
OBLIGATORIO FINAL ADMINISTRACIÓN BASE DE DATOS

MODALIDAD ONLINE

La sociedad médica "SALUD_URUGUAY" se dedica a la atención de pacientes en un centro hospitalario.

Dicha sociedad cuenta con numerosos doctores, enfermeros, servicio de limpieza, socios, etc.

Cada socio concurre a la clínica para pedir un diagnóstico, repetir medicamentos y/o recibir un tratamiento. Esta clínica cuenta con un servicio de camas para pacientes que han sido operados.

Dado que no cuentan con una base de datos para organizar su empresa, lo contratan a usted para ayudarlos con este trabajo.

Para ello, tenemos una primera cita donde nos dan la siguiente información:

1. La empresa cuenta con 20 doctores, 10 enfermeros, 20 personas en el servicio de limpieza, 50 socios (este número no es fijo, dado que varía cada día).
2. La empresa ofrece los siguientes servicios:
 - a. agenda primera consulta
 - b. agenda para recibir medicamentos
 - c. agenda consulta tratamiento
 - d. agenda para operación
 - e. agenda de cama
3. La empresa cuenta con 10 camas, dos por sala. Es decir, hay cinco salas.
4. Cada paciente tiene un Dr. cabecera, el cual si está de licencia tiene un Dr. suplente que atiende a dicho paciente.
5. Cada paciente internado es atendido por un Dr. (Dr. asociado o suplente) y por muchos enfermeros
6. La empresa cuenta con 50 medicamentos diferentes para las diferentes enfermedades que tienen los pacientes.
7. La empresa cuenta además con varios proveedores que son quienes suministran los diferentes medicamentos.

1. **Se deberá confeccionar el MER que usted crea necesario para reflejar la realidad de la empresa. ¿Qué preguntas se le ocurren para poder realizarlo? Además de dicho diagrama se deberán identificar las restricciones o supuestos que crea necesario en el que baso su MER.**

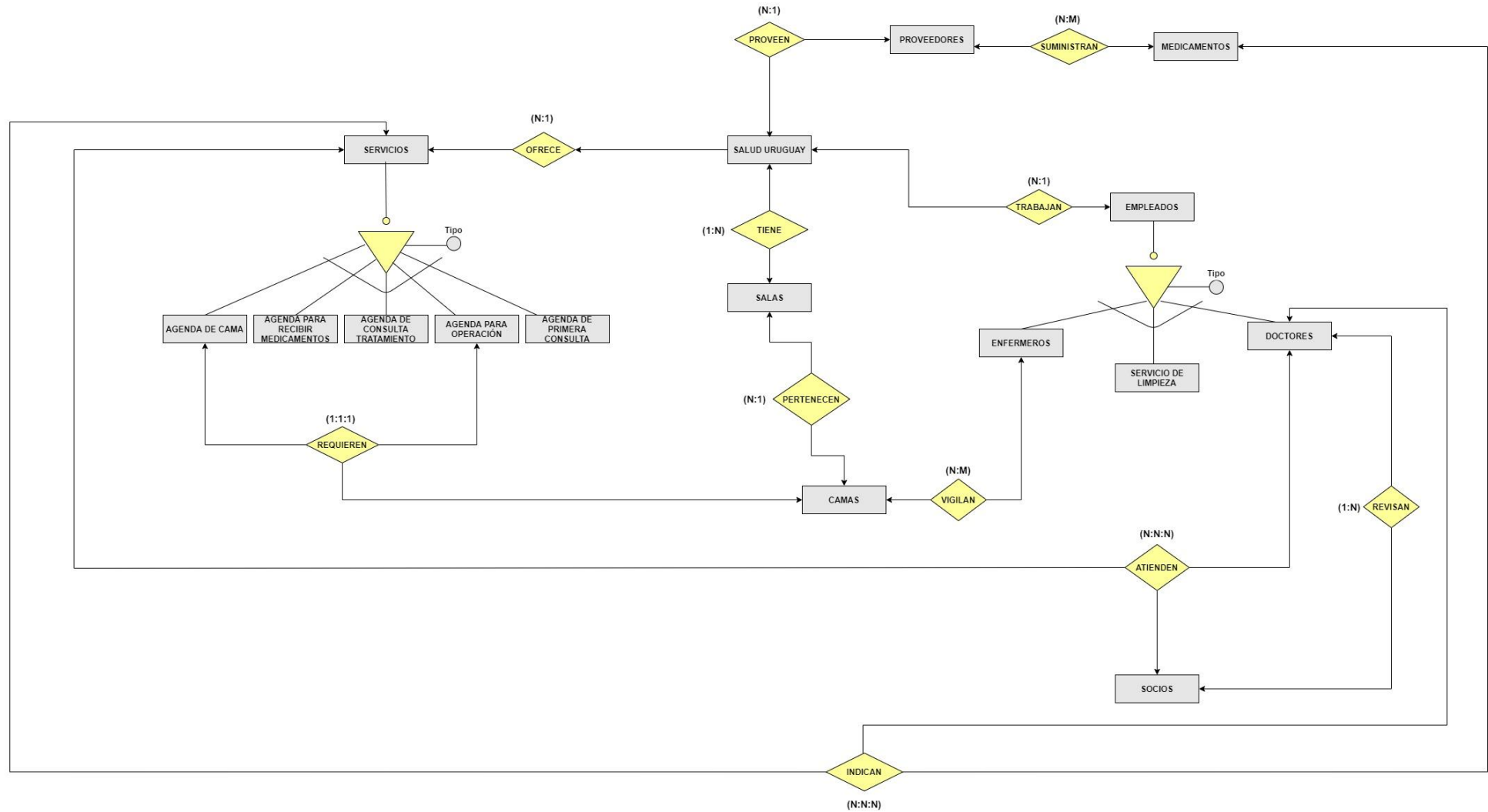
Preguntas:

- ¿Cuáles especialidades tienen los doctores?
- ¿Qué tipo de doctor atiende al paciente?
- ¿Qué salas tiene el hospital?
- ¿Cuáles son los tipos de asistencia (servicios) que recibe el socio?
- ¿Qué enfermero vigila cada habitación?
- ¿Cuántos pacientes son atendidos?

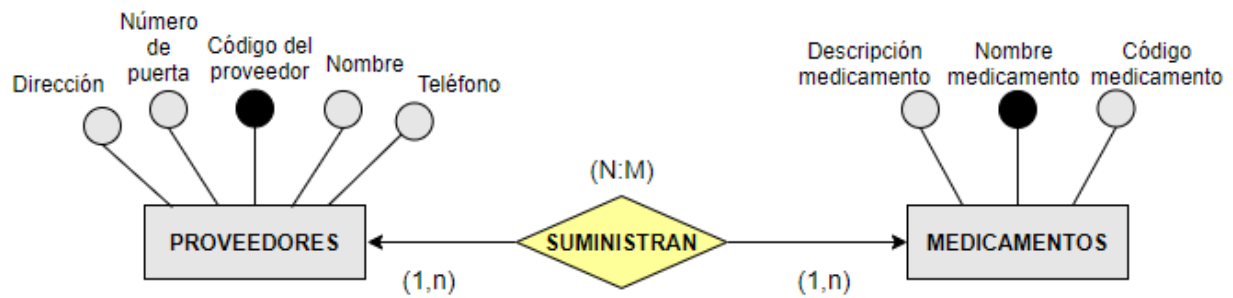
Supuestos:

- Relación de jerarquía exclusiva para empleados.
- Relación de jerarquía exclusiva para servicios.
- Relación de jerarquía exclusiva para doctores.
- Relación de jerarquía exclusiva para socios.
- El centro médico puede aumentar su cantidad de socios en el futuro, entonces va a necesitar más de 2 camas por sala.
- La empresa cuenta con 20 doctores, 10 enfermeros y 20 personas en el servicio de limpieza, pero suponemos que, al aumentar la cantidad de socios en el futuro, también aumentará la cantidad de empleados.
- Clínica = Sociedad médica = Empresa.
- Paciente = Socio.
- Médico = Doctor.

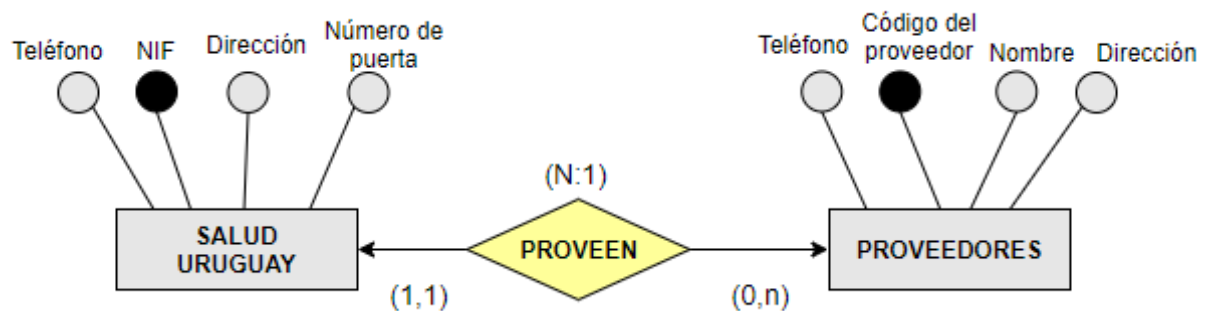
Modelo Entidad-Relación



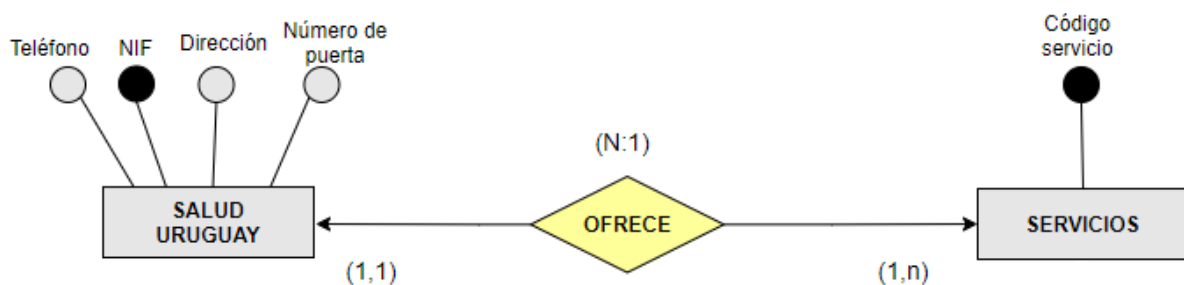
Los atributos de cada entidad son los siguientes:



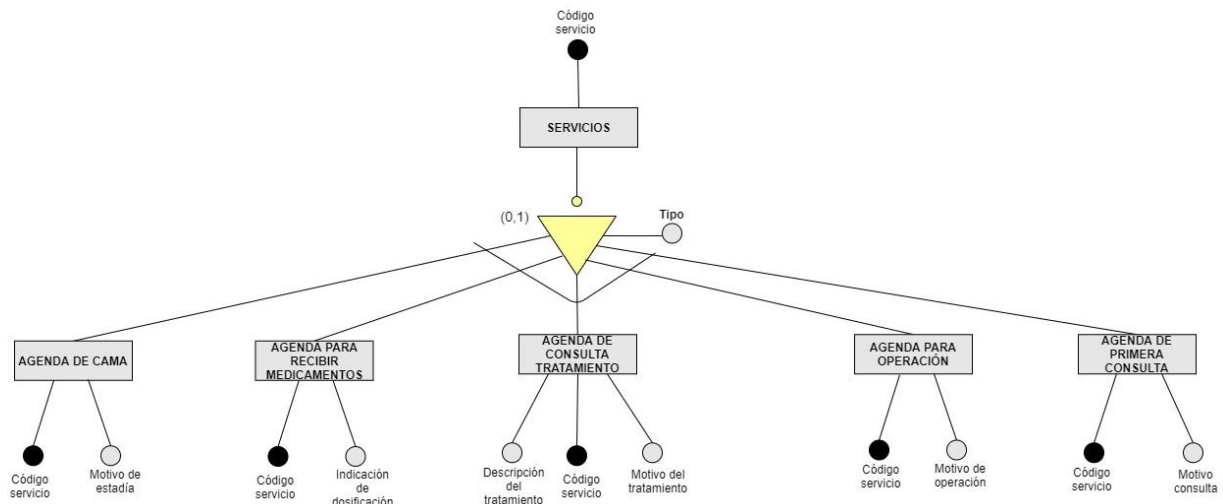
Uno o varios medicamentos son suministrados por al menos uno o varios proveedores y uno o varios proveedores suministran al menos uno o varios medicamentos.



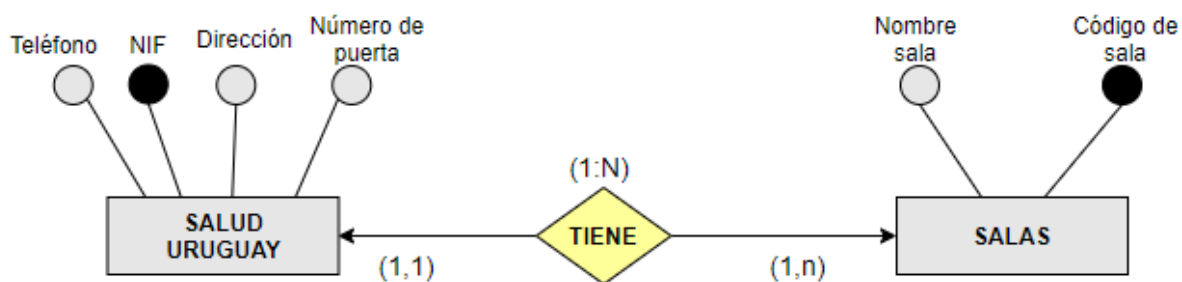
Un proveedor provee al centro de salud y Salud Uruguay puede ser provisto por ningún proveedor o varios proveedores.



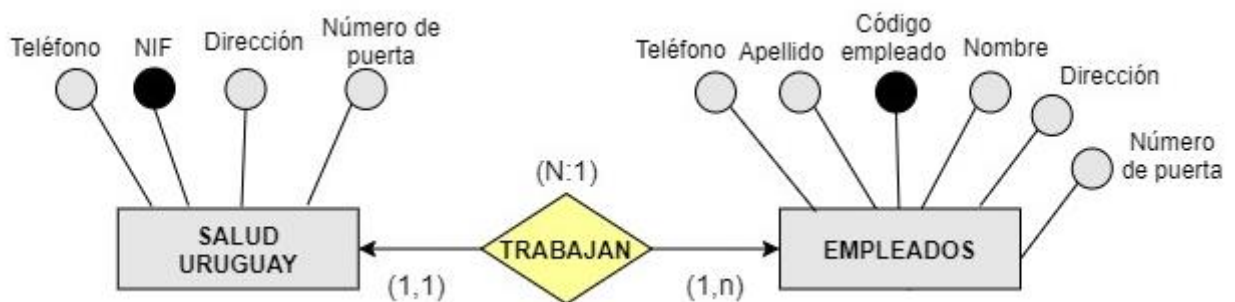
Salud Uruguay ofrece uno o varios servicios y los servicios son ofrecidos a Salud Uruguay, siendo él un único centro de salud.



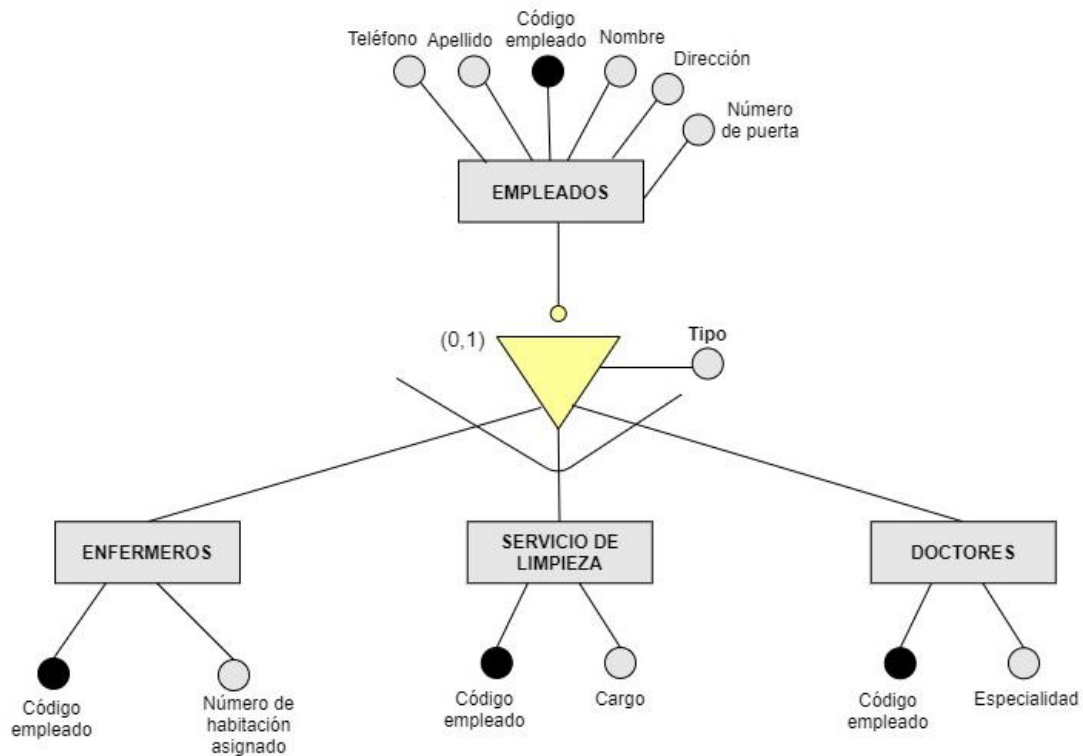
Como la jerarquía es exclusiva, un servicio solo puede ser agenda de cama, recibir medicamentos, consulta tratamiento, operación y primera consulta, por lo tanto la cardinalidad para cada uno de los tipos de servicio sería (0,1) dado que una instancia de servicio puede ser solo de un tipo.



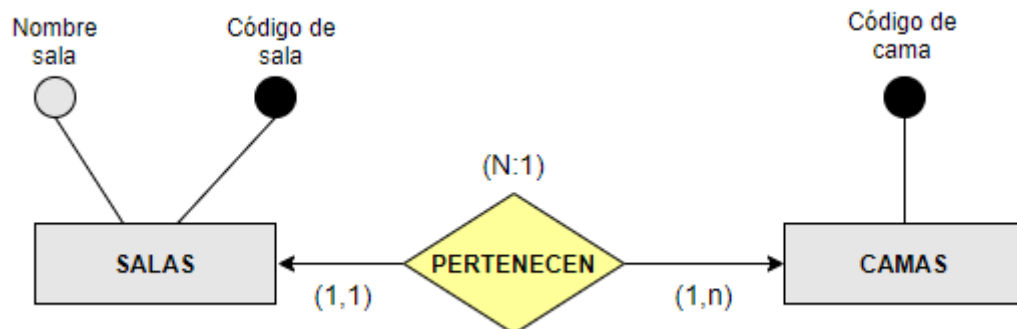
Salud Uruguay tiene una o varias salas y las salas pertenecen solamente a un centro de salud.



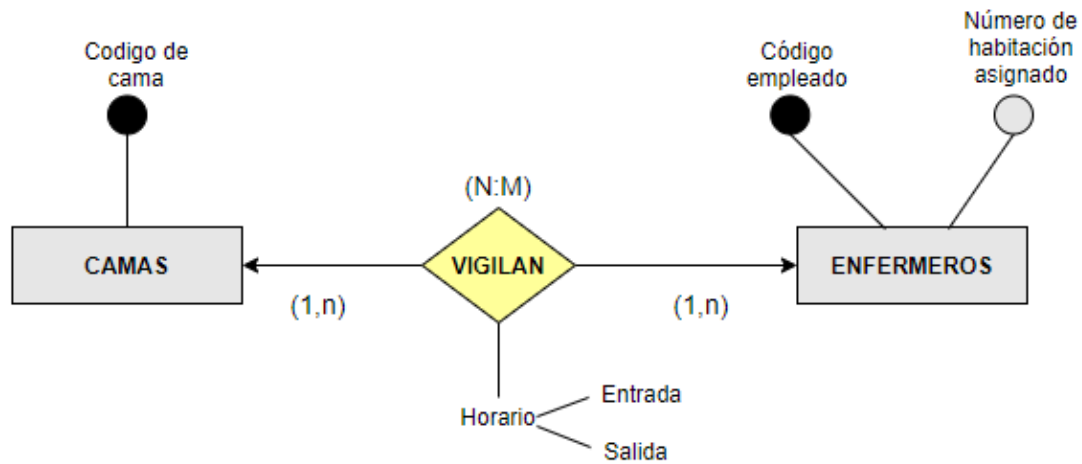
Los empleados trabajan solo en Salud Uruguay y en Salud Uruguay trabajan uno o varios empleados. Esto se contempla bajo el supuesto que la empresa puede aumentar la cantidad de empleados en el futuro.



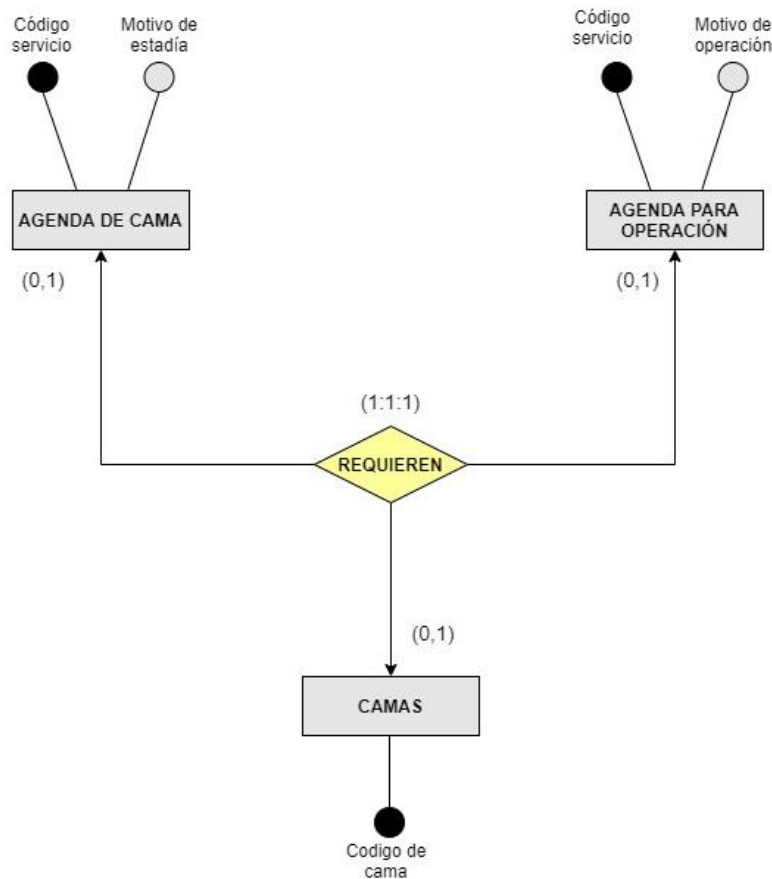
Como la jerarquía es exclusiva, un empleado solo puede ser enfermero, doctor o servicio de limpieza, por lo tanto la cardinalidad para cada uno de los tipos de empleados sería (0,1) dado que una instancia de empleado puede ser solo de un tipo.



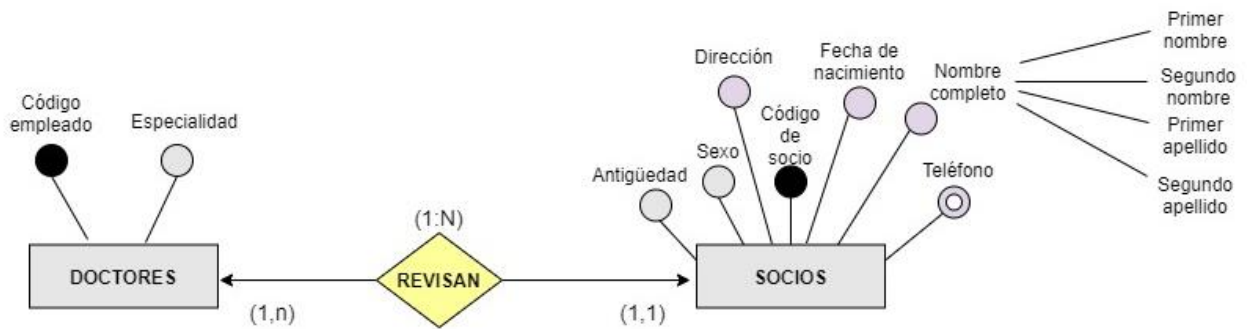
Una cama pertenece a una única sala y una sala puede tener una o varias camas. Esto se contempla bajo el supuesto que Salud Uruguay puede aumentar su cantidad de socios, entonces va a requerir más de 2 camas por sala.



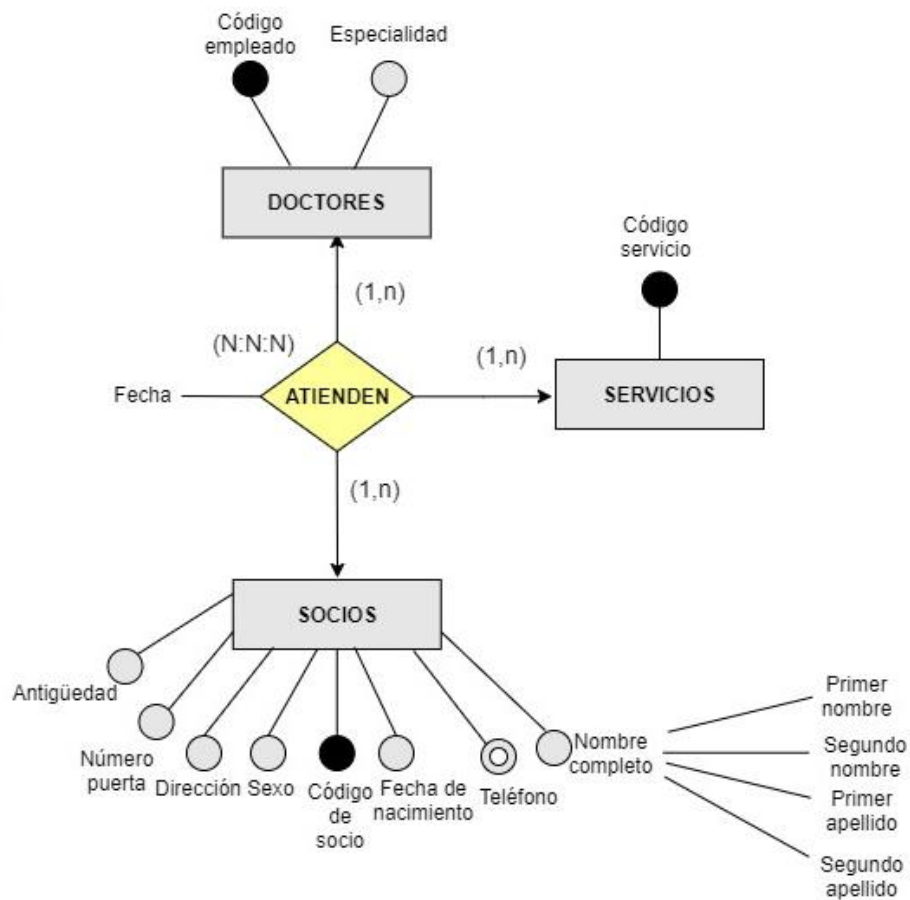
Uno o varios enfermeros vigilan las camas y las camas son vigiladas por uno o varios enfermeros.



Una o ninguna cama puede ser requerida para una operación y para un servicio de agenda de cama, una o ninguna operación requiere una cama y un servicio de agenda de cama requiere utilizar una o ninguna cama.



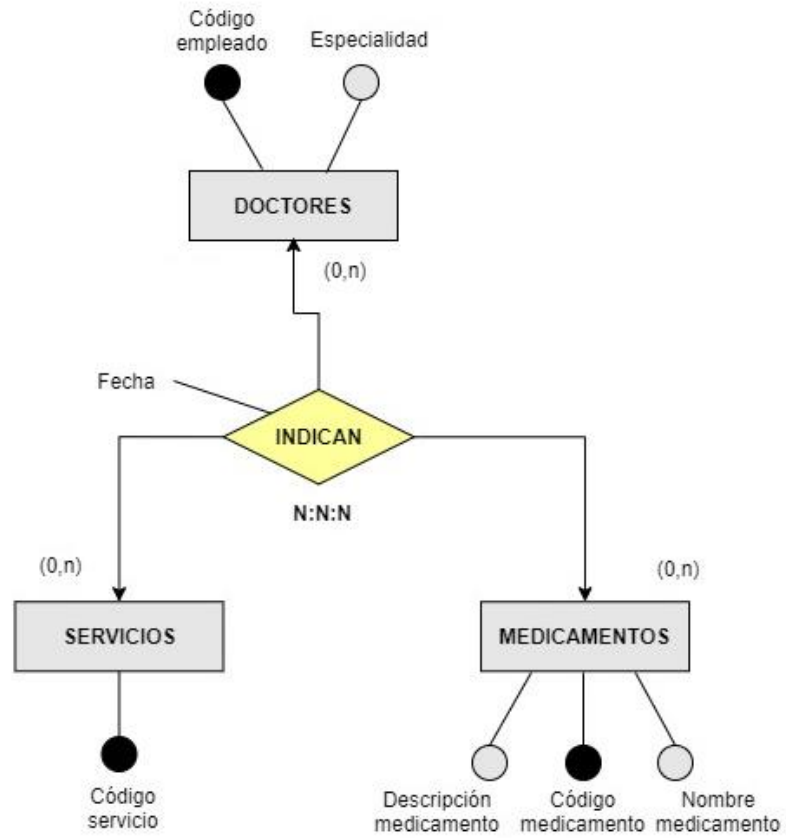
Un doctor revisa a varios socios y uno o varios socios son revisados por un doctor.



Un socio puede ser atendido por muchos doctores y un médico puede atender a muchos pacientes.

Un doctor puede atender muchos servicios y un servicio es atendido por varios doctores.

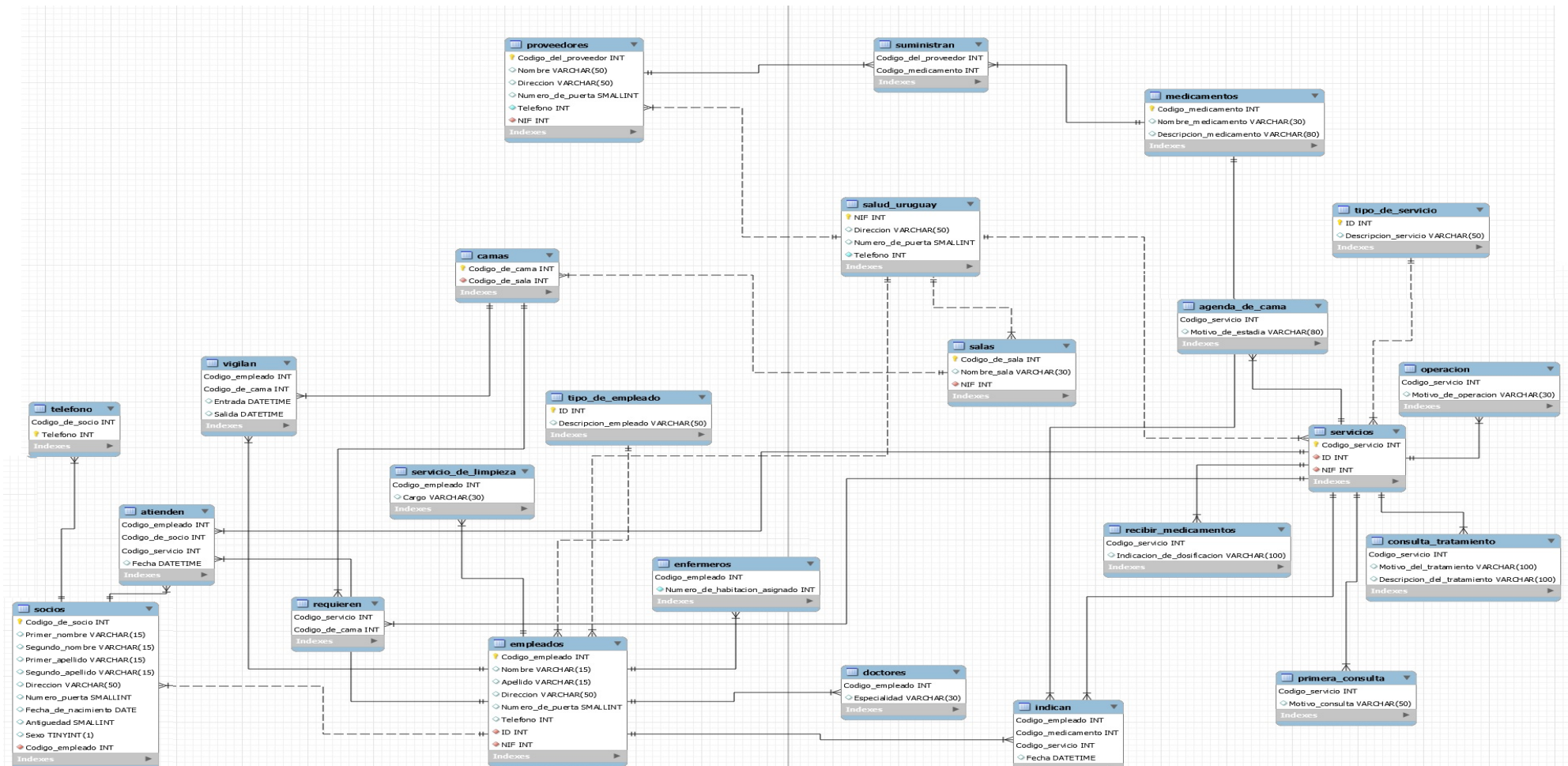
Un socio puede ser atendido por muchos servicios y un servicio es atendido por varios socios.



Ninguno o varios doctores indican medicamentos y servicios, ninguno o varios medicamentos son indicados por doctores y ninguno o varios servicios son indicados por doctores.

2. Se deberá confeccionar el Modelo Relacional basándonos en el diagrama del se pide uno.

Modelo Relacional



3. Se debe indicar las consultas de creación de objetos de base de datos (tablas, índices, fk, etc). Además de las consultas de inserción de datos en cada una de ellas (al menos dos registros por tabla).

El schema se llama “proyecto_final”.

En el anexo se encuentran las sentencias para ser ejecutadas.

- **Tabla Salud_Uruguay**

```
CREATE TABLE proyecto_final.Salud_Uruguay(
  NIF int not null,
  Direccion varchar(50),
  Numero_de_puerta smallint,
  Telefono int not null,
  PRIMARY KEY (NIF)
);
```

```
insert into proyecto_final.salud_uruguay(NIF, Direccion, Numero_de_puerta, Telefono)
values (1001, 'Av.Dr.Luis Alberto de Herrera', 1390, 26221005);
```

	NIF	Direccion	Numero_de_puerta	Telefono
▶	1001	Av.Dr.Luis Alberto de Herrera	1390	26221005
*	NULL	NULL	NULL	NULL

- **Tabla Proveedores**

```
CREATE TABLE proyecto_final.proveedores(
  Codigo_del_proveedor int not null auto_increment,
  Nombre varchar(50),
  Direccion varchar(50),
  Numero_de_puerta smallint,
  Telefono int not null,
  NIF int not null,
  PRIMARY KEY (Codigo_del_proveedor),
  INDEX (Nombre),
  CONSTRAINT NIF_fk FOREIGN KEY (NIF) REFERENCES proyecto_final.salud_uruguay(NIF)
);
```

```
insert into proyecto_final.proveedores(Codigo_del_proveedor, Nombre, Direccion, Numero_de_puerta, Telefono,NIF)
values (1, 'Scienza', 'Av.Dr.Luis Alberto de Herrera', 1248, 219889000,1001), (2, 'Roemmers', 'Zelmar Micheline', 1230, 29029626,1001),
(3, 'Laboratorios Gautier', 'Democracia', 2132, 24010594,1001);
```

	Codigo_del_proveedor	Nombre	Direccion	Numero_de_puerta	Telefono	NIF
▶	1	Scienza	Av.Dr.Luis Alberto de Herrera	1248	219889000	1001
	2	Roemmers	Zelmar Micheline	1230	29029626	1001
	3	Laboratorios Gautier	Democracia	2132	24010594	1001
*	NULL	NULL	NULL	NULL	NULL	NULL

- Tabla Medicamentos

```
CREATE TABLE proyecto_final.medicamentos(
Codigo_medicamento int not null auto_increment,
Nombre_medicamento varchar(30),
Descripcion_medicamento varchar(80),
PRIMARY KEY (Codigo_medicamento)
);
```

```
insert into proyecto_final.medicamentos(Codigo_medicamento, Nombre_medicamento, Descripcion_medicamento)
values (1, 'Amoxidal 500', 'Antibiótico bactericida de amplio espectro'), (2, 'Dolex', 'Comprimidos que contienen paracetamol'),
(3, 'Dolex Flex', 'Contiene Orfenadrina que es un relajante muscular del musculo esquelético'),
(4, 'Priorin', 'Capsulas que contribuye al mantenimiento del cabello en condiciones normales'),
(5, 'Colagenol C', 'Suplemento nutricional que ayuda al desarrollo muscular'), (6, 'Plagrel', 'Prevenir en adultos la formación de coagulos sanguíneos');
```

	Codigo_medicamento	Nombre_medicamento	Descripcion_medicamento
▶	1	Amoxidal 500	Antibiótico bactericida de amplio espectro
	2	Dolex	Comprimidos que contienen paracetamol
	3	Dolex Flex	Contiene Orfenadrina que es un relajante musc...
	4	Priorin	Capsulas que contribuye al mantenimiento del c...
	5	Colagenol C	Suplemento nutricional que ayuda al desarrollo ...
	6	Plagrel	Prevenir en adultos la formación de coagulos sa...
*	NULL	NULL	NULL

- Tabla Suministran

```
CREATE TABLE proyecto_final.suministran(
Codigo_del_proveedor int not null,
Codigo_medicamento int not null,
PRIMARY KEY (Codigo_del_proveedor, Codigo_medicamento),
CONSTRAINT Codigo_del_proveedor_fork FOREIGN KEY (Codigo_del_proveedor) REFERENCES proyecto_final.proveedores(Codigo_del_proveedor),
CONSTRAINT Codigo_medicamento_fk FOREIGN KEY (Codigo_medicamento) REFERENCES proyecto_final.medicamentos(Codigo_medicamento)
);
```

```
insert into proyecto_final.suministran(Codigo_del_proveedor, Codigo_medicamento)
values (1, 1),(2, 2),(2, 3),(3, 4),(1, 5), (2, 6);
```

	Codigo_del_proveedor	Codigo_medicamento
▶	1	1
	2	2
	2	3
	3	4
	1	5
	2	6
*	NULL	NULL

- Tabla Salas

```
CREATE TABLE proyecto_final.salas(
Codigo_de_sala int not null auto_increment,
Nombre_sala varchar(30),
NIF int not null,
PRIMARY KEY (Codigo_de_sala),
CONSTRAINT NIF_foreignk FOREIGN KEY (NIF) REFERENCES proyecto_final.salud_uruguay(NIF)
);
```

```
insert into proyecto_final.salas(Codigo_de_sala, Nombre_sala, NIF)
values (1, 'Maternidad', 1001),(2, 'Emergencias', 1001),(3, 'Operaciones', 1001), (4, 'General', 1001);
```

	Codigo_de_sala	Nombre_sala	NIF
▶	1	Maternidad	1001
	2	Emergencias	1001
	3	Operaciones	1001
	4	General	1001
*	NULL	NULL	NULL

- Tabla Tipo_de_servicio

```
CREATE TABLE proyecto_final.tipo_de_servicio(
ID int not null auto_increment,
Descripcion_servicio varchar(50),
PRIMARY KEY (ID)
);
```

```
insert into proyecto_final.tipo_de_servicio(ID, Descripcion_servicio)
values (1, 'Agenda de cama'),(2,'Recibir medicamentos'),(3,'Consulta tratamiento'),(4,'Operacion'),(5,'Primera consulta');
```

	ID	Descripcion_servicio
▶	1	Agenda de cama
	2	Recibir medicamentos
	3	Consulta tratamiento
	4	Operacion
	5	Primera consulta
*	NULL	NULL

- **Tabla Servicios**

```
CREATE TABLE proyecto_final.servicios(
Codigo_servicio int not null auto_increment,
ID int not null,
NIF int not null,
PRIMARY KEY (Codigo_servicio),
CONSTRAINT ID_fk FOREIGN KEY (ID) REFERENCES proyecto_final.tipo_de_servicio(ID),
CONSTRAINT NIF_foreignkey FOREIGN KEY (NIF) REFERENCES proyecto_final.salud_uruguay(NIF)
);
```

```
insert into proyecto_final.servicios(Codigo_servicio, ID, NIF)
values (1, 1, 1001),(2, 2, 1001),(3, 3, 1001),(4, 4, 1001),(5, 5, 1001),(6, 1, 1001),(7, 2, 1001),(8, 3, 1001),(9, 4, 1001),(10, 5, 1001);
```

	Codigo_servicio	ID	NIF
▶	1	1	1001
	2	2	1001
	3	3	1001
	4	4	1001
	5	5	1001
	6	1	1001
	7	2	1001
	8	3	1001
	9	4	1001
	10	5	1001
*	NULL	NULL	NULL

- **Tabla Consulta_tratamiento (Agenda de consulta tratamiento)**

```
CREATE TABLE proyecto_final.consulta_tratamiento(
Codigo_servicio int not null,
Motivo_del_tratamiento varchar(100),
Descripcion_del_tratamiento varchar(100),
PRIMARY KEY (Codigo_servicio),
CONSTRAINT Codigo_servicio_fk FOREIGN KEY (Codigo_servicio) REFERENCES proyecto_final.servicios(Codigo_servicio)
);
```

```
insert into proyecto_final.consulta_tratamiento(Codigo_servicio, Motivo_del_tratamiento, Descripcion_del_tratamiento)
values (3, 'Otitis media aguda', 'Se le indica al paciente la ingesta de antibiotico del via oral'),
(8, 'Dislocacion de hombro', 'Se le ajusta el hombro al paciente y se le indica ingerir paracetamol');
```

	Codigo_servicio	Motivo_del_tratamiento	Descripcion_del_tratamiento
▶	3	Otitis media aguda	Se le indica al paciente la ingesta de antibiotico ...
	8	Dislocacion de hombro	Se le ajusta el hombro al paciente y se le indica i...
*	NULL	NULL	NULL

- **Tabla Recibir_medicamentos (Agenda para recibir medicamentos)**

```
CREATE TABLE proyecto_final.recibir_medicamentos(
Codigo_servicio int not null,
Indicacion_de_dosificacion varchar(100),
PRIMARY KEY (Codigo_servicio),
CONSTRAINT Codigo_servicio_foreignk FOREIGN KEY (Codigo_servicio) REFERENCES proyecto_final.servicios(Codigo_servicio)
);
```

```
insert into proyecto_final.recibir_medicamentos(Codigo_servicio, Indicacion_de_dosificacion)
values (2, 'Amoxidal 500mg durante 2 semanas, 2 comprimidos por dia'),(7,'Dolex durante 1 semana cada 8 horas');
```

	Codigo_servicio	Indicacion_de_dosificacion
▶	2	Amoxidal 500mg durante 2 semanas, 2 comprimi...
	7	Dolex durante 1 semana cada 8 horas
*	NULL	NULL

- **Tabla Primera_consulta (Agenda de primera consulta)**

```
CREATE TABLE proyecto_final.primera_consulta(
Codigo_servicio int not null,
Motivo_consulta varchar(50),
PRIMARY KEY (Codigo_servicio),
CONSTRAINT Codigo_servicio_foreignkey FOREIGN KEY (Codigo_servicio) REFERENCES proyecto_final.servicios(Codigo_servicio)
);
```

```
insert into proyecto_final.primera_consulta(Codigo_servicio, Motivo_consulta)
values (5, 'Dolor intenso en oido derecho'),(10,'Dolor lumbar');
```

	Codigo_servicio	Motivo_consulta
▶	5	Dolor intenso en oido derecho
	10	Dolor lumbar
*	NULL	NULL

- **Tabla Operacion (Agenda para operación)**

```
CREATE TABLE proyecto_final.operacion(
Codigo_servicio int not null,
Motivo_de_operacion varchar(30),
PRIMARY KEY (Codigo_servicio),
CONSTRAINT Codigo_servicio_forekey FOREIGN KEY (Codigo_servicio) REFERENCES proyecto_final.servicios(Codigo_servicio)
);
```

```
insert into proyecto_final.operacion(Codigo_servicio, Motivo_de_operacion)
values (4, 'Frecuencia cardiaca anormal'),(9, 'Apendicitis');
```

	Codigo_servicio	Motivo_de_operacion
▶	4	Frecuencia cardiaca anormal
	9	Apendicitis
*	NULL	NULL

- **Tabla Agenda_de_cama**

```
CREATE TABLE proyecto_final.agenda_de_cama(
Codigo_servicio int not null,
Motivo_de_estadia varchar(80),
PRIMARY KEY (Codigo_servicio),
CONSTRAINT Codigo_servicio_fork FOREIGN KEY (Codigo_servicio) REFERENCES proyecto_final.servicios(Codigo_servicio)
);
```

```
insert into proyecto_final.agenda_de_cama(Codigo_servicio, Motivo_de_estadia)
values (1, 'Parto prematuro'),(6,'Rotura de apendice');
```

	Codigo_servicio	Motivo_de_estadia
▶	1	Parto prematuro
	6	Rotura de apendice
✱	NULL	NULL

- **Tabla Tipo_de_empleado**

```
CREATE TABLE proyecto_final.tipo_de_empleado(
ID int not null auto_increment,
Descripcion_empleado varchar(50),
PRIMARY KEY (ID)
);
```

```
insert into proyecto_final.tipo_de_empleado(ID, Descripcion_empleado)
values (1, 'Enfermero'),(2,'Doctor'),(3,'Servicio limpieza');
```

	ID	Descripcion_empleado
▶	1	Enfermero
	2	Doctor
	3	Servicio limpieza
✱	NULL	NULL

- **Tabla Empleados**

```
CREATE TABLE proyecto_final.empleados(
Codigo_empleado int not null auto_increment,
Nombre varchar(15),
Apellido varchar(15),
Direccion varchar(50),
Numero_de_puerta smallint,
Telefono int,
ID int not null,
NIF int not null,
PRIMARY KEY (Codigo_empleado),
CONSTRAINT ID_fkey FOREIGN KEY (ID) REFERENCES proyecto_final.tipo_de_empleado(ID),
CONSTRAINT NIF_fkey FOREIGN KEY (NIF) REFERENCES proyecto_final.salud_uruguay(NIF)
);
```

```
insert into proyecto_final.empleados(Codigo_empleado, Nombre, Apellido, Direccion, Numero_de_puerta, Telefono, ID, NIF)
values (1, 'Juana', 'Rodriguez', '21 de Setiembre', 2179, 27117186, 1, 1001),
(2, 'Federico', 'Diaz', 'Av. Julio Herrera y Reissig', 565, 27142714, 1, 1001), (3, 'Maria', 'Perez', 'Av. Italia', 5775, 26017733, 1, 1001),
(4, 'Paula', 'Gomez', 'Orinoco', 4951, 26196857, 2, 1001), (5, 'Jorge', 'Perez', 'Orinoco', 4951, 26196857, 2, 1001),
(6, 'Lara', 'Gonzalez', 'Av. 8 de Octubre', 2709, 24873375, 2, 1001), (7, 'Maria', 'Gonzalez', 'Jose Enrique Rodo', 1671, 24011010, 3, 1001),
(8, 'Mario', 'Sosa', '26 de Marzo', 3572, 26225077, 3, 1001), (9, 'Leonel', 'Lopez', 'Magallanes', 1434, 24002859, 3, 1001),
(10, 'Juan', 'Ponce', 'Av. Italia', 5775, 26017733, 2, 1001), (11, 'Jorge', 'Facal', 'Av. Italia', 5800, 26017834, 2, 1001),
(12, 'Alan', 'Cortez', 'Sarandi', 429, 29165016, 2, 1001);
```

	Codigo_empleado	Nombre	Apellido	Direccion	Numero_de_puerta	Telefono	ID	NIF
▶	1	Juana	Rodriguez	21 de Setiembre	2179	27117186	1	1001
	2	Federico	Diaz	Av. Julio Herrera y Reissig	565	27142714	1	1001
	3	Maria	Perez	Av. Italia	5775	26017733	1	1001
	4	Paula	Gomez	Orinoco	4951	26196857	2	1001
	5	Jorge	Perez	Orinoco	4951	26196857	2	1001
	6	Lara	Gonzalez	Av. 8 de Octubre	2709	24873375	2	1001
	7	Maria	Gonzalez	Jose Enrique Rodo	1671	24011010	3	1001
	8	Mario	Sosa	26 de Marzo	3572	26225077	3	1001
	9	Leonel	Lopez	Magallanes	1434	24002859	3	1001
	10	Juan	Ponce	Av. Italia	5775	26017733	2	1001
	11	Jorge	Facal	Av. Italia	5800	26017834	2	1001
	12	Alan	Cortez	Sarandi	429	29165016	2	1001
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- **Tabla Servicio_de_limpieza**

```
CREATE TABLE proyecto_final.servicio_de_limpieza(
Codigo_empleado int not null,
Cargo varchar(30),
PRIMARY KEY (Codigo_empleado),
CONSTRAINT Codigo_empleado_fkey FOREIGN KEY (Codigo_empleado) REFERENCES proyecto_final.empleados(Codigo_empleado)
);
```

```
insert into proyecto_final.servicio_de_limpieza(Codigo_empleado, Cargo)
values (1, 'Limpiador areas criticas'), (2, 'Encargada sector'), (3, 'Limpiador areas generales');
```

	Codigo_empleado	Cargo
▶	1	Limpiador areas criticas
	2	Encargada sector
	3	Limpiador areas generales
*	NULL	NULL

- **Tabla Enfermeros**

```
CREATE TABLE proyecto_final.enfermeros(
Codigo_empleado int not null,
Numero_de_habitacion_asignado int not null,
PRIMARY KEY (Codigo_empleado),
CONSTRAINT Codigo_empleado_fk FOREIGN KEY (Codigo_empleado) REFERENCES proyecto_final.empleados(Codigo_empleado)
);
```

```
insert into proyecto_final.enfermeros(Codigo_empleado, Numero_de_habitacion_asignado)
values (1, 204),(2,301),(3,102);
```

	Codigo_empleado	Numero_de_habitacion_asignado
▶	1	204
	2	301
	3	102
*	NULL	NULL

- **Tabla Doctores**

```
CREATE TABLE proyecto_final.doctores(
Codigo_empleado int not null,
Especialidad varchar(30),
PRIMARY KEY (Codigo_empleado),
CONSTRAINT Codigo_empleado_fork FOREIGN KEY (Codigo_empleado) REFERENCES proyecto_final.empleados(Codigo_empleado)
);
```

```
insert into proyecto_final.doctores(Codigo_empleado, Especialidad)
values (4, 'internista'), (5, 'internista'), (6, 'cirujano'), (10, 'medicina general'), (11, 'cirujano'), (12, 'internista');
```

	Codigo_empleado	Especialidad
▶	4	internista
	5	internista
	6	cirujano
	10	medicina general
	11	cirujano
	12	internista
*	NULL	NULL

- **Tabla Camas**

```
CREATE TABLE proyecto_final.camas(
Codigo_de_cama int not null auto_increment,
Codigo_de_sala int not null,
PRIMARY KEY (Codigo_de_cama),
CONSTRAINT Codigo_de_sala_fk FOREIGN KEY (Codigo_de_sala) REFERENCES proyecto_final.salas(Codigo_de_sala)
);
```

```
insert into proyecto_final.camas(Codigo_de_cama, Codigo_de_sala)
values (1, 1),(2, 2),(3, 3);
```

	Codigo_de_cama	Codigo_de_sala
▶	1	1
	2	2
	3	3
*	NULL	NULL

- Tabla Vigilan

```
CREATE TABLE proyecto_final.vigilan(
Codigo_empleado int not null,
Codigo_de_cama int not null,
Entrada datetime,
Salida datetime,
PRIMARY KEY (Codigo_empleado, Codigo_de_cama),
CONSTRAINT Codigo_empleado_forkey FOREIGN KEY (Codigo_empleado) REFERENCES proyecto_final.empleados(Codigo_empleado),
CONSTRAINT Codigo_de_cama_fk FOREIGN KEY (Codigo_de_cama) REFERENCES proyecto_final.camas(Codigo_de_cama)
);

insert into proyecto_final.vigilan(Codigo_de_cama, Codigo_empleado, Entrada, Salida)
values (1, 1, '2020-08-11 17:37:00', '2020-04-22 17:57:00'),(2,1, '2020-04-22 11:07:00', '2020-04-22 11:10:00'),
(3,2, '2020-02-15 08:41:00', '2020-02-15 09:06:00');
```

	Codigo_empleado	Codigo_de_cama	Entrada	Salida
▶	1	1	2020-08-11 17:37:00	2020-04-22 17:57:00
	1	2	2020-04-22 11:07:00	2020-04-22 11:10:00
	2	3	2020-02-15 08:41:00	2020-02-15 09:06:00
✱	NULL	NULL	NULL	NULL

- Tabla Socios

```
CREATE TABLE proyecto_final.socios(
Codigo_de_socio int not null auto_increment,
Primer_nombre varchar(15),
Segundo_nombre varchar(15),
Primer_apellido varchar(15),
Segundo_apellido varchar(15),
Direccion varchar(50),
Numero_puerta smallint,
Fecha_de_nacimiento date,
Antiguedad smallint,
Sexo bool,
Codigo_empleado int not null,
PRIMARY KEY (Codigo_de_socio),
CONSTRAINT Codigo_empleado_foreignkey FOREIGN KEY (Codigo_empleado) REFERENCES proyecto_final.empleados(Codigo_empleado)
);

insert into proyecto_final.socios(Codigo_de_socio, Primer_nombre, Segundo_nombre, Primer_apellido, Segundo_apellido, Direccion, Numero_puerta, Fecha_de_nacimiento, Antiguedad, Sexo, Codigo_empleado)
values (1, 'Olivia', 'Maria', 'Fernandez', 'Perez', 'Av. Joaquín Suarez', 2727, '1981-06-10', 5, 1, 4), (2, 'Julio', 'Jose', 'Fernandez', 'Perez', 'Av. Joaquín Suarez', 2727, '1989-12-11', 5, 0, 6),
(3, 'Natalia', 'Mercedes', 'Martinez', 'Sanchez', 'Av. 18 de Julio', 1078, '1991-11-11', 15, 1, 4), (4, 'Chiara', 'Cara', 'Ferragni', 'Serragni', 'Placido Ellauri', 3494, '1987-05-07', 30, 1, 11),
(5, 'Federico', 'Leonardo', 'Berrinzaghi', 'Catino', 'Av. 8 de Octubre', 3395, '1989-10-15', 24, 0, 5), (6, 'Daniela', 'Carmen', 'Mosco', 'Sbres', 'Obligado', 955, '1989-12-27', 8, 1, 11),
(7, 'Ana', 'Maria', 'Lopez', 'Gomez', 'Av. Alfredo Arocena', 1584, '1966-06-06', 32, 1, 11), (8, 'Daniel', 'Vicente', 'Rodriguez', 'Perez', 'Av. Alfredo Arocena', 1584, '1977-07-07', 27, 0, 10),
(9, 'Elizabeth', 'Agustina', 'Iglesias', 'Cartier', 'Av. Gral. Jose Maria Paz', 1429, '1981-12-02', 2, 0, 11), (10, 'Sofia', 'Alba', 'Lombardi', 'Murphy', 'Blvr. Gral. Artigas', 3817, '2014-07-31', 3, 1, 4),
(11, 'Isabella', 'Cristina', 'Moretti', 'Walsh', '21 de Setiembre', 2701, '2006-03-13', 12, 1, 12), (12, 'Sara', 'Valeria', 'Santoro', 'Byrne', 'Av. Gral. Rondeau', 1437, '2011-11-05', 9, 1, 4),
(13, 'Alice', 'Paola', 'Leone', 'Ferrari', 'Colonia', 1651, '2017-02-28', 1, 1, 6), (14, 'Alessia', 'Alana', 'Contee', 'Ricci', 'Julio Herrera y Obes', 1466, '2004-04-17', 4, 1, 10),
(15, 'Giorgio', '', 'Armani', 'Raimondi', 'Joaquín Nuñez', 2787, '1934-07-11', 20, 0, 10);
```

	Codigo_de_socio	Primer_nombre	Segundo_nombre	Primer_apellido	Segundo_apellido	Direccion	Numero_puerta	Fecha_de_nacimiento	Antiguedad	Sexo	Codigo_empleado
▶	1	Olivia	Maria	Fernandez	Perez	Av. Joaquín Suarez	2727	1981-06-10	5	1	4
	2	Julio	Jose	Fernandez	Perez	Av. Joaquín Suarez	2727	1989-12-11	5	0	6
	3	Natalia	Mercedes	Martinez	Sanchez	Av. 18 de Julio	1078	1991-11-11	15	1	4
	4	Chiara	Cara	Ferragni	Serragni	Placido Ellauri	3494	1987-05-07	30	1	11
	5	Federico	Leonardo	Berrinzaghi	Catino	Av. 8 de Octubre	3395	1989-10-15	24	0	5
	6	Daniela	Carmen	Mosco	Sbres	Obligado	955	1989-12-27	8	1	11
	7	Ana	Maria	Lopez	Gomez	Av. Alfredo Arocena	1584	1966-06-06	32	1	11
	8	Daniel	Vicente	Rodriguez	Perez	Av. Alfredo Arocena	1584	1977-07-07	27	0	10
	9	Elizabeth	Agustina	Iglesias	Cartier	Av. Gral. Jose Maria Paz	1429	1981-12-02	2	0	11
	10	Sofia	Alba	Lombardi	Murphy	Blvr. Gral. Artigas	3817	2014-07-31	3	1	4
	11	Isabella	Cristina	Moretti	Walsh	21 de Setiembre	2701	2006-03-13	12	1	12
	12	Sara	Valeria	Santoro	Byrne	Av. Gral. Rondeau	1437	2011-11-05	9	1	4
	13	Alice	Paola	Leone	Ferrari	Colonia	1651	2017-02-28	1	1	6
	14	Alessia	Alana	Contee	Ricci	Julio Herrera y Obes	1466	2004-04-17	4	1	10
	15	Giorgio		Armani	Raimondi	Joaquín Nuñez	2787	1934-07-11	20	0	10
✱	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- **Tabla Telefono**

```
CREATE TABLE proyecto_final.telefono(
Codigo_de_socio int not null,
Telefono int,
PRIMARY KEY (Codigo_de_socio,Telefono),
CONSTRAINT Codigo_de_socio_foreignkey FOREIGN KEY (Codigo_de_socio) REFERENCES proyecto_final.socios(Codigo_de_socio)
);
```

```
insert into proyecto_final.telefono(Codigo_de_socio, Telefono)
values (1, 23365451), (1, 099282546), (2, 23365451), (2, 098417666), (3, 093123456), (4, 099214412),
(5, 092555444), (6, 27144545), (7, 24184547), (8, 24184547), (9, 24002233), (10, 22098076), (10, 099222333),
(11, 27122202), (11, 094121212), (12, 29000803), (12, 092414547), (13, 24009071), (13, 095555555), (14, 20308701), (14, 099999999), (15, 092222222);
```

	Codigo_de_socio	Telefono
▶	1	23365451
	1	99282546
	2	23365451
	2	98417666
	3	93123456
	4	99214412
	5	92555444
	6	27144545
	7	24184547
	8	24184547
	9	24002233
	10	22098076
	10	99222333
	11	27122202
	11	94121212
	12	29000803
	12	92414547
	13	24009071
	13	95555555
	14	20308701
	14	99999999
	15	92222222
*	NULL	NULL

- **Tabla Atienden**

```
CREATE TABLE proyecto_final.atienden(
Codigo_empleado int not null,
Codigo_de_socio int not null,
Codigo_servicio int not null,
Fecha datetime,
PRIMARY KEY (Codigo_empleado, Codigo_de_socio, Codigo_servicio),
CONSTRAINT Codigo_empleado_foreignk FOREIGN KEY (Codigo_empleado) REFERENCES proyecto_final.empleados(Codigo_empleado),
CONSTRAINT Codigo_de_socio_fkey FOREIGN KEY (Codigo_de_socio) REFERENCES proyecto_final.socios(Codigo_de_socio),
CONSTRAINT Codigo_servicio_fkey FOREIGN KEY (Codigo_servicio) REFERENCES proyecto_final.servicios(Codigo_servicio)
);

insert into proyecto_final.atienden(Codigo_empleado, Codigo_de_socio, Codigo_servicio, Fecha)
values (5, 3, 3, '2020-04-14 15:50:00'),(6, 4, 10, '2019-12-27 08:50:00'),
(5, 5, 5, '2020-08-27 07:20:00'),(12,5,10,'2020-08-07 15:50:00'),(6,2,9,'2020-08-22 23:50:00'),
(11, 6, 6, '2020-09-15 19:26:00'), (10, 8, 3, '2020-09-29 14:22:00'), (6, 7, 10, '2020-08-09 11:50:00'), (6, 9, 4, '2020-01-14 13:26:00'),
(5, 1, 1, '2020-09-22 05:30:00'), (6, 9, 6, '2020-09-18 03:33:00'), (5, 3, 6, '2020-09-20 06:56:00'), (11, 7, 9, '2020-09-23 04:27:00'),
(12, 5, 9, '2020-09-27 02:22:00'), (12,11,1,'2020-09-26 12:32:00'), (4,12,6,'2020-09-30 22:02:00'),
(6,13,9,'2020-09-30 19:54:00'), (12,11,8,'2020-10-02 09:34:00'), (11,10,9,'2020-10-12 03:27:00'), (6,13,3,'2020-10-18 16:21:00'),
(10,14,2,'2020-05-26 12:44:00'), (4,12,7,'2020-06-06 10:01:00'), (6,14,9,'2020-08-09 07:50:00'),
(10,15,10,'2020-08-21 15:51:00'), (10,15,5,'2020-08-11 09:57:00');
```

	Codigo_empleado	Codigo_de_socio	Codigo_servicio	Fecha
▶	4	12	6	2020-09-30 22:02:00
	4	12	7	2020-06-06 10:01:00
	5	1	1	2020-09-22 05:30:00
	5	3	3	2020-04-14 15:50:00
	5	3	6	2020-09-20 06:56:00
	5	5	5	2020-08-27 07:20:00
	6	2	9	2020-08-22 23:50:00
	6	4	10	2019-12-27 08:50:00
	6	7	10	2020-08-09 11:50:00
	6	9	4	2020-01-14 13:26:00
	6	9	6	2020-09-18 03:33:00
	6	13	3	2020-10-18 16:21:00
	6	13	9	2020-09-30 19:54:00
	6	14	9	2020-08-09 07:50:00
	10	8	3	2020-09-29 14:22:00
	10	14	2	2020-05-26 12:44:00
	10	15	5	2020-08-11 09:57:00
	10	15	10	2020-08-21 15:51:00
	11	6	6	2020-09-15 19:26:00
	11	7	9	2020-09-23 04:27:00
	11	10	9	2020-10-12 03:27:00
	12	5	9	2020-09-27 02:22:00
	12	5	10	2020-08-07 15:50:00
	12	11	1	2020-09-26 12:32:00
	12	11	8	2020-10-02 09:34:00
*	NULL	NULL	NULL	NULL

- **Tabla Requieren**

```
CREATE TABLE proyecto_final.requieren(
  Codigo_servicio int not null,
  Codigo_de_cama int not null,
  PRIMARY KEY (Codigo_servicio, Codigo_de_cama),
  CONSTRAINT Codigo_servicio_foreigkey FOREIGN KEY (Codigo_servicio) REFERENCES proyecto_final.servicios(Codigo_servicio),
  CONSTRAINT Codigo_de_cama_foreigkey FOREIGN KEY (Codigo_de_cama) REFERENCES proyecto_final.camas(Codigo_de_cama)
);
```

```
insert into proyecto_final.requieren(Codigo_servicio, Codigo_de_cama)
values (1, 1),(6,2), (4,3);
```

	Codigo_servicio	Codigo_de_cama
▶	1	1
	6	2
	4	3
*	NULL	NULL

- **Tabla Indican**

```
CREATE TABLE proyecto_final.indican(
  Codigo_empleado int not null,
  Codigo_medimento int not null,
  Codigo_servicio int not null,
  Fecha datetime,
  PRIMARY KEY (Codigo_empleado, Codigo_medimento, Codigo_servicio),
  CONSTRAINT Codigo_empleado_forek FOREIGN KEY (Codigo_empleado) REFERENCES proyecto_final.empleados(Codigo_empleado),
  CONSTRAINT Codigo_medimento_forek FOREIGN KEY (Codigo_medimento) REFERENCES proyecto_final.medicamentos(Codigo_medimento),
  CONSTRAINT Codigo_servicio_forek FOREIGN KEY (Codigo_servicio) REFERENCES proyecto_final.servicios(Codigo_servicio)
);
```

```
insert into proyecto_final.indican(Codigo_empleado, Codigo_medimento, Codigo_servicio, Fecha)
values (5,1,3,'2020-04-14 16:00:00'), (5,2,5,'2020-08-27 07:30:00'), (12,3,10,'2020-08-07 15:58:00'), (10,1,3,'2020-09-29 14:30:00'),
(12,3,8,'2020-10-02 09:47:00'), (10,1,2,'2020-05-26 12:49:00'), (4,2,7,'2020-06-06 10:09:00'), (10,2,2,'2020-08-20 10:46:00');
```

	Codigo_empleado	Codigo_medimento	Codigo_servicio	Fecha
▶	4	2	7	2020-06-06 10:09:00
	5	1	3	2020-04-14 16:00:00
	5	2	5	2020-08-27 07:30:00
	10	1	2	2020-05-26 12:49:00
	10	1	3	2020-09-29 14:30:00
	10	2	2	2020-08-20 10:46:00
	12	3	8	2020-10-02 09:47:00
	12	3	10	2020-08-07 15:58:00
*	NULL	NULL	NULL	NULL

4. Se deben indicar las consultas que brinden la siguiente información.

En el anexo se encuentran las sentencias para ser ejecutadas.

A. Cantidad de socios que tiene "SALUD_URUGUAY".

```
SELECT count(*) as Cantidad_de_socios
FROM proyecto_final.socios;
```

	Cantidad_de_socios
▶	15

B. Cantidad de socios distintos que realizaron una consulta por primera vez el último mes (agenda primer consulta).

Supongo que el último mes es Agosto.

Esta consulta la realicé de tres formas distintas, obteniendo el mismo resultado.

Forma 1:

```
SELECT count(DISTINCT A.Codigo_de_socio) as Cantidad_socios_primera_consulta
FROM proyecto_final.atienden as A INNER JOIN proyecto_final.servicios as SV ON
A.Codigo_servicio = SV.Codigo_servicio INNER JOIN proyecto_final.tipo_de_servicio as TSV ON SV.ID = TSV.ID
WHERE A.Fecha BETWEEN '2020-08-01 00:00:00' AND '2020-08-31 23:59:59' AND Descripcion_servicio = 'Primera consulta';
```

	Cantidad_socios_primera_consulta
▶	3

Si quisiéramos ver en detalle a qué pacientes corresponden esos tres servicios, se debe hacer la siguiente consulta:

```
SELECT S.Primer_nombre, S.Primer_apellido, count(DISTINCT A.Codigo_de_socio) as Cantidad_socios_primera_consulta
FROM proyecto_final.atienden as A INNER JOIN proyecto_final.servicios as SV ON
A.Codigo_servicio = SV.Codigo_servicio INNER JOIN proyecto_final.tipo_de_servicio as TSV ON
SV.ID = TSV.ID
INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio = S.Codigo_de_socio
WHERE A.Fecha BETWEEN '2020-08-01 00:00:00' AND '2020-08-31 23:59:59' AND Descripcion_servicio = 'Primera consulta'
GROUP BY A.Codigo_de_socio;
```

	Primer_nombre	Primer_apellido	Cantidad_socios_primera_consulta
▶	Federico	Berrinzaghi	1
	Ana	Lopez	1
	Giorgio	Armani	1

Forma 2:

```
SELECT count(DISTINCT A.Codigo_de_socio) as Cantidad_socios_primera_consulta
FROM proyecto_final.atienden as A INNER JOIN proyecto_final.servicios as SV ON
A.Codigo_servicio = SV.Codigo_servicio INNER JOIN proyecto_final.tipo_de_servicio as TSV ON
SV.ID = TSV.ID
WHERE A.Fecha >= '2020-08-01 00:00:00' AND A.Fecha <='2020-08-31 23:59:59' AND Descripcion_servicio = 'Primera consulta';
```


Forma 3:

```
SELECT count(DISTINCT A.Codigo_de_socio) as Cantidad_socios_primera_consulta
FROM proyecto_final.atienden as A INNER JOIN proyecto_final.servicios as SV ON
A.Codigo_servicio = SV.Codigo_servicio INNER JOIN proyecto_final.tipo_de_servicio as TSV ON
SV.ID = TSV.ID
WHERE A.Fecha<= NOW() and A.Fecha >= DATE_SUB(NOW(), INTERVAL '1' MONTH) AND Descripcion_servicio = 'Primera consulta';
```

C. El proveedor que más medicamentos suministró a la empresa.

```
SELECT P.Nombre, count(*) as Cantidad_de_medicamentos_suministrados
FROM proyecto_final.suministran SM INNER JOIN proyecto_final.proveedores as P ON
SM.Codigo_del_proveedor = P.Codigo_del_proveedor INNER JOIN proyecto_final.medicamentos as M ON
SM.Codigo_medicamento = M.Codigo_medicamento
GROUP BY P.Nombre
ORDER BY Cantidad_de_medicamentos_suministrados DESC
Limit 1;
```

	Nombre	Cantidad_de_medicamentos_suministrados
►	Roemmers	3

D. El medicamento más recetado el último mes.

Supongo que el último mes es Agosto.

Esta consulta la realicé de tres formas distintas, obteniendo el mismo resultado.

Supongo que los tipos de servicios en donde se pueden recetar medicamentos son Primera consulta, Recibir medicamentos y Consulta tratamiento.

Forma 1:

```
SELECT M.Nombre_medicamento, count(*) as Cantidad_medicamentos
FROM proyecto_final.indican I INNER JOIN proyecto_final.doctores as D ON
I.Codigo_empleado = D.Codigo_empleado INNER JOIN proyecto_final.medicamentos as M ON
I.Codigo_medicamento = M.Codigo_medicamento INNER JOIN proyecto_final.servicios as SV ON
I.Codigo_servicio = SV.Codigo_servicio
INNER JOIN proyecto_final.tipo_de_servicio as TSV ON SV.ID = TSV.ID
WHERE I.Fecha BETWEEN '2020-08-01 00:00:00' AND '2020-08-31 23:59:59' AND TSV.Descripcion_servicio in ('Primera consulta', 'Recibir medicamentos', 'Consulta tratamiento')
GROUP BY M.Nombre_medicamento
ORDER BY Cantidad_medicamentos DESC
LIMIT 1;
```

	Nombre_medicamento	Cantidad_medicamentos
►	Dolex	2

Forma 2:

```
SELECT M.Nombre_medicamento, count(*) as Cantidad_medicamentos
FROM proyecto_final.indican I INNER JOIN proyecto_final.doctores as D ON
I.Codigo_empleado = D.Codigo_empleado INNER JOIN proyecto_final.medicamentos as M ON
I.Codigo_medicamento = M.Codigo_medicamento INNER JOIN proyecto_final.servicios as SV ON
I.Codigo_servicio = SV.Codigo_servicio
INNER JOIN proyecto_final.tipo_de_servicio as TSV ON SV.ID = TSV.ID
WHERE I.Fecha >= '2020-08-01 00:00:00' AND I.Fecha <= '2020-08-31 23:59:59' AND TSV.Descripcion_servicio in ('Primera consulta', 'Recibir medicamentos', 'Consulta tratamiento')
GROUP BY M.Nombre_medicamento
ORDER BY Cantidad_medicamentos DESC
LIMIT 1;
```

Forma 3:

```

SELECT M.Nombre_medimento, count(*) as Cantidad_medamentos
FROM proyecto_final.indican I INNER JOIN proyecto_final.doctores as D ON
I.Codigo_empleado = D.Codigo_empleado INNER JOIN proyecto_final.medamentos as M ON
I.Codigo_medimento = M.Codigo_medimento INNER JOIN proyecto_final.servicios as SV ON
I.Codigo_servicio = SV.Codigo_servicio
INNER JOIN proyecto_final.tipo_de_servicio as TSV ON SV.ID = TSV.ID
WHERE I.Fecha <= NOW() and I.Fecha >= DATE_SUB(NOW(), INTERVAL '1' MONTH) AND TSV.Descripcion_servicio in ('Primera consulta', 'Recibir medicamentos', 'Consulta tratamiento')
GROUP BY M.Nombre_medimento
ORDER BY Cantidad_medamentos DESC
LIMIT 1;

```

E. Nombre, apellido, teléfono/s y doctor cabecera de socios internados la última semana.

Supongo que la última semana es la semana que va del 16/09/2020 al 23/09/2020.

Esta consulta la realicé de tres formas distintas, obteniendo el mismo resultado.

Forma 1:

```

SELECT S.Primer_nombre as Primer_nombre_socio, S.Segundo_nombre as Segundo_nombre_socio, S.Primer_apellido as Primer_apellido_socio,
S.Segundo_apellido as Segundo_apellido_socio, T.telefono as Telefono_socio, E.Nombre as Nombre_doctor_cabecera, E.Apellido as Apellido_doctor_cabecera, A.Fecha
FROM proyecto_final.atienden A INNER JOIN proyecto_final.servicios as SV ON
A.Codigo_servicio = SV.Codigo_servicio
INNER JOIN proyecto_final.tipo_de_servicio as TSV ON SV.ID = TSV.ID
INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio = S.Codigo_de_socio
INNER JOIN proyecto_final.empleados as E ON A.Codigo_empleado = E.Codigo_empleado
INNER JOIN proyecto_final.telefono as T ON S.Codigo_de_socio = T.Codigo_de_socio
WHERE TSV.Descripcion_servicio in ('Agenda de cama', 'Operacion') AND A.Fecha BETWEEN '2020-09-16 00:00:00' AND '2020-09-23 23:59:59'
ORDER BY A.Fecha DESC;

```

	Primer_nombre_socio	Segundo_nombre_socio	Primer_apellido_socio	Segundo_apellido_socio	Telefono_socio	Nombre_doctor_cabecera	Apellido_doctor_cabecera	Fecha
▶	Ana	Maria	Lopez	Gomez	24184547	Jorge	Facal	2020-09-23 04:27:00
	Olivia	Maria	Fernandez	Perez	23365451	Jorge	Perez	2020-09-22 05:30:00
	Olivia	Maria	Fernandez	Perez	99282546	Jorge	Perez	2020-09-22 05:30:00
	Natalia	Mercedes	Martinez	Sanchez	93123456	Jorge	Perez	2020-09-20 06:56:00
	Elizabeth	Agustina	Iglesias	Cartier	24002233	Lara	Gonzalez	2020-09-18 03:33:00

Forma 2:

```

SELECT S.Primer_nombre as Primer_nombre_socio, S.Segundo_nombre as Segundo_nombre_socio, S.Primer_apellido as Primer_apellido_socio,
S.Segundo_apellido as Segundo_apellido_socio, T.telefono as Telefono_socio, E.Nombre as Nombre_doctor_cabecera, E.Apellido as Apellido_doctor_cabecera, A.Fecha
FROM proyecto_final.atienden A INNER JOIN proyecto_final.servicios as SV ON
A.Codigo_servicio = SV.Codigo_servicio
INNER JOIN proyecto_final.tipo_de_servicio as TSV ON SV.ID = TSV.ID
INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio = S.Codigo_de_socio
INNER JOIN proyecto_final.empleados as E ON A.Codigo_empleado = E.Codigo_empleado
INNER JOIN proyecto_final.telefono as T ON S.Codigo_de_socio = T.Codigo_de_socio
WHERE TSV.Descripcion_servicio in ('Agenda de cama', 'Operacion') AND A.Fecha >= '2020-09-16 00:00:00' AND A.Fecha <= '2020-09-23 23:59:59'
ORDER BY A.Fecha DESC;

```

Forma 3:

```

SELECT S.Primer_nombre as Primer_nombre_socio, S.Segundo_nombre as Segundo_nombre_socio, S.Primer_apellido as Primer_apellido_socio,
S.Segundo_apellido as Segundo_apellido_socio, T.telefono as Telefono_socio, E.Nombre as Nombre_doctor_cabecera, E.Apellido as Apellido_doctor_cabecera, A.Fecha
FROM proyecto_final.atienden A INNER JOIN proyecto_final.servicios as SV ON
A.Codigo_servicio = SV.Codigo_servicio
INNER JOIN proyecto_final.tipo_de_servicio as TSV ON SV.ID = TSV.ID
INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio = S.Codigo_de_socio
INNER JOIN proyecto_final.empleados as E ON A.Codigo_empleado = E.Codigo_empleado
INNER JOIN proyecto_final.telefono as T ON S.Codigo_de_socio = T.Codigo_de_socio
WHERE TSV.Descripcion_servicio in ('Agenda de cama', 'Operacion') and A.Fecha <= NOW() and A.Fecha >= DATE_SUB(NOW(), INTERVAL '1' WEEK)
ORDER BY A.Fecha DESC;

```

F. Que doctor atendió a un paciente X y en que tipo de consulta ¿Fue un doctor suplente o un titular (doctor cabecera del socio)?

Como la tabla Socios tiene la columna “Codigo_empleado” y la tabla Atienden también, hice un primer join “EE” para unir la tabla Socios con la tabla Empleados por dicha columna y así obtener la información de los doctores de cabecera de los pacientes y luego usé un segundo join “E” para unir la tabla Atienden con la tabla Empleados por la columna “Codigo_empleado” y así obtener la información de los doctores que atendieron a los pacientes en esa fecha determinada.

```
SELECT S.Primer_nombre as Primer_nombre_paciente, S.Segundo_nombre as Segundo_nombre_paciente, S.Primer_apellido as Primer_apellido_paciente,
S.Segundo_apellido as Segundo_apellido_paciente, E.Nombre as Nombre_doctor_que_atendió, E.Apellido as Apellido_doctor_que_atendió,
TSV.Descripcion_servicio as Tipo_de_servicio, EE.Nombre as Nombre_medico_cabecera, EE.Apellido as Apellido_medico_cabecera, A.Fecha
FROM proyecto_final.atienden A INNER JOIN proyecto_final.servicios as SV ON
A.Codigo_servicio = SV.Codigo_servicio
INNER JOIN proyecto_final.tipo_de_servicio as TSV ON SV.ID = TSV.ID
INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio = S.Codigo_de_socio
INNER JOIN proyecto_final.empleados as EE ON S.Codigo_empleado = EE.Codigo_empleado
INNER JOIN proyecto_final.empleados as E ON A.Codigo_empleado = E.Codigo_empleado
INNER JOIN proyecto_final.tipo_de_empleado as TE ON E.ID = TE.ID
ORDER BY A.Fecha DESC;
```

	Primer_nombre_paciente	Segundo_nombre_paciente	Primer_apellido_paciente	Segundo_apellido_paciente	Nombre_doctor_que_atendió	Apellido_doctor_que_atendió	Tipo_de_servicio	Nombre_medico_cabecera	Apellido_medico_cabecera	Fecha
▶	Alice	Paola	Leone	Ferrari	Lara	Gonzalez	Consulta tratamiento	Lara	Gonzalez	2020-10-18 16:21:00
	Sofia	Alba	Lombardi	Murphy	Jorge	Facal	Operacion	Paula	Gomez	2020-10-12 03:27:00
	Isabella	Cristina	Moretti	Walsh	Alan	Cortez	Consulta tratamiento	Alan	Cortez	2020-10-02 09:34:00
	Sara	Valeria	Santoro	Byrne	Paula	Gomez	Agenda de cama	Paula	Gomez	2020-09-30 22:02:00
	Alice	Paola	Leone	Ferrari	Lara	Gonzalez	Operacion	Lara	Gonzalez	2020-09-30 19:54:00
	Daniel	Vicente	Rodriguez	Perez	Juan	Ponce	Consulta tratamiento	Juan	Ponce	2020-09-29 14:22:00
	Federico	Leonardo	Berrinzaghi	Catino	Alan	Cortez	Operacion	Jorge	Perez	2020-09-27 02:22:00
	Isabella	Cristina	Moretti	Walsh	Alan	Cortez	Agenda de cama	Alan	Cortez	2020-09-26 12:32:00
	Ana	Maria	Lopez	Gomez	Jorge	Facal	Operacion	Jorge	Facal	2020-09-23 04:27:00
	Olivia	Maria	Fernandez	Perez	Jorge	Perez	Agenda de cama	Paula	Gomez	2020-09-22 06:30:00
	Natalia	Mercedes	Martinez	Sanchez	Jorge	Perez	Agenda de cama	Paula	Gomez	2020-09-20 06:56:00
	Elizabeth	Agustina	Iglesias	Cartier	Lara	Gonzalez	Agenda de cama	Jorge	Facal	2020-09-18 03:33:00
	Daniela	Carmen	Mosco	Sbros	Jorge	Facal	Agenda de cama	Jorge	Facal	2020-09-15 19:26:00
	Federico	Leonardo	Berrinzaghi	Catino	Jorge	Perez	Primera consulta	Jorge	Perez	2020-08-27 07:20:00
	Julio	Jose	Fernandez	Perez	Lara	Gonzalez	Operacion	Lara	Gonzalez	2020-08-22 23:50:00
	Giorgio		Armani	Raimondi	Juan	Ponce	Primera consulta	Juan	Ponce	2020-08-21 15:51:00
	Giorgio		Armani	Raimondi	Juan	Ponce	Primera consulta	Juan	Ponce	2020-08-11 09:57:00
	Ana	Maria	Lopez	Gomez	Lara	Gonzalez	Primera consulta	Jorge	Facal	2020-08-09 11:50:00
	Alessia	Alana	Contee	Ricci	Lara	Gonzalez	Operacion	Juan	Ponce	2020-08-09 07:50:00
	Federico	Leonardo	Berrinzaghi	Catino	Alan	Cortez	Primera consulta	Jorge	Perez	2020-08-07 15:50:00
	Sara	Valeria	Santoro	Byrne	Paula	Gomez	Recibir medicamentos	Paula	Gomez	2020-06-06 10:01:00
	Alessia	Alana	Contee	Ricci	Juan	Ponce	Recibir medicamentos	Juan	Ponce	2020-05-26 12:44:00
	Natalia	Mercedes	Martinez	Sanchez	Jorge	Perez	Consulta tratamiento	Paula	Gomez	2020-04-14 15:50:00
	Elizabeth	Agustina	Iglesias	Cartier	Lara	Gonzalez	Operacion	Jorge	Facal	2020-01-14 13:26:00
	Chiara	Cara	Ferragni	Serragni	Lara	Gonzalez	Primera consulta	Jorge	Facal	2019-12-27 08:50:00

Pacientes que fueron atendidos por su doctor de cabecera:

```
SELECT S.Primer_nombre as Primer_nombre_paciente, S.Segundo_nombre as Segundo_nombre_paciente, S.Primer_apellido as Primer_apellido_paciente,
S.Segundo_apellido as Segundo_apellido_paciente, TSV.Descripcion_servicio as Tipo_de_servicio, A.Fecha
FROM proyecto_final.atienden A INNER JOIN proyecto_final.servicios as SV ON
A.Codigo_servicio = SV.Codigo_servicio
INNER JOIN proyecto_final.tipo_de_servicio as TSV ON SV.ID = TSV.ID
INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio = S.Codigo_de_socio
INNER JOIN proyecto_final.empleados as EE ON S.Codigo_empleado = EE.Codigo_empleado
INNER JOIN proyecto_final.empleados as E ON A.Codigo_empleado = E.Codigo_empleado
INNER JOIN proyecto_final.tipo_de_empleado as TE ON E.ID = TE.ID
WHERE EE.Nombre = E.Nombre AND EE.Apellido = E.Apellido
ORDER BY A.Fecha DESC;
```

	Primer_nombre_paciente	Segundo_nombre_paciente	Primer_apellido_paciente	Segundo_apellido_paciente	Tipo_de_servicio	Fecha
▶	Alice	Paola	Leone	Ferrari	Consulta tratamiento	2020-10-18 16:21:00
	Isabella	Cristina	Moretti	Walsh	Consulta tratamiento	2020-10-02 09:34:00
	Sara	Valeria	Santoro	Byrne	Agenda de cama	2020-09-30 22:02:00
	Alice	Paola	Leone	Ferrari	Operacion	2020-09-30 19:54:00
	Daniel	Vicente	Rodriguez	Perez	Consulta tratamiento	2020-09-29 14:22:00
	Isabella	Cristina	Moretti	Walsh	Agenda de cama	2020-09-26 12:32:00
	Ana	Maria	Lopez	Gomez	Operacion	2020-09-23 04:27:00
	Daniela	Carmen	Mosco	Sbres	Agenda de cama	2020-09-15 19:26:00
	Federico	Leonardo	Berrinzaghi	Catino	Primera consulta	2020-08-27 07:20:00
	Julio	Jose	Fernandez	Perez	Operacion	2020-08-22 23:50:00
	Giorgio		Armani	Raimondi	Primera consulta	2020-08-21 15:51:00
	Giorgio		Armani	Raimondi	Primera consulta	2020-08-11 09:57:00
	Sara	Valeria	Santoro	Byrne	Recibir medicamentos	2020-06-06 10:01:00
	Alessia	Alana	Contee	Ricci	Recibir medicamentos	2020-05-26 12:44:00

Pacientes que fueron atendidos por un suplente:

```

SELECT S.Primer_nombre as Primer_nombre_paciente, S.Segundo_nombre as Segundo_nombre_paciente, S.Primer_apellido as Primer_apellido_paciente,
S.Segundo_apellido as Segundo_apellido_paciente, TSV.Descripcion_servicio as Tipo_de_servicio, A.Fecha
FROM proyecto_final.atienden A INNER JOIN proyecto_final.servicios as SV ON
A.Codigo_servicio = SV.Codigo_servicio
INNER JOIN proyecto_final.tipo_de_servicio as TSV ON SV.ID = TSV.ID
INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio = S.Codigo_de_socio
INNER JOIN proyecto_final.empleados as EE ON S.Codigo_empleado = EE.Codigo_empleado
INNER JOIN proyecto_final.empleados as E ON A.Codigo_empleado = E.Codigo_empleado
INNER JOIN proyecto_final.tipo_de_empleado as TE ON E.ID = TE.ID
WHERE EE.Nombre <> E.Nombre AND EE.Apellido <> E.Apellido
ORDER BY A.Fecha DESC;

```

	Primer_nombre_paciente	Segundo_nombre_paciente	Primer_apellido_paciente	Segundo_apellido_paciente	Tipo_de_servicio	Fecha
▶	Sofia	Alba	Lombardi	Murphy	Operacion	2020-10-12 03:27:00
	Federico	Leonardo	Berrinzaghi	Catino	Operacion	2020-09-27 02:22:00
	Olivia	Maria	Fernandez	Perez	Agenda de cama	2020-09-22 05:30:00
	Natalia	Mercedes	Martinez	Sanchez	Agenda de cama	2020-09-20 06:56:00
	Elizabeth	Agustina	Iglesias	Cartier	Agenda de cama	2020-09-18 03:33:00
	Ana	Maria	Lopez	Gomez	Primera consulta	2020-08-09 11:50:00
	Alessia	Alana	Contee	Ricci	Operacion	2020-08-09 07:50:00
	Federico	Leonardo	Berrinzaghi	Catino	Primera consulta	2020-08-07 15:50:00
	Natalia	Mercedes	Martinez	Sanchez	Consulta tratamiento	2020-04-14 15:50:00
	Elizabeth	Agustina	Iglesias	Cartier	Operacion	2020-01-14 13:26:00
	Chiara	Cara	Ferragni	Serragni	Primera consulta	2019-12-27 08:50:00

G. Listado de doctores que atendieron al menos tres socios el último mes.

Supongo que el último mes es Agosto.

Esta consulta la realicé de tres formas distintas, obteniendo el mismo resultado.

Forma 1:

```
SELECT E.Nombre as Nombre_doctor, E.Apellido as Apellido_doctor, count(*) as Cantidad_de_pacientes_atendidos, D.Especialidad
FROM proyecto_final.atienden A INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio = S.Codigo_de_socio
INNER JOIN proyecto_final.empleados as E ON A.Codigo_empleado = E.Codigo_empleado
INNER JOIN proyecto_final.tipo_de_empleado as TE ON E.ID = TE.ID
INNER JOIN proyecto_final.doctores as D ON E.Codigo_empleado = D.Codigo_empleado
WHERE A.Fecha BETWEEN '2020-08-01 00:00:00' AND '2020-08-31 23:59:59'
GROUP BY E.Codigo_empleado
HAVING count(*) >= 3;
```

	Nombre_doctor	Apellido_doctor	Cantidad_de_pacientes_atendidos	Especialidad
▶	Lara	Gonzalez	3	cirujano

Lo agrupé por “Codigo_empleado” porque es un valor único, no lo agrupé por nombre y apellido del doctor porque puede pasar que 2 doctores tengan el mismo nombre y apellido.

En el supuesto de que el nombre y apellido de los doctores fuera único, la consulta sería la siguiente:

```
SELECT E.Nombre as Nombre_doctor, E.Apellido as Apellido_doctor, count(*) as Cantidad_de_pacientes_atendidos, D.Especialidad
FROM proyecto_final.atienden A INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio = S.Codigo_de_socio
INNER JOIN proyecto_final.empleados as E ON A.Codigo_empleado = E.Codigo_empleado
INNER JOIN proyecto_final.tipo_de_empleado as TE ON E.ID = TE.ID
INNER JOIN proyecto_final.doctores as D ON E.Codigo_empleado = D.Codigo_empleado
WHERE A.Fecha BETWEEN '2020-08-01 00:00:00' AND '2020-08-31 23:59:59'
GROUP BY E.Nombre, E.Apellido
HAVING count(*) >= 3;
```

Forma 2:

```
SELECT E.Nombre as Nombre_doctor, E.Apellido as Apellido_doctor, count(*) as Cantidad_de_pacientes_atendidos, D.Especialidad
FROM proyecto_final.atienden A INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio = S.Codigo_de_socio
INNER JOIN proyecto_final.empleados as E ON A.Codigo_empleado = E.Codigo_empleado
INNER JOIN proyecto_final.tipo_de_empleado as TE ON E.ID = TE.ID
INNER JOIN proyecto_final.doctores as D ON E.Codigo_empleado = D.Codigo_empleado
WHERE A.Fecha <= '2020-08-31 23:59:59' AND A.Fecha >= '2020-08-01 00:00:00'
GROUP BY E.Codigo_empleado
HAVING count(*) >= 3;
```

Forma 3:

```
SELECT E.Nombre as Nombre_doctor, E.Apellido as Apellido_doctor, count(*) as Cantidad_de_pacientes_atendidos, D.Especialidad
FROM proyecto_final.atienden A INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio = S.Codigo_de_socio
INNER JOIN proyecto_final.empleados as E ON A.Codigo_empleado = E.Codigo_empleado
INNER JOIN proyecto_final.tipo_de_empleado as TE ON E.ID = TE.ID
INNER JOIN proyecto_final.doctores as D ON E.Codigo_empleado = D.Codigo_empleado
WHERE A.Fecha <= NOW() and A.Fecha >= DATE_SUB(NOW(), INTERVAL '1' MONTH)
GROUP BY E.Codigo_empleado
HAVING count(*) >= 3;
```

H. Socios que agendaron una primera consulta y no recibieron ningún otro servicio.

Agrupo por “Codigo_de_socio” porque es el valor que luego voy a contar y ver si no se repite, si se repite quiere decir que el socio recibió más de un servicio. El único caso posible en que el socio puede repetirse es cuando todas sus consultas fueron primeras consultas.

```
SELECT S.Primer_nombre as Primer_nombre_paciente, S.Segundo_nombre as Segundo_nombre_paciente, S.Primer_apellido as Primer_apellido_paciente,
S.Segundo_apellido as Segundo_apellido_paciente, TSV.Descripcion_servicio
FROM proyecto_final.atienden A INNER JOIN proyecto_final.servicios as SV ON A.Codigo_servicio = SV.Codigo_servicio
INNER JOIN proyecto_final.tipo_de_servicio as TSV ON SV.ID = TSV.ID
INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio = S.Codigo_de_socio
GROUP BY S.Codigo_de_socio
HAVING count(*)>=1 AND TSV.Descripcion_servicio = 'Primera consulta';
```

	Primer_nombre_paciente	Segundo_nombre_paciente	Primer_apellido_paciente	Segundo_apellido_paciente	Descripcion_servicio
▶	Giorgio		Armani	Raimondi	Primera consulta
	Chiara	Cara	Ferragni	Serragni	Primera consulta

ANEXOS

- **Consultas de creación de objetos de base de datos:**

```
CREATE TABLE proyecto_final.Salud_Uruguay(  
NIF int not null,  
Direccion varchar(50),  
Numero_de_puerta smallint,  
Telefono int not null,  
PRIMARY KEY (NIF)  
);
```

```
CREATE TABLE proyecto_final.proveedores(  
Codigo_del_proveedor int not null auto_increment,  
Nombre varchar(50),  
Direccion varchar(50),  
Numero_de_puerta smallint,  
Telefono int not null,  
NIF int not null,  
PRIMARY KEY (Codigo_del_proveedor),  
INDEX (Nombre),  
CONSTRAINT NIF_fk FOREIGN KEY (NIF) REFERENCES proyecto_final.salud_uruguay(NIF)  
);
```

```
CREATE TABLE proyecto_final.medicamentos(  
Codigo_medimento int not null auto_increment,  
Nombre_medimento varchar(30),  
Descripcion_medimento varchar(80),  
PRIMARY KEY (Codigo_medimento)  
);
```

```
CREATE TABLE proyecto_final.suministran(  
Codigo_del_proveedor int not null,  
Codigo_medicamento int not null,  
PRIMARY KEY (Codigo_del_proveedor, Codigo_medicamento),  
CONSTRAINT Codigo_del_proveedor_fork FOREIGN KEY (Codigo_del_proveedor) REFERENCES  
proyecto_final.proveedores(Codigo_del_proveedor),  
CONSTRAINT Codigo_medicamento_fk FOREIGN KEY (Codigo_medicamento) REFERENCES  
proyecto_final.medicamentos(Codigo_medicamento)  
);
```

```
CREATE TABLE proyecto_final.salas(  
Codigo_de_sala int not null auto_increment,  
Nombre_sala varchar(30),  
NIF int not null,  
PRIMARY KEY (Codigo_de_sala),  
CONSTRAINT NIF_foreignk FOREIGN KEY (NIF) REFERENCES proyecto_final.salud_uruguay(NIF)  
);
```

```
CREATE TABLE proyecto_final.tipo_de_servicio(  
ID int not null auto_increment,  
Descripcion_servicio varchar(50),  
PRIMARY KEY (ID)  
);
```

```
CREATE TABLE proyecto_final.servicios(  
Codigo_servicio int not null auto_increment,  
ID int not null,  
NIF int not null,  
PRIMARY KEY (Codigo_servicio),  
CONSTRAINT ID_fk FOREIGN KEY (ID) REFERENCES proyecto_final.tipo_de_servicio(ID),  
CONSTRAINT NIF_foreignkey FOREIGN KEY (NIF) REFERENCES  
proyecto_final.salud_uruguay(NIF)
```


);

```
CREATE TABLE proyecto_final.consulta_tratamiento(  
Codigo_servicio int not null,  
Motivo_del_tratamiento varchar(100),  
Descripcion_del_tratamiento varchar(100),  
PRIMARY KEY (Codigo_servicio),  
CONSTRAINT Codigo_servicio_fk FOREIGN KEY (Codigo_servicio) REFERENCES  
proyecto_final.servicios(Codigo_servicio)  
);
```

```
CREATE TABLE proyecto_final.recibir_medicamentos(  
Codigo_servicio int not null,  
Indicacion_de_dosificacion varchar(100),  
PRIMARY KEY (Codigo_servicio),  
CONSTRAINT Codigo_servicio_foreignk FOREIGN KEY (Codigo_servicio) REFERENCES  
proyecto_final.servicios(Codigo_servicio)  
);
```

```
CREATE TABLE proyecto_final.primera_consulta(  
Codigo_servicio int not null,  
Motivo_consulta varchar(50),  
PRIMARY KEY (Codigo_servicio),  
CONSTRAINT Codigo_servicio_foreignkey FOREIGN KEY (Codigo_servicio) REFERENCES  
proyecto_final.servicios(Codigo_servicio)  
);
```

```
CREATE TABLE proyecto_final.operacion(  
Codigo_servicio int not null,  
Motivo_de_operacion varchar(30),  
PRIMARY KEY (Codigo_servicio),
```

```
CONSTRAINT Codigo_servicio_forekey FOREIGN KEY (Codigo_servicio) REFERENCES  
proyecto_final.servicios(Codigo_servicio)  
);
```

```
CREATE TABLE proyecto_final.agenda_de_cama(  
Codigo_servicio int not null,  
Motivo_de_estadia varchar(80),  
PRIMARY KEY (Codigo_servicio),  
CONSTRAINT Codigo_servicio_fork FOREIGN KEY (Codigo_servicio) REFERENCES  
proyecto_final.servicios(Codigo_servicio)  
);
```

```
CREATE TABLE proyecto_final.tipo_de_empleado(  
ID int not null auto_increment,  
Descripcion_empleado varchar(50),  
PRIMARY KEY (ID)  
);
```

```
CREATE TABLE proyecto_final.empleados(  
Codigo_empleado int not null auto_increment,  
Nombre varchar(15),  
Apellido varchar(15),  
Direccion varchar(50),  
Numero_de_puerta smallint,  
Telefono int,  
ID int not null,  
NIF int not null,  
PRIMARY KEY (Codigo_empleado),  
CONSTRAINT ID_fkey FOREIGN KEY (ID) REFERENCES proyecto_final.tipo_de_empleado(ID),  
CONSTRAINT NIF_fkey FOREIGN KEY (NIF) REFERENCES proyecto_final.salud_uruguay(NIF)  
);
```

```
CREATE TABLE proyecto_final.servicio_de_limpieza(  
Codigo_empleado int not null,  
Cargo varchar(30),  
PRIMARY KEY (Codigo_empleado),  
CONSTRAINT Codigo_empleado_fkey FOREIGN KEY (Codigo_empleado) REFERENCES  
proyecto_final.empleados(Codigo_empleado)  
);
```

```
CREATE TABLE proyecto_final.enfermeros(  
Codigo_empleado int not null,  
Numero_de_habitacion_asignado int not null,  
PRIMARY KEY (Codigo_empleado),  
CONSTRAINT Codigo_empleado_fk FOREIGN KEY (Codigo_empleado) REFERENCES  
proyecto_final.empleados(Codigo_empleado)  
);
```

```
CREATE TABLE proyecto_final.doctores(  
Codigo_empleado int not null,  
Especialidad varchar(30),  
PRIMARY KEY (Codigo_empleado),  
CONSTRAINT Codigo_empleado_fork FOREIGN KEY (Codigo_empleado) REFERENCES  
proyecto_final.empleados(Codigo_empleado)  
);
```

```
CREATE TABLE proyecto_final.camas(  
Codigo_de_cama int not null auto_increment,  
Codigo_de_sala int not null,  
PRIMARY KEY (Codigo_de_cama),  
CONSTRAINT Codigo_de_sala_fk FOREIGN KEY (Codigo_de_sala) REFERENCES  
proyecto_final.salas(Codigo_de_sala)  
);
```

```
CREATE TABLE proyecto_final.vigilan(  
Codigo_empleado int not null,  
Codigo_de_cama int not null,  
Entrada datetime,  
Salida datetime,  
PRIMARY KEY (Codigo_empleado, Codigo_de_cama),  
CONSTRAINT Codigo_empleado_forkey FOREIGN KEY (Codigo_empleado) REFERENCES  
proyecto_final.empleados(Codigo_empleado),  
CONSTRAINT Codigo_de_cama_fk FOREIGN KEY (Codigo_de_cama) REFERENCES  
proyecto_final.camas(Codigo_de_cama)  
);
```

```
CREATE TABLE proyecto_final.socios(  
Codigo_de_socio int not null auto_increment,  
Primer_nombre varchar(15),  
Segundo_nombre varchar(15),  
Primer_apellido varchar(15),  
Segundo_apellido varchar(15),  
Direccion varchar(50),  
Numero_puerta smallint,  
Fecha_de_nacimiento date,  
Antiguedad smallint,  
Sexo bool,  
Codigo_empleado int not null,  
PRIMARY KEY (Codigo_de_socio),  
CONSTRAINT Codigo_empleado_foreignkey FOREIGN KEY (Codigo_empleado) REFERENCES  
proyecto_final.empleados(Codigo_empleado)  
);
```

```
CREATE TABLE proyecto_final.telefono(  
Codigo_de_socio int not null,  
Telefono int,
```

```
PRIMARY KEY (Codigo_de_socio,Telefono),  
  
CONSTRAINT Codigo_de_socio_foreignkey FOREIGN KEY (Codigo_de_socio) REFERENCES  
proyecto_final.socios(Codigo_de_socio)  
  
);
```

```
CREATE TABLE proyecto_final.atienden(  
  
Codigo_empleado int not null,  
  
Codigo_de_socio int not null,  
  
Codigo_servicio int not null,  
  
Fecha datetime,  
  
PRIMARY KEY (Codigo_empleado, Codigo_de_socio, Codigo_servicio),  
  
CONSTRAINT Codigo_empleado_foreignk FOREIGN KEY (Codigo_empleado) REFERENCES  
proyecto_final.empleados(Codigo_empleado),  
  
CONSTRAINT Codigo_de_socio_fkey FOREIGN KEY (Codigo_de_socio) REFERENCES  
proyecto_final.socios(Codigo_de_socio),  
  
CONSTRAINT Codigo_servicio_fkey FOREIGN KEY (Codigo_servicio) REFERENCES  
proyecto_final.servicios(Codigo_servicio)  
  
);
```

```
CREATE TABLE proyecto_final.requieren(  
  
Codigo_servicio int not null,  
  
Codigo_de_cama int not null,  
  
PRIMARY KEY (Codigo_servicio, Codigo_de_cama),  
  
CONSTRAINT Codigo_servicio_foreigkey FOREIGN KEY (Codigo_servicio) REFERENCES  
proyecto_final.servicios(Codigo_servicio),  
  
CONSTRAINT Codigo_de_cama_foreigkey FOREIGN KEY (Codigo_de_cama) REFERENCES  
proyecto_final.camas(Codigo_de_cama)  
  
);
```

```
CREATE TABLE proyecto_final.indican(  
  
Codigo_empleado int not null,  
  
Codigo_medimento int not null,  
  
Codigo_servicio int not null,
```

Fecha datetime,

PRIMARY KEY (Codigo_empleado, Codigo_medicamento, Codigo_servicio),

CONSTRAINT Codigo_empleado_forek FOREIGN KEY (Codigo_empleado) REFERENCES
proyecto_final.empleados(Codigo_empleado),

CONSTRAINT Codigo_medicamento_forek FOREIGN KEY (Codigo_medicamento) REFERENCES
proyecto_final.medicamentos(Codigo_medicamento),

CONSTRAINT Codigo_servicio_forek FOREIGN KEY (Codigo_servicio) REFERENCES
proyecto_final.servicios(Codigo_servicio)

);

- **Consultas de inserción de datos:**

```
insert into proyecto_final.salud_uruguay(NIF, Direccion, Numero_de_puerta, Telefono)
values (1001, 'Av.Dr.Luis Alberto de Herrera', 1390, 26221005);
```

```
insert into proyecto_final.proveedores(Codigo_del_proveedor, Nombre, Direccion,
Numero_de_puerta, Telefono,NIF)
values (1, 'Scienza', 'Av.Dr.Luis Alberto de Herrera', 1248, 219889000,1001), (2, 'Roemmers',
'Zelmar Michelini', 1230, 29029626,1001),
(3, 'Laboratorios Gautier', 'Democracia', 2132, 24010594,1001);
```

```
insert into proyecto_final.medicamentos(Codigo_medicamento, Nombre_medicamento,
Descripcion_medicamento)
values (1, 'Amoxidal 500', 'Antibiótico bactericida de amplio espectro'), (2, 'Dolex',
'Comprimidos que contienen paracetamol'),
(3, 'Dolex Flex', 'Contiene Orfenadrina que es un relajante muscular del musculo esquelético'),
(4, 'Priorin', 'Capsulas que contribuye al mantenimiento del cabello en condiciones normales'),
(5,'Colagenol C', 'Suplemento nutricional que ayuda al desarrollo muscular'), (6,'Plagrel',
'Prevenir en adultos la formación de coagulos sanguineos');
```

```
insert into proyecto_final.suministran(Codigo_del_proveedor, Codigo_medicamento)
values (1, 1),(2, 2),(2, 3),(3, 4),(1, 5), (2, 6);
```

```
insert into proyecto_final.salas(Codigo_de_sala, Nombre_sala, NIF)
values (1, 'Maternidad', 1001),(2, 'Emergencias', 1001),(3, 'Operaciones', 1001), (4, 'General',
1001);
```

```
insert into proyecto_final.tipo_de_servicio(ID, Descripcion_servicio)
values (1, 'Agenda de cama'),(2,'Recibir medicamentos'),(3,'Consulta
tratamiento'),(4,'Operacion'),(5,'Primera consulta');
```

```
insert into proyecto_final.servicios(Codigo_servicio, ID, NIF)
values (1, 1, 1001),(2, 2, 1001),(3, 3, 1001),(4, 4, 1001),(5, 5, 1001),(6, 1, 1001),(7, 2, 1001),(8,
3, 1001),(9, 4, 1001),(10, 5, 1001);
```

```
insert into proyecto_final.consulta_tratamiento(Codigo_servicio, Motivo_del_tratamiento,
Descripcion_del_tratamiento)
values (3, 'Otitis media aguda', 'Se le indica al paciente la ingesta de antibiotico del via oral'),
(8, 'Dislocacion de hombro','Se le ajusta el hombro al paciente y se le indica ingerir
paracetamol');
```

```
insert into proyecto_final.recibir_medicamentos(Codigo_servicio, Indicacion_de_dosificacion)
values (2, 'Amoxidal 500mg durante 2 semanas, 2 comprimidos por dia'),(7,'Dolex durante 1
semana cada 8 horas');
```

```
insert into proyecto_final.primera_consulta(Codigo_servicio, Motivo_consulta)
values (5, 'Dolor intenso en oido derecho'),(10,'Dolor lumbar');
```

```
insert into proyecto_final.operacion(Codigo_servicio, Motivo_de_operacion)
values (4, 'Frecuencia cardiaca anormal'),(9, 'Apendicitis');
```

```
insert into proyecto_final.agenda_de_cama(Codigo_servicio, Motivo_de_estadia)
values (1, 'Parto prematuro'),(6,'Rotura de apendice');
```

```
insert into proyecto_final.tipo_de_empleado(ID, Descripcion_empleado)
values (1, 'Enfermero'),(2,'Doctor'),(3,'Servicio limpieza');
```

```
insert into proyecto_final.empleados(Codigo_empleado, Nombre, Apellido, Direccion,  
Numero_de_puerta, Telefono, ID, NIF)
```

```
values (1, 'Juana', 'Rodriguez', '21 de Setiembre', 2179, 27117186, 1, 1001),
```

```
(2, 'Federico', 'Diaz', 'Av. Julio Herrera y Reissig', 565, 27142714, 1, 1001), (3, 'Maria', 'Perez', 'Av.  
Italia', 5775, 26017733, 1, 1001),
```

```
(4, 'Paula', 'Gomez', 'Orinoco', 4951, 26196857, 2, 1001), (5, 'Jorge', 'Perez', 'Orinoco', 4951,  
26196857, 2, 1001),
```

```
(6, 'Lara', 'Gonzalez', 'Av. 8 de Octubre', 2709, 24873375, 2, 1001), (7, 'Maria', 'Gonzalez', 'Jose  
Enrique Rodo', 1671, 24011010, 3, 1001),
```

```
(8, 'Mario', 'Sosa', '26 de Marzo', 3572, 26225077, 3, 1001), (9, 'Leonel', 'Lopez', 'Magallanes',  
1434, 24002859, 3, 1001),
```

```
(10, 'Juan', 'Ponce', 'Av. Italia', 5775, 26017733, 2, 1001), (11, 'Jorge', 'Facal', 'Av. Italia', 5800,  
26017834, 2, 1001),
```

```
(12, 'Alan', 'Cortez', 'Sarandi', 429, 29165016, 2, 1001);
```

```
insert into proyecto_final.servicio_de_limpieza(Codigo_empleado, Cargo)
```

```
values (1, 'Limpiador areas criticas'), (2, 'Encargada sector'), (3, 'Limpiador areas generales');
```

```
insert into proyecto_final.enfermeros(Codigo_empleado, Numero_de_habitacion_asignado)
```

```
values (1, 204), (2, 301), (3, 102);
```

```
insert into proyecto_final.doctores(Codigo_empleado, Especialidad)
```

```
values (4, 'internista'), (5, 'internista'), (6, 'cirujano'), (10, 'medicina general'), (11, 'cirujano'),  
(12, 'internista');
```

```
insert into proyecto_final.camas(Codigo_de_cama, Codigo_de_sala)
```

```
values (1, 1), (2, 2), (3, 3);
```

```
insert into proyecto_final.vigilan(Codigo_de_cama, Codigo_empleado, Entrada, Salida)
```

```
values (1, 1, '2020-08-11 17:37:00', '2020-04-22 17:57:00'), (2, 1, '2020-04-22 11:07:00', '2020-  
04-22 11:10:00'),
```

```
(3, 2, '2020-02-15 08:41:00', '2020-02-15 09:06:00');
```



```
insert into proyecto_final.socios(Codigo_de_socio, Primer_nombre, Segundo_nombre,
Primer_apellido, Segundo_apellido, Direccion, Numero_puerta, Fecha_de_nacimiento,
Antiguedad, Sexo, Codigo_empleado)
```

```
values (1, 'Olivia', 'Maria', 'Fernandez', 'Perez', 'Av. Joaquin Suarez', 2727, '1981-06-10', 5, 1,
4),(2, 'Julio', 'Jose', 'Fernandez', 'Perez', 'Av. Joaquin Suarez', 2727, '1989-12-11', 5, 0, 6),
(3, 'Natalia', 'Mercedes', 'Martinez', 'Sanchez', 'Av. 18 de Julio', 1078, '1991-11-11', 15, 1, 4),(4,
'Chiara', 'Cara', 'Ferragni', 'Serragni', 'Placido Ellauri', 3494, '1987-05-07', 30, 1, 11),
(5, 'Federico', 'Leonardo', 'Berrinzaghi', 'Catino', 'Av. 8 de Octubre', 3395, '1989-10-15', 24, 0, 5),
(6, 'Daniela', 'Carmen', 'Mosco', 'Sbres', 'Obligado', 955, '1989-12-27', 8, 1, 11),
(7, 'Ana', 'Maria', 'Lopez', 'Gomez', 'Av. Alfredo Arocena', 1584, '1966-06-06', 32, 1, 11),(8,
'Daniel', 'Vicente', 'Rodriguez', 'Perez', 'Av. Alfredo Arocena', 1584, '1977-07-07', 27, 0, 10),
(9, 'Elizabeth', 'Agustina', 'Iglesias', 'Cartier', 'Av. Gral. Jose Maria Paz', 1429, '1981-12-02', 2, 0,
11), (10, 'Sofia', 'Alba', 'Lombardi', 'Murphy', 'Blvr. Gral. Artigas', 3817, '2014-07-31', 3, 1, 4),
(11, 'Isabella', 'Cristina', 'Moretti', 'Walsh', '21 de Setiembre', 2701, '2006-03-13', 12, 1, 12),(12,
'Sara', 'Valeria', 'Santoro', 'Byrne', 'Av. Gral. Rondeau', 1437, '2011-11-05', 9, 1, 4),
(13, 'Alice', 'Paola', 'Leone', 'Ferrari', 'Colonia', 1651, '2017-02-28', 1, 1, 6),(14, 'Alessia', 'Alana',
'Contee', 'Ricci', 'Julio Herrera y Obes', 1466, '2004-04-17', 4, 1, 10),
(15, 'Giorgio', '', 'Armani', 'Raimondi', 'Joaquin Nuñez', 2787, '1934-07-11', 20, 0, 10);
```

```
insert into proyecto_final.telefono(Codigo_de_socio, Telefono)
```

```
values (1, 23365451), (1, 099282546), (2, 23365451), (2, 098417666), (3, 093123456), (4,
099214412),
(5, 092555444), (6, 27144545), (7, 24184547), (8, 24184547), (9, 24002233), (10, 22098076),
(10, 099222333),
(11, 27122202), (11, 094121212), (12, 29000803), (12, 092414547), (13, 24009071), (13,
095555555), (14, 20308701), (14, 099999999), (15, 092222222);
```

```
insert into proyecto_final.atienden(Codigo_empleado, Codigo_de_socio, Codigo_servicio,
Fecha)
```

```
values (5, 3, 3, '2020-04-14 15:50:00'), (6, 4, 10, '2019-12-27 08:50:00'),
(5, 5, 5, '2020-08-27 07:20:00'), (12, 5, 10, '2020-08-07 15:50:00'), (6, 2, 9, '2020-08-22 23:50:00'),
(11, 6, 6, '2020-09-15 19:26:00'), (10, 8, 3, '2020-09-29 14:22:00'), (6, 7, 10, '2020-08-09
11:50:00'), (6, 9, 4, '2020-01-14 13:26:00'),
(5, 1, 1, '2020-09-22 05:30:00'), (6, 9, 6, '2020-09-18 03:33:00'), (5, 3, 6, '2020-09-20 06:56:00'),
(11, 7, 9, '2020-09-23 04:27:00'),
(12, 5, 9, '2020-09-27 02:22:00'), (12, 11, 1, '2020-09-26 12:32:00'), (4, 12, 6, '2020-09-30
22:02:00'),
```

```
(6,13,9,'2020-09-30 19:54:00'), (12,11,8,'2020-10-02 09:34:00'), (11,10,9,'2020-10-12  
03:27:00'), (6,13,3,'2020-10-18 16:21:00'),  
  
(10,14,2,'2020-05-26 12:44:00'), (4,12,7,'2020-06-06 10:01:00'), (6,14,9,'2020-08-09 07:50:00'),  
  
(10,15,10,'2020-08-21 15:51:00'), (10,15,5,'2020-08-11 09:57:00');
```

```
insert into proyecto_final.requieren(Codigo_servicio, Codigo_de_cama)  
values (1, 1),(6,2), (4,3);
```

```
insert into proyecto_final.indican(Codigo_empleado, Codigo_medicamento, Codigo_servicio,  
Fecha)  
  
values (5,1,3,'2020-04-14 16:00:00'), (5,2,5,'2020-08-27 07:30:00'), (12,3,10,'2020-08-07  
15:58:00'), (10,1,3,'2020-09-29 14:30:00'),  
  
(12,3,8,'2020-10-02 09:47:00'), (10,1,2,'2020-05-26 12:49:00'), (4,2,7,'2020-06-06 10:09:00'),  
(10,2,2,'2020-08-20 10:46:00');
```

- **Cantidad de socios que tiene “SALUD_URUGUAY”.**

```
SELECT count(*) as Cantidad_de_socios  
  
FROM proyecto_final.socios;
```

- **Cantidad de socios distintos que realizaron una consulta por primera vez el último mes (agenda primer consulta).**

Forma 1:

```
SELECT count(DISTINCT A.Codigo_de_socio) as Cantidad_socios_primera_consulta  
  
FROM proyecto_final.atienden as A INNER JOIN proyecto_final.servicios as SV ON  
  
A.Codigo_servicio = SV.Codigo_servicio INNER JOIN proyecto_final.tipo_de_servicio as TSV ON  
SV.ID = TSV.ID  
  
WHERE A.Fecha BETWEEN '2020-08-01 00:00:00' AND '2020-08-31 23:59:59' AND  
Descripcion_servicio = 'Primera consulta';
```

Forma 2:

```
SELECT count(DISTINCT A.Codigo_de_socio) as Cantidad_socios_primera_consulta  
  
FROM proyecto_final.atienden as A INNER JOIN proyecto_final.servicios as SV ON  
  
A.Codigo_servicio = SV.Codigo_servicio INNER JOIN proyecto_final.tipo_de_servicio as TSV ON
```

SV.ID = TSV.ID

WHERE A.Fecha >= '2020-08-01 00:00:00' AND A.Fecha <='2020-08-31 23:59:59' AND
Descripcion_servicio = 'Primera consulta';

Forma 3:

SELECT count(DISTINCT A.Codigo_de_socio) as Cantidad_socios_primera_consulta
FROM proyecto_final.atienden as A INNER JOIN proyecto_final.servicios as SV ON
A.Codigo_servicio = SV.Codigo_servicio INNER JOIN proyecto_final.tipo_de_servicio as TSV ON
SV.ID = TSV.ID

WHERE A.Fecha<= NOW() and A.Fecha >= DATE_SUB(NOW(), INTERVAL '1' MONTH) AND
Descripcion_servicio = 'Primera consulta';

Detalle de a qué pacientes corresponden esos tres servicios:

SELECT S.Primer_nombre, S.Primer_apellido, count(DISTINCT A.Codigo_de_socio) as
Cantidad_socios_primera_consulta
FROM proyecto_final.atienden as A INNER JOIN proyecto_final.servicios as SV ON
A.Codigo_servicio = SV.Codigo_servicio INNER JOIN proyecto_final.tipo_de_servicio as TSV ON
SV.ID = TSV.ID

INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio = S.Codigo_de_socio

WHERE A.Fecha >= '2020-08-01 00:00:00' AND A.Fecha <='2020-08-31 23:59:59' AND
Descripcion_servicio = 'Primera consulta'

GROUP BY A.Codigo_de_socio;

- **El proveedor que más medicamentos suministró a la empresa.**

SELECT P.Nombre, count(*) as Cantidad_de_medicamentos_suministrados
FROM proyecto_final.suministran SM INNER JOIN proyecto_final.proveedores as P ON
SM.Codigo_del_proveedor = P.Codigo_del_proveedor INNER JOIN
proyecto_final.medicamentos as M ON
SM.Codigo_medicamento = M.Codigo_medicamento

GROUP BY P.Nombre

ORDER BY Cantidad_de_medicamentos_suministrados DESC

Limit 1;

- El medicamento más recetado el último mes.

Forma 1:

```
SELECT M.Nombre_medicamento, count(*) as Cantidad_medicamentos
FROM proyecto_final.indican I INNER JOIN proyecto_final.doctores as D ON
I.Codigo_empleado = D.Codigo_empleado INNER JOIN proyecto_final.medicamentos as M ON
I.Codigo_medicamento = M.Codigo_medicamento INNER JOIN proyecto_final.servicios as SV
ON I.Codigo_servicio = SV.Codigo_servicio
INNER JOIN proyecto_final.tipo_de_servicio as TSV ON SV.ID = TSV.ID
WHERE I.Fecha BETWEEN '2020-08-01 00:00:00' AND '2020-08-31 23:59:59' AND
TSV.Descripcion_servicio in ('Primera consulta', 'Recibir medicamentos', 'Consulta
tratamiento')
GROUP BY M.Nombre_medicamento
ORDER BY Cantidad_medicamentos DESC
LIMIT 1;
```

Forma 2:

```
SELECT M.Nombre_medicamento, count(*) as Cantidad_medicamentos
FROM proyecto_final.indican I INNER JOIN proyecto_final.doctores as D ON
I.Codigo_empleado = D.Codigo_empleado INNER JOIN proyecto_final.medicamentos as M ON
I.Codigo_medicamento = M.Codigo_medicamento INNER JOIN proyecto_final.servicios as SV
ON I.Codigo_servicio = SV.Codigo_servicio
INNER JOIN proyecto_final.tipo_de_servicio as TSV ON SV.ID = TSV.ID
WHERE I.Fecha >= '2020-08-01 00:00:00' AND I.Fecha <= '2020-08-31 23:59:59' AND
TSV.Descripcion_servicio in ('Primera consulta', 'Recibir medicamentos', 'Consulta
tratamiento')
GROUP BY M.Nombre_medicamento
ORDER BY Cantidad_medicamentos DESC
LIMIT 1;
```

Forma 3:

```
SELECT M.Nombre_medicamento, count(*) as Cantidad_medicamentos
FROM proyecto_final.indican I INNER JOIN proyecto_final.doctores as D ON
I.Codigo_empleado = D.Codigo_empleado INNER JOIN proyecto_final.medicamentos as M ON
```

```
I.Codigo_medicamento = M.Codigo_medicamento INNER JOIN proyecto_final.servicios as SV  
ON I.Codigo_servicio = SV.Codigo_servicio
```

```
INNER JOIN proyecto_final.tipo_de_servicio as TSV ON SV.ID = TSV.ID
```

```
WHERE I.Fecha <= NOW() and I.Fecha >= DATE_SUB(NOW(), INTERVAL '1' MONTH) AND  
TSV.Descripcion_servicio in ('Primera consulta', 'Recibir medicamentos', 'Consulta  
tratamiento')
```

```
GROUP BY M.Nombre_medicamento
```

```
ORDER BY Cantidad_medicamentos DESC
```

```
LIMIT 1;
```

- **Nombre, apellido, teléfono/s y doctor cabecera de socios internados la última semana.**

Forma 1:

```
SELECT S.Primer_nombre as Primer_nombre_socio, S.Segundo_nombre as  
Segundo_nombre_socio, S.Primer_apellido as Primer_apellido_socio,  
S.Segundo_apellido as Segundo_apellido_socio, T.telefono as Telefono_socio, E.Nombre as  
Nombre_doctor_cabecera, E.Apellido as Apellido_doctor_cabecera, A.Fecha  
FROM proyecto_final.atienden A INNER JOIN proyecto_final.servicios as SV ON  
A.Codigo_servicio = SV.Codigo_servicio  
INNER JOIN proyecto_final.tipo_de_servicio as TSV ON SV.ID = TSV.ID  
INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio = S.Codigo_de_socio  
INNER JOIN proyecto_final.empleados as E ON A.Codigo_empleado = E.Codigo_empleado  
INNER JOIN proyecto_final.telefono as T ON S.Codigo_de_socio = T.Codigo_de_socio  
WHERE TSV.Descripcion_servicio in ('Agenda de cama', 'Operacion') AND A.Fecha BETWEEN  
'2020-09-16 00:00:00' AND '2020-09-23 23:59:59'  
ORDER BY A.Fecha DESC;
```

Forma 2:

```
SELECT S.Primer_nombre as Primer_nombre_socio, S.Segundo_nombre as  
Segundo_nombre_socio, S.Primer_apellido as Primer_apellido_socio,  
S.Segundo_apellido as Segundo_apellido_socio, T.telefono as Telefono_socio, E.Nombre as  
Nombre_doctor_cabecera, E.Apellido as Apellido_doctor_cabecera, A.Fecha  
FROM proyecto_final.atienden A INNER JOIN proyecto_final.servicios as SV ON  
A.Codigo_servicio = SV.Codigo_servicio  
INNER JOIN proyecto_final.tipo_de_servicio as TSV ON SV.ID = TSV.ID  
INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio = S.Codigo_de_socio
```

```
INNER JOIN proyecto_final.empleados as E ON A.Codigo_empleado = E.Codigo_empleado
INNER JOIN proyecto_final.telefono as T ON S.Codigo_de_socio = T.Codigo_de_socio
WHERE TSV.Descripcion_servicio in ('Agenda de cama', 'Operacion') AND A.Fecha >= '2020-09-16 00:00:00' AND A.Fecha <='2020-09-23 23:59:59'
ORDER BY A.Fecha DESC;
```

Forma 3:

```
SELECT S.Primer_nombre as Primer_nombre_socio, S.Segundo_nombre as
Segundo_nombre_socio, S.Primer_apellido as Primer_apellido_socio,
S.Segundo_apellido as Segundo_apellido_socio, T.telefono as Telefono_socio, E.Nombre as
Nombre_doctor_cabecera, E.Apellido as Apellido_doctor_cabecera, A.Fecha
FROM proyecto_final.atienden A INNER JOIN proyecto_final.servicios as SV ON
A.Codigo_servicio = SV.Codigo_servicio
INNER JOIN proyecto_final.tipo_de_servicio as TSV ON SV.ID = TSV.ID
INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio = S.Codigo_de_socio
INNER JOIN proyecto_final.empleados as E ON A.Codigo_empleado = E.Codigo_empleado
INNER JOIN proyecto_final.telefono as T ON S.Codigo_de_socio = T.Codigo_de_socio
WHERE TSV.Descripcion_servicio in ('Agenda de cama', 'Operacion') and A.Fecha<= NOW() and
A.Fecha >= DATE_SUB(NOW(), INTERVAL '1' WEEK)
ORDER BY A.Fecha DESC;
```

- **Que doctor atendió a un paciente X y en que tipo de consulta ¿Fue un doctor suplente o un titular (doctor cabecera del socio)?**

```
SELECT S.Primer_nombre as Primer_nombre_paciente, S.Segundo_nombre as
Segundo_nombre_paciente, S.Primer_apellido as Primer_apellido_paciente,
S.Segundo_apellido as Segundo_apellido_paciente, E.Nombre as
Nombre_doctor_que_atendió, E.Apellido as Apellido_doctor_que_atendió,
TSV.Descripcion_servicio as Tipo_de_servicio, EE.Nombre as Nombre_medico_cabecera,
EE.Apellido as Apellido_medico_cabecera, A.Fecha
FROM proyecto_final.atienden A INNER JOIN proyecto_final.servicios as SV ON
A.Codigo_servicio = SV.Codigo_servicio
INNER JOIN proyecto_final.tipo_de_servicio as TSV ON SV.ID = TSV.ID
INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio = S.Codigo_de_socio
INNER JOIN proyecto_final.empleados as EE ON S.Codigo_empleado = EE.Codigo_empleado
```

```
INNER JOIN proyecto_final.empleados as E ON A.Codigo_empleado = E.Codigo_empleado
INNER JOIN proyecto_final.tipo_de_empleado as TE ON E.ID = TE.ID
ORDER BY A.Fecha DESC;
```

Pacientes que fueron atendidos por su doctor de cabecera:

```
SELECT S.Primer_nombre as Primer_nombre_paciente, S.Segundo_nombre as
Segundo_nombre_paciente, S.Primer_apellido as Primer_apellido_paciente,
S.Segundo_apellido as Segundo_apellido_paciente, TSV.Descripcion_servicio as
Tipo_de_servicio, A.Fecha
FROM proyecto_final.atienden A INNER JOIN proyecto_final.servicios as SV ON
A.Codigo_servicio = SV.Codigo_servicio
INNER JOIN proyecto_final.tipo_de_servicio as TSV ON SV.ID = TSV.ID
INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio = S.Codigo_de_socio
INNER JOIN proyecto_final.empleados as EE ON S.Codigo_empleado = EE.Codigo_empleado
INNER JOIN proyecto_final.empleados as E ON A.Codigo_empleado = E.Codigo_empleado
INNER JOIN proyecto_final.tipo_de_empleado as TE ON E.ID = TE.ID
WHERE EE.Nombre = E.Nombre AND EE.Apellido = E.Apellido
ORDER BY A.Fecha DESC;
```

Pacientes que fueron atendidos por un suplente:

```
SELECT S.Primer_nombre as Primer_nombre_paciente, S.Segundo_nombre as
Segundo_nombre_paciente, S.Primer_apellido as Primer_apellido_paciente,
S.Segundo_apellido as Segundo_apellido_paciente, TSV.Descripcion_servicio as
Tipo_de_servicio, A.Fecha
FROM proyecto_final.atienden A INNER JOIN proyecto_final.servicios as SV ON
A.Codigo_servicio = SV.Codigo_servicio
INNER JOIN proyecto_final.tipo_de_servicio as TSV ON SV.ID = TSV.ID
INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio = S.Codigo_de_socio
INNER JOIN proyecto_final.empleados as EE ON S.Codigo_empleado = EE.Codigo_empleado
INNER JOIN proyecto_final.empleados as E ON A.Codigo_empleado = E.Codigo_empleado
INNER JOIN proyecto_final.tipo_de_empleado as TE ON E.ID = TE.ID
WHERE EE.Nombre <> E.Nombre AND EE.Apellido <> E.Apellido
```

ORDER BY A.Fecha DESC;

- **Listado de doctores que atendieron al menos tres socios el último mes.**

Forma 1:

```
SELECT E.Nombre as Nombre_doctor, E.Apellido as Apellido_doctor, count(*) as
Cantidad_de_pacientes_atendidos, D.Especialidad

FROM proyecto_final.atienden A INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio
= S.Codigo_de_socio

INNER JOIN proyecto_final.empleados as E ON A.Codigo_empleado = E.Codigo_empleado

INNER JOIN proyecto_final.tipo_de_empleado as TE ON E.ID = TE.ID

INNER JOIN proyecto_final.doctores as D ON E.Codigo_empleado = D.Codigo_empleado

WHERE A.Fecha BETWEEN '2020-08-01 00:00:00' AND '2020-08-31 23:59:59'

GROUP BY E.Codigo_empleado

HAVING count(*) >= 3;
```

Forma 2:

```
SELECT E.Nombre as Nombre_doctor, E.Apellido as Apellido_doctor, count(*) as
Cantidad_de_pacientes_atendidos, D.Especialidad

FROM proyecto_final.atienden A INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio
= S.Codigo_de_socio

INNER JOIN proyecto_final.empleados as E ON A.Codigo_empleado = E.Codigo_empleado

INNER JOIN proyecto_final.tipo_de_empleado as TE ON E.ID = TE.ID

INNER JOIN proyecto_final.doctores as D ON E.Codigo_empleado = D.Codigo_empleado

WHERE A.Fecha <= '2020-08-31 23:59:59' AND A.Fecha >= '2020-08-01 00:00:00'

GROUP BY E.Codigo_empleado

HAVING count(*) >= 3;
```

Forma 3:

```
SELECT E.Nombre as Nombre_doctor, E.Apellido as Apellido_doctor, count(*) as
Cantidad_de_pacientes_atendidos, D.Especialidad

FROM proyecto_final.atienden A INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio
= S.Codigo_de_socio

INNER JOIN proyecto_final.empleados as E ON A.Codigo_empleado = E.Codigo_empleado

INNER JOIN proyecto_final.tipo_de_empleado as TE ON E.ID = TE.ID

INNER JOIN proyecto_final.doctores as D ON E.Codigo_empleado = D.Codigo_empleado
```



```
WHERE A.Fecha <= NOW() and A.Fecha >= DATE_SUB(NOW(), INTERVAL '1' MONTH)

GROUP BY E.Codigo_empleado

HAVING count(*) >= 3;
```

Supuesto de que el nombre y apellido de los doctores fuera único:

```
SELECT E.Nombre as Nombre_doctor, E.Apellido as Apellido_doctor, count(*) as
Cantidad_de_pacientes_atendidos, D.Especialidad

FROM proyecto_final.atienden A INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio
= S.Codigo_de_socio

INNER JOIN proyecto_final.empleados as E ON A.Codigo_empleado = E.Codigo_empleado

INNER JOIN proyecto_final.tipo_de_empleado as TE ON E.ID = TE.ID

INNER JOIN proyecto_final.doctores as D ON E.Codigo_empleado = D.Codigo_empleado

WHERE A.Fecha BETWEEN '2020-08-01 00:00:00' AND '2020-08-31 23:59:59'

GROUP BY E.Nombre, E.Apellido

HAVING count(*) >= 3;
```

- **Socios que agendaron una primera consulta y no recibieron ningún otro servicio.**

```
SELECT S.Primer_nombre as Primer_nombre_paciente, S.Segundo_nombre as
Segundo_nombre_paciente, S.Primer_apellido as Primer_apellido_paciente,
S.Segundo_apellido as Segundo_apellido_paciente, TSV.Descripcion_servicio

FROM proyecto_final.atienden A INNER JOIN proyecto_final.servicios as SV ON
A.Codigo_servicio = SV.Codigo_servicio

INNER JOIN proyecto_final.tipo_de_servicio as TSV ON SV.ID = TSV.ID

INNER JOIN proyecto_final.socios as S ON A.Codigo_de_socio = S.Codigo_de_socio

GROUP BY S.Codigo_de_socio

HAVING count(*)>=1 AND TSV.Descripcion_servicio = 'Primera consulta';
```