

# Fernando Rivero

Segunda pre-entrega SQL



# Presentación proyecto

**Modelo de negocio:** consultorio de kinesiología

1 Propietaria única.

6 Profesionales de salud

1 Secretario

Atención de prepagas

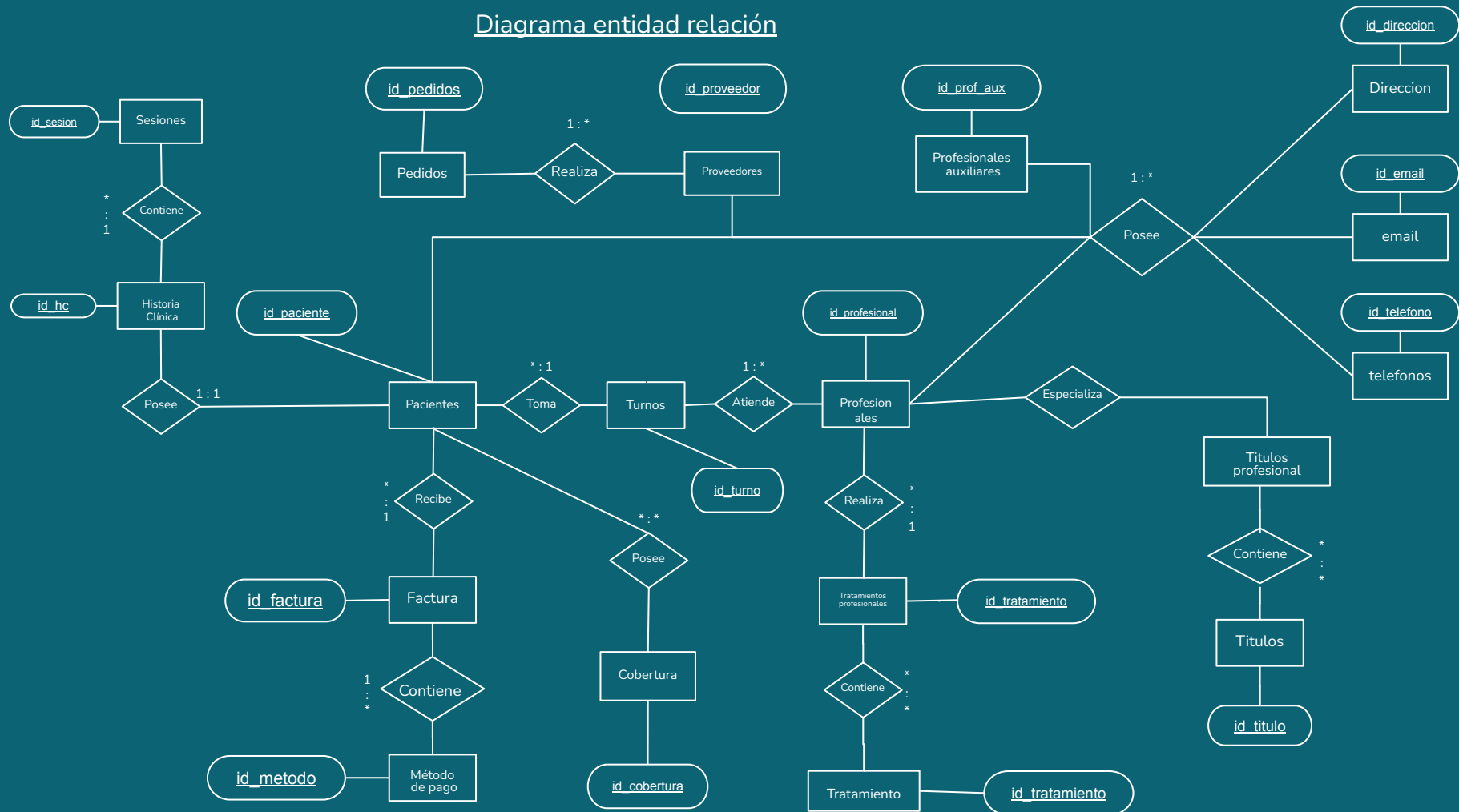
40 turnos aprox por día

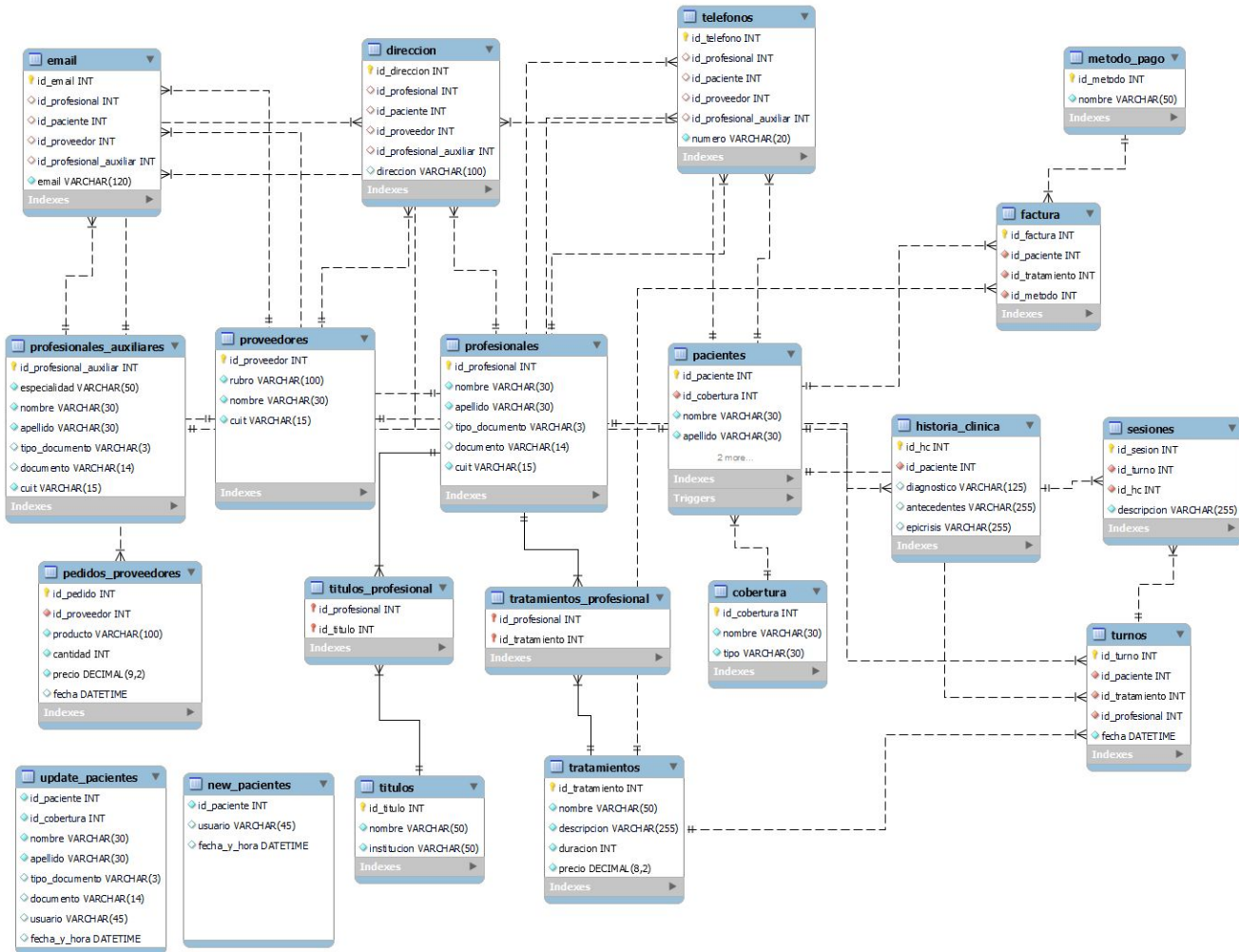
**Objetivo:** desarrollo de un sistema de base datos con el fin de almacenar y gestionar información del consultorio de modo tal que facilite la concreción de acciones propias y la obtención de métricas de forma sencilla.

**Motivo:** reforzar a través de datos varios aspectos del consultorio:

- 1) Datos y métricas para toma de decisiones
- 2) Gestión de hechos inherentes al consultorio
- 3) Control de procesos

## Diagrama entidad relación





# Vistas

**cobertura\_view:** permite obtener métricas sobre la cantidad de pacientes por cobertura. Combina información de las tablas "cobertura" y "pacientes".

**pagos\_view:** permite obtener métricas sobre la cantidad de veces que se utiliza cada método de pago. Combina las tablas "metodo\_pago" y "factura"

**pedidos\_proveedores\_view:** Detalle de pedidos a proveedores incluyendo cálculo del total. Permite obtener a través de métricas rubros de mayor gasto en insumos. Combina las tablas "proveedores" y "pedido\_proveedores"

**historia\_clinica\_view:** muestra en forma cronológica un resumen de la historia clínica del paciente. Combina las siguientes tablas: historia\_clinica, pacientes, turnos, tratamientos, profesionales.

**turno\_view:** Vista de información detallada de turnos en orden cronológico. Permite realizar un seguimiento de los mismos. Combina las siguientes tablas: pacientes, profesionales, tratamientos y turnos

# Funciones

**calcular\_iva:** permite calcular el monto del IVA a partir del monto bruto de tratamiento para la facturación. Recibe por parámetro el monto bruto (tipo decimal) sobre el cual se calculará el 21%.

Su objetivo es calcular el total de facturación al cliente en base al monto de bruto del precio del tratamiento.

ej de uso: `SELECT calcular_iva (500) AS IVA`

**caja\_diaria:** permite calcular el total de ingresos de una determinada fecha. Recibe por parámetro una fecha (tipo: date) y devuelve el total recaudado ese día (tipo: decimal)

Su objetivo es obtener métricas y facilitar la contabilidad.

`SELECT caja_diaria('2023-07-23') AS caja;`

# Stored procedures

**sp\_pacientes\_orden:** permite consultar una tabla ordenándola a través de dos parámetros: "campo" y "ord"

El parámetro campo responde al nombre de una columna de la tabla consultorio.pacientes.

El parámetro ord indica si se quiere el tipo de ordenamiento en forma ascendente o descendente. A tal fin sólo admite "ASC" o "DESC"

Vinculado a la tabla pacientes

Ej de uso: `Call consultorio.sp_pacientes_orden('nombre', 'ASC');`

**sp\_insertar\_tratamientos:** su objetivo es la inserción de registros en la tabla tratamientos, realizando un procedimiento de verificación que evite errores y simplifique al cliente el procedimiento.

Vinculado a la tabla tratamientos

Ej de uso: `CALL sp_insertar_tratamientos('masaje', 'Limpieza profunda de los dientes', 45, 150);`

# Triggers

`tr_add_new_pacientes:`

Tipo: AFTER INSERT

Tabla vinculada: Pacientes

Crea un log en la tabla new\_pacientes donde se almacena el id del paciente, usuario y fecha de la acción. Su objetivo es llevar un control en el proceso de creación de pacientes mediante su registro.

`tr_update_pacientes:`

Tipo: BEFORE UPDATE

Tabla vinculada: Pacientes

Crea un log en la tabla update\_pacientes donde se almacena la información de un paciente antes de realizarse modificaciones sobre el mismo. Su objetivo es llevar un control en el proceso de actualización y generar un respaldo de la información ante tal acción