

# Observatorio para medir avances de un plan de desarrollo

Proyecto ingeniería de datos  
Laura Sofia Salazar Reyes



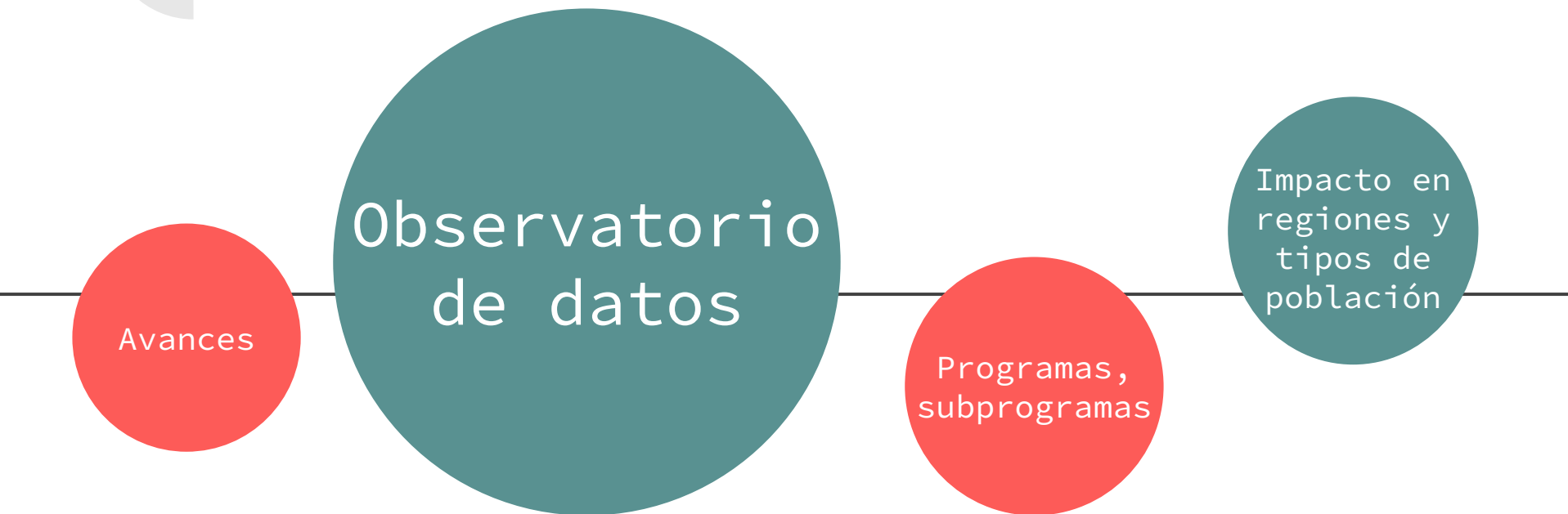


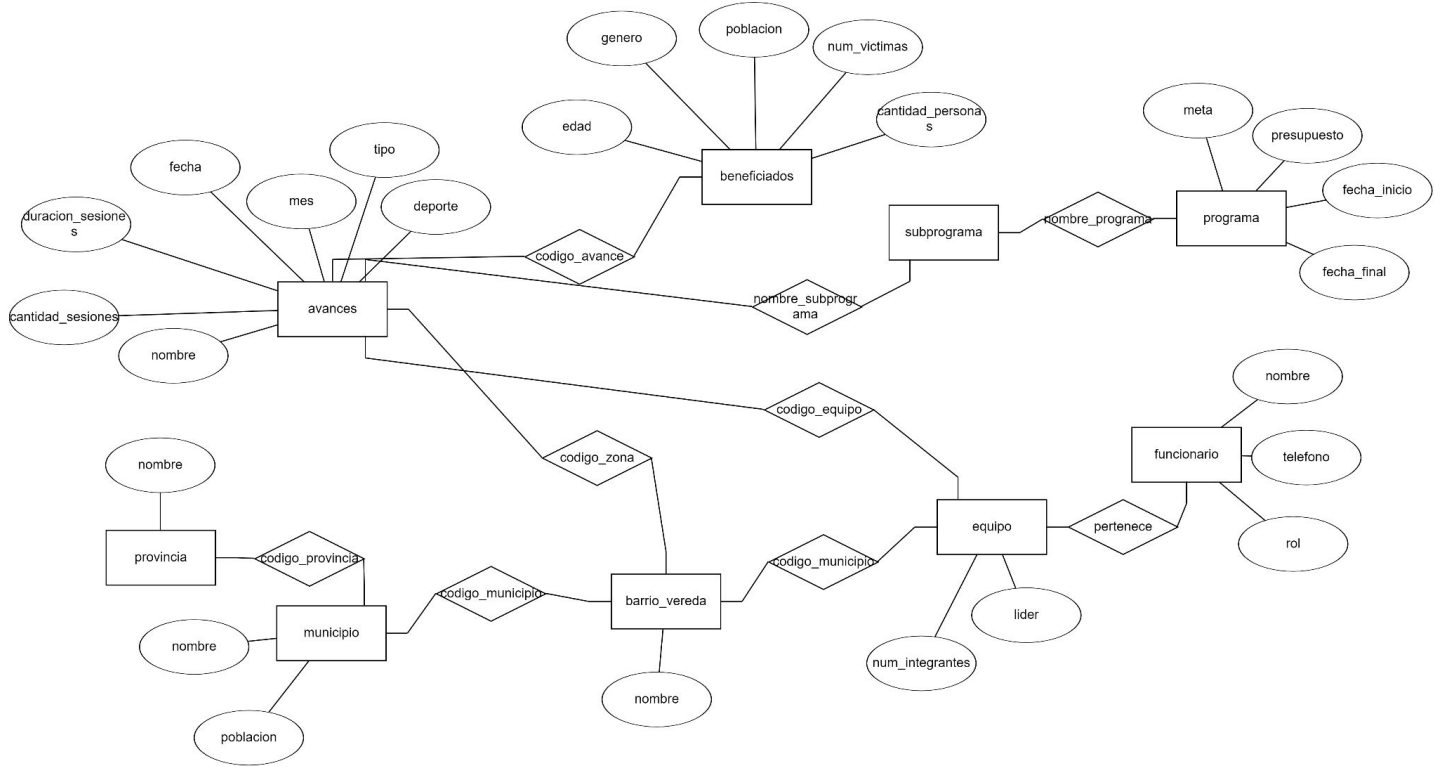
# Problema

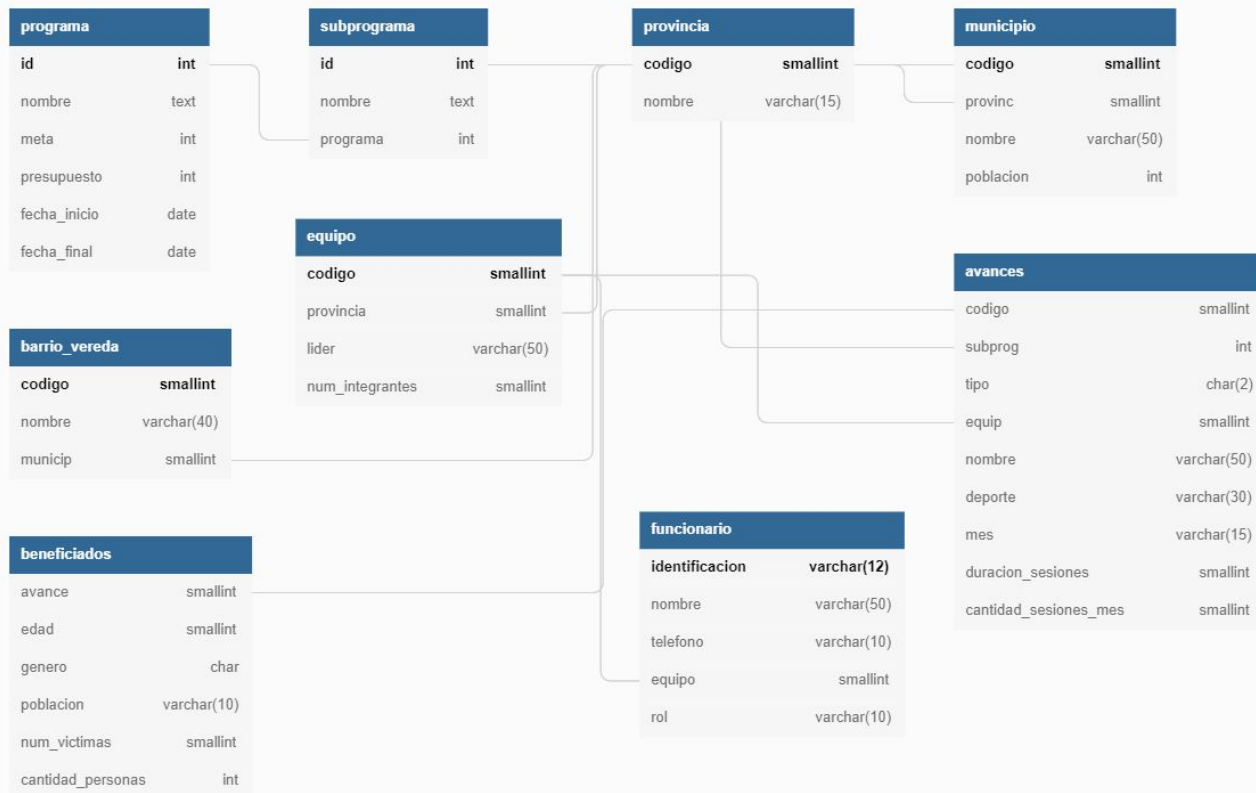
- Manejo de cantidades muy grandes de datos
- Dificultad al ingresar, visualizar y analizar los datos
- Avance y efectividad en el plan de desarrollo



# Solución







# Creación tablas

```
create table programa (  
  id int primary key,  
  nombre text,  
  meta int,  
  presupuesto int,  
  fecha_inicio date,  
  fecha_final date  
);
```

```
create table subprograma (  
  id int primary key,  
  nombre text,  
  programa int references programa(id)  
);
```

```
create table provincia (  
  codigo smallint primary key,  
  nombre varchar(15) not null  
);
```

```
create table municipio (  
  codigo smallint primary key,  
  provinc smallint references provincia(codigo),  
  nombre varchar(50) not null,  
  poblacion int not null  
);
```

```
create table barrio_vereda (  
  codigo smallint primary key,  
  nombre varchar(40),  
  municip smallint references municipio(codigo)  
);
```

```
create table equipo (  
  codigo smallint primary key,  
  provincia smallint not null references provincia(codigo),  
  líder varchar(50) not null,  
  num_integrantes smallint  
);
```

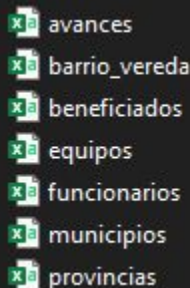
```
create table funcionario (  
  identificacion varchar(12) primary key,  
  nombre varchar(50) not null,  
  telefono varchar(10),  
  equipo smallint references equipo(codigo),  
  rol varchar(10)  
);
```

```
create table avances (  
  codigo smallint unique,  
  subprog int REFERENCES subprograma(id),  
  tipo char(2) not null check(tipo in ('ev', 'pc')),  
  equip smallint references equipo(codigo),  
  zona smallint references barrio_vereda(codigo),  
  nombre varchar(50),  
  deporte varchar(30),  
  mes varchar(15),  
  duracion_sesiones smallint,  
  cantidad_sesiones_mes smallint,  
  primary key(codigo, subprog, tipo, equip, zona)  
);
```

```
create table beneficiados(  
  avance smallint references avances(codigo),  
  edad smallint not null,  
  genero char check (genero in ('m','f')),  
  poblacion varchar(10) check(poblacion in ('mestizos','indig','campes','disc','afro')),  
  num_victimas smallint,  
  cantidad_personas int,  
  primary key(avance, edad, genero)  
);
```

# Inserción datos

```
copy provincia from 'C:/psg/provincias.csv' delimiter ';' csv header;  
copy municipio from 'C:/psg/municipios.csv' delimiter ';' csv header;  
copy barrio_vereda from 'C:/psg/barrio_vereda.csv' delimiter ';' csv header;  
copy equipo from 'C:/psg/equipos.csv' delimiter ';' csv header;  
copy funcionario from 'C:/psg/funcionarios.csv' delimiter ';' csv header;  
copy avances from 'C:/psg/avances.csv' delimiter ';' csv header;  
copy beneficiados from 'C:/psg/beneficiados.csv' delimiter ';' csv header;
```



- avances
- barrio\_vereda
- beneficiados
- equipos
- funcionarios
- municipios
- provincias

# Creación roles

```
create role gerente;  
grant select, insert, delete, update on all tables  
to gerente;
```

```
create role lider_equipo;  
grant select, insert, update on avances, beneficiados  
to lider_equipo;
```

```
create table avances_updates_log (  
  usuario varchar(40),  
  update_time timestamp with time zone,  
  update_log avances  
);  
  
create or replace function log_avances_update()  
  returns trigger  
  language 'plpgsql'  
as $$  
BEGIN  
  if new <> old THEN  
    insert into avances_updates_log(usuario, update_time, update_log) values (current_user, now(), new);  
  end if;  
  return new;  
end;  
$$;  
  
create trigger avances_updates  
after update on avances  
for each row  
execute procedure log_avances_update();
```



# Algunas consultas

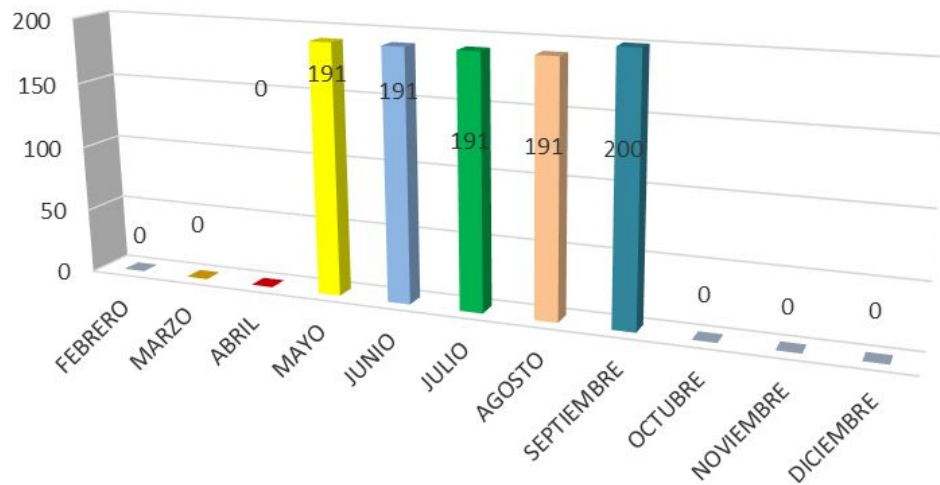
```
SELECT genero, sum(cantidad_personas),  
       CASE  
         WHEN edad > 0  
           AND edad <= 5 THEN '0-5 años'  
         WHEN edad > 5  
           AND edad <= 12 THEN '6-12 años'  
         WHEN edad > 12  
           AND edad <= 17 THEN '12-17 años'  
         WHEN edad > 17  
           AND edad <= 29 THEN '18-29 años'  
         WHEN edad > 29  
           AND edad <= 59 THEN '30-59 años'  
         WHEN edad > 60 THEN 'mayor de 60 años'  
       END rango  
FROM beneficiados  
GROUP BY rango, genero;
```

```
select a.mes, sum(b.cantidad_personas)  
from beneficiados b natural join avances a  
group by mes;
```

```
select distinct a.subprog, p.nombre  
from avances a, provincia p, barrio_vereda b, municipio m  
where a.zona = b.codigo and b.municip = m.codigo and m.provinc = p.codigo
```

# Gráficos

34.2 BOYACÁ AVANZA POR LA VIDA SALUDABLE DE LA PERSONA MAYOR



# Futuro del proyecto