRADILLAS
6	GENOTIPADO1	10	1	909	C09	[...] PACK TUBOS DE ENSAYO
7	SECUENCIACIÓN1	20	2	7	INF2	ORDENADOR ÚLTIMA GENERACIÓN
8	SECUENCIACIÓN1	20	2	202	MM55	SECUENCIADOR
9	SECUENCIACIÓN1	20	2	404	PIP44	MICROPIPETA
10	SECUENCIACIÓN1	20	2	606	AG100	KIT EXTRACCIÓN DNA
11	SECUENCIACIÓN1	20	2	707	PCR007	REACTIVOS PARA PCR
12	SECUENCIACIÓN1	20	2	808	TT88	GRADILLAS
13	SECUENCIACIÓN1	20	2	909	C09	[...] PACK TUBOS DE ENSAYO
14	HIBRIDACIÓN	30	3	303	AP99	TERMOCICLADOR
15	HIBRIDACIÓN	30	3	404	PIP44	MICROPIPETA
16	HIBRIDACIÓN	30	3	505	PIP56	MICROPIPETA
17	HIBRIDACIÓN	30	3	707	PCR007	REACTIVOS PARA PCR
18	HIBRIDACIÓN	30	3	808	TT88	GRADILLAS



5. A causa de un incendio ocurrido en septiembre de 2020, el laboratorio 1 tuvo que cerrar. ¿Hubo algún laboratorio que se abriera posterior a este que ofreciera el mismo tipo de servicio?

```
select a.id_lab, a.nombre_servicio, c.fecha_alta as "nuevo laboratorio", b.fecha_inicio as "inicio servicio", b.fecha_fin as "fin servicio"
from servicios a natural join servicios2 b natural join laboratorios c
group by a.nombre_servicio, a.id_lab, b.fecha_inicio, c.fecha_alta
having a.nombre_servicio = 'GENOTIPADO MUESTRAS'
```

	id_lab numeric (2)	nombre_servicio character varying (28)	nuevo laboratorio date	inicio servicio date	fin servicio date
1	1	GENOTIPADO MUESTRAS	2019-07-17	2019-07-17	2021-10-20
2	5	GENOTIPADO MUESTRAS	2021-10-28	2021-10-28	[null]

6. Muestra el DNI y nombre de los empleados/as que más horas semanales hayan trabajado en cada proyecto.

```
select p.DNI, nombre
from personal p join inv_pro inv on p.DNI=inv.DNI
where horas in (select max(horas)
                from inv_pro
                group by id_pro)
```

	dni [PK] numeric (6)	nombre character varying (24)
1	111111	ALBERTO
2	222222	CARMEN
3	333333	PEDRO
4	444444	ALTEA
5	555555	ALICIA
6	666666	AGUSTÍN
7	777777	ROBERTO
8	999999	LUCÍA
9	111222	NEREA
10	222333	MARÍA
11	444555	ANA
12	555666	NURIA
13	777888	INÉS
14	999000	MÓNICA
15	567890	EMILIO
16	321654	NOELIA
17	7	PAULA
18	111000	JAVIER
19	333000	ALEJANDRO

7. Muestra todos los datos de aquellos empleados/as que más ganan para cada rango en la empresa.

```
select *
from personal p
where salario = (select max(salario)
                from personal
                where rango = p.rango)
order by salario, rango
```

	dni [PK] numeric (6)	nombre character varying (24)	rango character varying (20)	salario numeric (7,2)	dirección character varying (34)	provincia character varying (24)	cp numeric (5)
1	444555	ANA	TÉCNICA LAB	1800.00	Avda. Juan Flórez, 08	CORUÑA	15009
2	777888	INÉS	TÉCNICA LAB	1800.00	Rda. Nelle 15	CORUÑA	15009
3	333000	ALEJANDRO	TÉCNICO LAB	1800.00	Avda. Méndez Núñez, 8	CORUÑA	15009
4	555555	ALICIA	TÉCNICO LAB	1800.00	Avda. de Arteixo, 50	CORUÑA	15009
5	86420	VICENTE	TÉCNICO LAB	1800.00	Calle Posse, 19	CORUÑA	15009
6	567890	EMILIO	TÉCNICO LAB	1800.00	Rda. de Outeiro, 22	CORUÑA	15009
7	987654	ANTONIO	TÉCNICO LAB	1800.00	Calle Matadero, 10	CORUÑA	15009
8	999000	MÓNICA	BIÓLOGA	2200.00	Calle Posse, 25	CORUÑA	15009
9	333333	PEDRO	BIÓLOGO	2200.00	Avda. Méndez Núñez, 7	CORUÑA	15009
10	130031	NIEVES	GENETISTA	2300.00	Praza Orense, 5	CORUÑA	15009
11	999999	LUCÍA	GENETISTA	2300.00	Rúa Nogueira, 3	SANTA CRUZ	15179
12	555666	NURIA	GENETISTA	2300.00	Rda. de Outeiro, 09	CORUÑA	15009
13	222222	CARMEN	GENETISTA	2300.00	Calle Posse, 29	CORUÑA	15009
14	111000	JAVIER	GENETISTA	2300.00	Rda. Nelle, 44	CORUÑA	15009
15	888888	ANDRÉS	GENETISTA	2300.00	Avda. de Arteixo, 37	CORUÑA	15009
16	222333	MARÍA	BIOQUÍMICA	2500.00	Rda. Nelle, 12	CORUÑA	15009
17	7	PAULA	BIOQUÍMICA	2500.00	Avda. Juan Flórez, 1	CORUÑA	15009
18	9009	SANDRA	BIOQUÍMICA	2500.00	Calle Matadero, 2	CORUÑA	15009
19	123456	MARTA	BIOQUÍMICA	2500.00	Calle Orzán, 29	CORUÑA	15009

8. Muestra la marca del equipamiento que ha implicado la mayor inversión realizada por la empresa.

```
select marca
from equipamiento2
group by marca
having avg(precio) >= all (
    select avg(precio)
    from equipamiento2
    group by marca
)
```

marca character varying (24)
1 ILLUMINA

9. Muestra, para cada servicio ofrecido por la empresa, su nombre y cuantas horas semanales dedican en total aquellos trabajadores/as cuyo salario es superior al salario medio de la empresa.

```
select s.nombre_servicio, sum(horas)
from servicios s join trabajar_en t on s.id_servicio = t.id_servicio join personal p on
t.DNI=p.DNI
where salario > (select avg(salario) from personal)
group by s.nombre_servicio
```

nombre_servicio character varying (28)	sum numeric
1 GENOTIPADO MUESTRAS	82
2 REACTIVOS PCR	40
3 ANÁLISIS DATOS	100
4 PREPARACIÓN OLIGOS PCR	76
5 SECUENCIAR	194

10. Muestra los proyectos de la empresa donde hay por lo menos dos trabajadores/as implicados en ellos

```
select id_pro
from inv_pro
group by id_pro
having count(*)>=2
```

	id_pro	
	numeric (2)	
1	22	
2	48	
3	9	
4	61	
5	10	
6	55	
7	39	
8	90	

## CONCLUSIONES

Las principales conclusiones del presente trabajo son:

- I. Se ha llevado a cabo un diseño de base de datos que se ajustaría a las demandas del cliente.
- II. El empleo de la bibliografía recomendada en la asignatura ha permitido tomar las decisiones adecuadas en el transcurso del diseño y justificarlas en base a los posibles problemas que podrían aparecer al realizar operaciones y trabajar en la base de datos.
- III. Se han planteado una serie de consultas de interés en la base de datos generada con lenguaje SQL mediante PostgreSQL aplicando los conocimientos adquiridos en las prácticas de la asignatura.

## BIBLIOGRAFÍA

Elmasri, R. & Navathe, S.B. (2008). *Fundamentos de sistemas de bases de datos*. 5ª Ed. Pearson Education

Stephens, R. (2009). *Diseño de bases de datos*. Anaya Multimedia

## ANEXO I

A continuación, se muestra el script (SQL) correspondiente a la base de datos realizada para el presente trabajo:

```
drop table if exists sedes1 cascade;
drop table if exists sedes2 cascade;
drop table if exists sedes3 cascade;
drop table if exists laboratorios cascade;
drop table if exists servicios cascade;
drop table if exists servicios2 cascade;
drop table if exists equipamiento1 cascade;
drop table if exists equipamiento2 cascade;
drop table if exists eq_lab cascade;
drop table if exists proyectos cascade;
drop table if exists personal cascade;
drop table if exists inv_pro cascade;
drop table if exists trabajar_en cascade;
```

```
CREATE TABLE sedes1 (
  id_sede numeric(2,0),
  polígono character varying(14) UNIQUE,
  área numeric(3,0) NOT NULL,
  n_parcela numeric(2,0),
  CONSTRAINT sedes1_pkey PRIMARY KEY (id_sede),
  CONSTRAINT sedes1_polígono_fkey FOREIGN KEY (polígono) REFERENCES sedes2(polígono),
  CONSTRAINT sedes1_área_fkey FOREIGN KEY (área) REFERENCES sedes3(área)
);
```

```
INSERT INTO sedes1 (id_sede, polígono, área, n_parcela) VALUES (10, 'PLATEA', 111, 99);
INSERT INTO sedes1 (id_sede, polígono, área, n_parcela) VALUES (20, 'CELLA', 222, 98);
INSERT INTO sedes1 (id_sede, polígono, área, n_parcela) VALUES (30, 'VILLARQUEMADO', 333, 97);
INSERT INTO sedes1 (id_sede, polígono, área, n_parcela) VALUES (40, 'ALBARRACÍN', 444, 96);
INSERT INTO sedes1 (id_sede, polígono, área, n_parcela) VALUES (50, 'PERALES', 555, 95);
INSERT INTO sedes1 (id_sede, polígono, área, n_parcela) VALUES (60, 'ALFAMBRA', 666, 94);
INSERT INTO sedes1 (id_sede, polígono, área, n_parcela) VALUES (70, 'VILLAFELISA', 777, 93);
INSERT INTO sedes1 (id_sede, polígono, área, n_parcela) VALUES (80, 'CULLERA', 888, 92);
```

```
CREATE TABLE sedes2 (
  polígono character varying(14) UNIQUE,
  impuestos numeric(4,2) NOT NULL,
  CONSTRAINT sedes2_pkey PRIMARY KEY (polígono)
);
```

```
INSERT INTO sedes2 (polígono, impuestos) VALUES ('PLATEA', 0.20);
INSERT INTO sedes2 (polígono, impuestos) VALUES ('CELLA', 0.40);
INSERT INTO sedes2 (polígono, impuestos) VALUES ('VILLARQUEMADO', 0.21);
INSERT INTO sedes2 (polígono, impuestos) VALUES ('ALBARRACÍN', 0.25);
INSERT INTO sedes2 (polígono, impuestos) VALUES ('PERALES', 0.30);
INSERT INTO sedes2 (polígono, impuestos) VALUES ('ALFAMBRA', 0.50);
INSERT INTO sedes2 (polígono, impuestos) VALUES ('VILLAFELISA', 0.08);
INSERT INTO sedes2 (polígono, impuestos) VALUES ('CULLERA', 0.12);
```

```
CREATE TABLE sedes3 (
    área numeric(3,0) NOT NULL,
    precio numeric(4,0) NOT NULL,
    CONSTRAINT sedes3_pkey PRIMARY KEY (área)
);
```

```
INSERT INTO sedes3 (área, precio) VALUES (111, 200);
INSERT INTO sedes3 (área, precio) VALUES (222, 1000);
INSERT INTO sedes3 (área, precio) VALUES (333, 20);
INSERT INTO sedes3 (área, precio) VALUES (444, 2200);
INSERT INTO sedes3 (área, precio) VALUES (555, 190);
INSERT INTO sedes3 (área, precio) VALUES (666, 500);
INSERT INTO sedes3 (área, precio) VALUES (777, 800);
INSERT INTO sedes3 (área, precio) VALUES (888, 350);
```

```
CREATE TABLE laboratorios (
    id_lab numeric(2,0) NOT NULL,
    nombre character varying(14) UNIQUE,
    fecha_alta date,
    id_sede numeric(2,0),
    CONSTRAINT laboratorios_pkey PRIMARY KEY (id_lab),
    CONSTRAINT laboratorios_id_sede_fkey FOREIGN KEY(id_sede) REFERENCES sedes1(id_sede)
);
```

```
INSERT INTO laboratorios (id_lab, nombre, fecha_alta, id_sede) VALUES (01, 'GENOTIPADO1',
    to_date('17/07/19','dd/mm/yy'), 10);
INSERT INTO laboratorios (id_lab, nombre, fecha_alta, id_sede) VALUES (02, 'SECUENCIACIÓN1',
    to_date('01/08/19','dd/mm/yy'), 20);
INSERT INTO laboratorios (id_lab, nombre, fecha_alta, id_sede) VALUES (03, 'HIBRIDACIÓN',
    to_date('12/01/20','dd/mm/yy'), 30);
INSERT INTO laboratorios (id_lab, nombre, fecha_alta, id_sede) VALUES (04, 'SONDAS_PCR',
    to_date('01/07/19','dd/mm/yy'), 40);
INSERT INTO laboratorios (id_lab, nombre, fecha_alta, id_sede) VALUES (05, 'GENOTIPADO2',
    to_date('28/10/21','dd/mm/yy'), 50);
INSERT INTO laboratorios (id_lab, nombre, fecha_alta, id_sede) VALUES (06, 'MUESTREO',
    to_date('09/06/19','dd/mm/yy'), 60);
INSERT INTO laboratorios (id_lab, nombre, fecha_alta, id_sede) VALUES (07, 'SECUENCIACIÓN2',
    to_date('01/12/21','dd/mm/yy'), 70);
INSERT INTO laboratorios (id_lab, nombre, fecha_alta, id_sede) VALUES (08, 'BIOINFORMÁTICA',
    to_date('01/01/20','dd/mm/yy'), 80);
```

```
CREATE TABLE servicios (
    id_servicio numeric(2,0) NOT NULL,
    nombre_servicio character varying(28),
    tipo_muestras character varying(28),
    fecha_inicio date,
    id_lab numeric(2,0) NOT NULL,
    CONSTRAINT servicios_pkey PRIMARY KEY (id_servicio),
    CONSTRAINT servicios_fecha_inicio_fkey FOREIGN KEY (fecha_inicio) REFERENCES
    servicios2(fecha_inicio),
    CONSTRAINT servicios_id_lab_fkey FOREIGN KEY (id_lab) REFERENCES laboratorios(id_lab)
);
```

```

INSERT INTO servicios (id_servicio, nombre_servicio, tipo_muestras, fecha_inicio, id_lab) VALUES (40,
'GENOTIPADO MUESTRAS', 'HERBÁCEOS', to_date('17/07/19','dd/mm/yy'), 01);
INSERT INTO servicios (id_servicio, nombre_servicio, tipo_muestras, fecha_inicio, id_lab) VALUES (42,
'SECUENCIAR', 'DNA', to_date('01/08/19','dd/mm/yy'), 02);
INSERT INTO servicios (id_servicio, nombre_servicio, tipo_muestras, fecha_inicio, id_lab) VALUES (44,
'PREPARACIÓN OLIGOS PCR', 'DNA', to_date('12/01/20','dd/mm/yy'), 03);
INSERT INTO servicios (id_servicio, nombre_servicio, tipo_muestras, fecha_inicio, id_lab) VALUES (46,
'REACTIVOS PCR', 'REACTIVOS QUÍMICOS', to_date('01/07/19','dd/mm/yy'), 04);
INSERT INTO servicios (id_servicio, nombre_servicio, tipo_muestras, fecha_inicio, id_lab) VALUES (48,
'GENOTIPADO MUESTRAS', 'HERBÁCEOS', to_date('28/10/21','dd/mm/yy'), 05);
INSERT INTO servicios (id_servicio, nombre_servicio, tipo_muestras, fecha_inicio, id_lab) VALUES (50,
'RECOGIDA DE MUESTRAS', 'HERBÁCEOS', to_date('09/06/19','dd/mm/yy'), 06);
INSERT INTO servicios (id_servicio, nombre_servicio, tipo_muestras, fecha_inicio, id_lab) VALUES (52,
'SECUENCIAR', 'DNA', to_date('01/12/21','dd/mm/yy'), 07);
INSERT INTO servicios (id_servicio, nombre_servicio, tipo_muestras, fecha_inicio, id_lab) VALUES (54,
'ANÁLISIS DATOS', 'DATOS INFORMÁTICOS', to_date('01/01/20','dd/mm/yy'), 08);
#INSERT INTO servicios (id_servicio, nombre_servicio, tipo_muestras, fecha_inicio, id_lab) VALUES
(56, 'JEFE LABORATORIO', NULL, to_date('01/07/19','dd/mm/yy'), );
#INSERT INTO servicios (id_servicio, nombre_servicio, tipo_muestras, fecha_inicio, id_lab) VALUES
(58, 'JEFE EMPRESA', NULL, to_date('01/07/19','dd/mm/yy'), );

```

```

CREATE TABLE servicios2 (
    fecha_inicio date,
    fecha_fin date,
    gastos numeric(7,2),
    id_lab numeric(2,0) NOT NULL,
    CONSTRAINT servicios2_pkey PRIMARY KEY (fecha_inicio),
    CONSTRAINT servicios2_id_lab_fkey FOREIGN KEY (id_lab) REFERENCES laboratorios(id_lab)
);

```

```

INSERT INTO servicios2 (fecha_inicio, fecha_fin, gastos, id_lab) VALUES
(to_date('17/07/19','dd/mm/yy'), to_date('20/10/21','dd/mm/yy'), 2300.00, 01);
INSERT INTO servicios2 (fecha_inicio, fecha_fin, gastos, id_lab) VALUES
(to_date('01/08/19','dd/mm/yy'), NULL, 3000.80, 02);
INSERT INTO servicios2 (fecha_inicio, fecha_fin, gastos, id_lab) VALUES
(to_date('12/01/20','dd/mm/yy'), NULL, 5000.20, 03);
INSERT INTO servicios2 (fecha_inicio, fecha_fin, gastos, id_lab) VALUES
(to_date('01/07/19','dd/mm/yy'), NULL, 9000.00, 04);
INSERT INTO servicios2 (fecha_inicio, fecha_fin, gastos, id_lab) VALUES
(to_date('28/10/21','dd/mm/yy'), NULL, 300.00, 05);
INSERT INTO servicios2 (fecha_inicio, fecha_fin, gastos, id_lab) VALUES
(to_date('09/06/19','dd/mm/yy'), NULL, 499.99, 06);
INSERT INTO servicios2 (fecha_inicio, fecha_fin, gastos, id_lab) VALUES
(to_date('01/12/21','dd/mm/yy'), to_date('20/12/21','dd/mm/yy'), 230.50, 07);
INSERT INTO servicios2 (fecha_inicio, fecha_fin, gastos, id_lab) VALUES
(to_date('01/01/20','dd/mm/yy'), NULL, 600.00, 08);

```

```

CREATE TABLE equipamiento1 (
    id_equipo numeric(3,0) NOT NULL,
    modelo character varying(14) UNIQUE,
    tipo character varying(30),
    CONSTRAINT equipamiento1_pkey PRIMARY KEY (id_equipo),
    CONSTRAINT equipamiento1_modelo_fkey FOREIGN KEY (modelo) REFERENCES
equipamiento2(modelo)

```

);

```
INSERT INTO equipamiento1 (id_equipo, modelo, tipo) VALUES (101, 'AX47', 'SECUENCIADOR');
INSERT INTO equipamiento1 (id_equipo, modelo, tipo) VALUES (202, 'MM55', 'SECUENCIADOR');
INSERT INTO equipamiento1 (id_equipo, modelo, tipo) VALUES (303, 'AP99', 'TERMOCICLADOR');
INSERT INTO equipamiento1 (id_equipo, modelo, tipo) VALUES (404, 'PIP44', 'MICROPIPETA');
INSERT INTO equipamiento1 (id_equipo, modelo, tipo) VALUES (505, 'PIP56', 'MICROPIPETA');
INSERT INTO equipamiento1 (id_equipo, modelo, tipo) VALUES (606, 'AG100', 'KIT EXTRACCIÓN
DNA');
INSERT INTO equipamiento1 (id_equipo, modelo, tipo) VALUES (707, 'PCR007', 'REACTIVOS PARA
PCR');
INSERT INTO equipamiento1 (id_equipo, modelo, tipo) VALUES (808, 'TT88', 'GRADILLAS');
INSERT INTO equipamiento1 (id_equipo, modelo, tipo) VALUES (909, 'C09', 'PACK TUBOS DE
ENSAYO');
INSERT INTO equipamiento1 (id_equipo, modelo, tipo) VALUES (007, 'INF2', 'ORDENADOR
ÚLTIMA GENERACIÓN');
```

```
CREATE TABLE equipamiento2 (
    modelo character varying(14),
    marca character varying(24),
    precio numeric(10,2),
    CONSTRAINT equipamiento2_pkey PRIMARY KEY (modelo)
);
```

```
INSERT INTO equipamiento2 (modelo, marca, precio) VALUES ('AX47', 'THERMO FISHER',
30000.00);
INSERT INTO equipamiento2 (modelo, marca, precio) VALUES ('MM55', 'ILLUMINA', 55000.00);
INSERT INTO equipamiento2 (modelo, marca, precio) VALUES ('AP99', 'THERMO FISHER',
80000.00);
INSERT INTO equipamiento2 (modelo, marca, precio) VALUES ('PIP44', 'FISHER SCIENTIFIC',
56.00);
INSERT INTO equipamiento2 (modelo, marca, precio) VALUES ('PIP56', 'TECNYLAB', 69.90);
INSERT INTO equipamiento2 (modelo, marca, precio) VALUES ('AG100', 'INVITROGEN', 135.00);
INSERT INTO equipamiento2 (modelo, marca, precio) VALUES ('PCR007', 'QUIAGEN', 320.00);
INSERT INTO equipamiento2 (modelo, marca, precio) VALUES ('TT88', 'THERMO FISHER', 32.00);
INSERT INTO equipamiento2 (modelo, marca, precio) VALUES ('C09', 'INVITROGEN', 45.00);
INSERT INTO equipamiento2 (modelo, marca, precio) VALUES ('INF2', 'ASUS', 2100.00);
```

```
CREATE TABLE eq_lab (
    id_lab numeric(2,0) NOT NULL REFERENCES laboratorios(id_lab),
    id_equipo numeric(2,0) NOT NULL REFERENCES equipamiento1(id_equipo),
    horas numeric(2,0) ,
    CONSTRAINT eq_lab_pkey PRIMARY KEY (id_lab, id_equipo)
);
```

```
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (01, 101, 46);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (01, 404, 46);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (01, 606, 46);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (01, 707, 46);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (01, 808, 46);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (01, 909, 46);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (02, 202, 45);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (02, 404, 45);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (02, 606, 45);
```



```

INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (02, 707, 45);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (02, 808, 45);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (02, 909, 45);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (02, 007, 45);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (03, 404, 30);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (03, 505, 30);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (03, 707, 30);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (03, 303, 30);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (03, 808, 30);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (03, 909, 30);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (04, 303, 20);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (04, 505, 20);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (04, 707, 20);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (04, 808, 20);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (04, 909, 20);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (04, 007, 20);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (05, 202, 50);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (05, 404, 50);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (05, 606, 50);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (05, 707, 50);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (05, 808, 50);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (05, 909, 50);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (06, 808, 30);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (06, 007, 30);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (07, 202, 47);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (07, 404, 47);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (07, 606, 47);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (07, 707, 47);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (07, 808, 47);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (07, 909, 47);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (07, 007, 47);
INSERT INTO eq_lab (id_lab, id_equipo, horas) VALUES (08, 202, 40);

```

```

CREATE TABLE proyectos (
  id_pro numeric(2,0) NOT NULL,
  nombre character varying(60),
  presupuesto numeric(7,2),
  id_lab numeric(2,0) NOT NULL,
  CONSTRAINT proyectos_pkey PRIMARY KEY (id_pro),
  CONSTRAINT proyectos_id_lab_fkey FOREIGN KEY (id_lab) REFERENCES laboratorios(id_lab)
);

```

```

INSERT INTO proyectos (id_pro, nombre, presupuesto, id_lab) VALUES (61, 'GENOTIPACIÓN
  ESPECIES DE INTERÉS', 100000.00, 01);
INSERT INTO proyectos (id_pro, nombre, presupuesto, id_lab) VALUES (90, 'SECUENCIAR TODAS
  ESPECIES', 800000, 02);
INSERT INTO proyectos (id_pro, nombre, presupuesto, id_lab) VALUES (22, 'FORMAR SONDAS PCR
  1', 40000, 03);
INSERT INTO proyectos (id_pro, nombre, presupuesto, id_lab) VALUES (10, 'FORMAR SONDAS PCR
  2', 70000, 04);
INSERT INTO proyectos (id_pro, nombre, presupuesto, id_lab) VALUES (39, 'GENOTIPARCIÓN
  NUEVAS ESPECIES', 100000, 05);
INSERT INTO proyectos (id_pro, nombre, presupuesto, id_lab) VALUES (48, 'RECOGIDA DIVERSAS
  Y POTENCIALES ESPECIES', 15000, 06);

```



```
INSERT INTO proyectos (id_pro, nombre, presupuesto, id_lab) VALUES (09, 'SECUENCIAR
VARIANTES DE ESPECIES DE INTERÉS', 75000, 07);
INSERT INTO proyectos (id_pro, nombre, presupuesto, id_lab) VALUES (55, 'DISEÑO DE SONDAS
PCR Y ANÁLISIS DE DATOS', 10000, 08);
```

```
CREATE TABLE personal (
  DNI numeric(6,0) NOT NULL,
  nombre character varying(24) NOT NULL,
  rango character varying(20) NOT NULL,
  salario numeric(7,2),
  dirección character varying(34),
  provincia character varying(24),
  CP numeric (5,0),
  CONSTRAINT personal_pkey PRIMARY KEY (DNI)
);
```

```
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (111111,
'ALBERTO', 'BIOINFORMÁTICO', 3000.00, 'Avda. Juan Flórez, 47', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (222222,
'CARMEN', 'GENETISTA', 2300.00, 'Calle Posse, 29', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (333333,
'PEDRO', 'BIÓLOGO', 2200.00, 'Avda. Méndez Núñez, 7', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (444444,
'ALTEA', 'BIOQUÍMICA', 2500.00, 'Avda. Juan Flórez, 27', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (555555,
'ALICIA', 'TÉCNICO LAB', 1800.00, 'Avda. de Arteixo, 50', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (666666,
'AGUSTÍN', 'BIOINFORMÁTICO', 3000.00, 'Rda. de Outeiro, 20', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (777777,
'ROBERTO', 'BIOINFORMÁTICO', 3000.00, 'Calle Orzán, 27', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (888888,
'ANDRÉS', 'GENETISTA', 2300.00, 'Avda. de Arteixo, 37', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (999999,
'LUCÍA', 'GENETISTA', 2300.00, 'Rúa Nogueira, 3', 'SANTA CRUZ', 15179);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (111222,
'NEREA', 'BIOQUÍMICA', 2500.00, 'Praza Orense, 5', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (222333,
'MARÍA', 'BIOQUÍMICA', 2500.00, 'Rda. Nelle, 12', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (444555,
'ANA', 'TÉCNICA LAB', 1800.00, 'Avda. Juan Flórez, 08', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (555666,
'NURIA', 'GENETISTA', 2300.00, 'Rda. de Outeiro, 09', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (777888,
'INÉS', 'TÉCNICA LAB', 1800.00, 'Rda. Nelle 15', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (999000,
'MÓNICA', 'BIÓLOGA', 2200.00, 'Calle Posse, 25', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (123456,
'MARTA', 'BIOQUÍMICA', 2500.00, 'Calle Orzán, 29', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (567890,
'EMILIO', 'TÉCNICO LAB', 1800.00, 'Rda. de Outeiro, 22', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (987654,
'ANTONIO', 'TÉCNICO LAB', 1800.00, 'Calle Matadero, 10', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (321654,
'NOELIA', 'BIOINFORMÁTIC', 3000.00, 'Calle Matadero, 10', 'CORUÑA', 15009);
```

```

INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (000007,
'PAULA', 'BIOQUÍMICA', 2500.00, 'Avda. Juan Flórez, 1', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (111000,
'JAVIER', 'GENETISTA', 2300.00, 'Rda. Nelle, 44', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (333000,
'ALEJANDRO', 'TÉCNICO LAB', 1800.00, 'Avda. Méndez Núñez, 8', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (009009,
'SANDRA', 'BIOQUÍMICA', 2500.00, 'Calle Matadero, 2', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (130031,
'NIEVES', 'GENETISTA', 2300.00, 'Praza Orense, 5', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (086420,
'VICENTE', 'TÉCNICO LAB', 1800.00, 'Calle Posse, 19', 'CORUÑA', 15009);
INSERT INTO personal (DNI, nombre, rango, salario, dirección, provincia, CP) VALUES (889900,
'MIRANDA', 'BIOINFORMÁTICA', 3000.00, 'Avda. Arteixo, 78', 'CORUÑA', 15009);

```

```

CREATE TABLE inv_pro (
    DNI numeric(6,0) NOT NULL REFERENCES personal(DNI),
    id_pro numeric(2,0) NOT NULL REFERENCES proyectos(id_pro),
    horas numeric(2,0),
    CONSTRAINT inv_pro_pkey PRIMARY KEY (DNI, id_pro)
);

```

```

INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (111111, 55, 30);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (222222, 61, 40);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (333333, 48, 40);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (444444, 10, 40);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (555555, 48, 50);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (666666, 22, 46);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (777777, 55, 30);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (888888, 90, 42);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (999999, 55, 30);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (111222, 61, 40);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (222333, 22, 46);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (444555, 10, 40);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (555666, 10, 40);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (777888, 61, 40);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (999000, 48, 40);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (123456, 90, 42);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (567890, 22, 46);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (987654, 90, 42);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (321654, 90, 45);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (000007, 39, 48);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (111000, 39, 48);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (333000, 39, 48);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (009009, 09, 52);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (130031, 09, 52);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (086420, 09, 52);
INSERT INTO inv_pro (DNI, id_pro, horas) VALUES (889900, 09, 52);

```

```

CREATE TABLE trabajar_en (
    DNI numeric(6,0) NOT NULL REFERENCES personal(DNI),
    id_servicio numeric(2,0) NOT NULL REFERENCES servicios(id_servicio),
    horas numeric(2,0),

```

```

CONSTRAINT trabajar_en_pkey PRIMARY KEY (DNI, id_servicio)
);

INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (111111, 54, 50);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (222222, 40, 40);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (333333, 50, 40);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (444444, 46, 40);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (555555, 50, 40);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (666666, 44, 38);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (777777, 54, 50);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (888888, 42, 45);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (999999, 54, 50);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (111222, 40, 40);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (222333, 44, 38);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (444555, 46, 40);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (555666, 46, 40);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (777888, 40, 40);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (999000, 50, 40);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (123456, 42, 45);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (567890, 44, 38);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (987654, 42, 45);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (321654, 42, 45);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (000007, 48, 42);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (111000, 48, 42);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (333000, 48, 42);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (009009, 52, 52);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (130031, 52, 52);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (086420, 52, 52);
INSERT INTO trabajar_en (DNI, id_servicio, horas) VALUES (889900, 52, 52);

```