

%% Punto 1

```
puedenCombinarse(Linea, OtraLinea):-  
    recorrido(Linea, Zona, Calle),  
    recorrido(OtraLinea, Zona, Calle),  
    Linea \= OtraLinea.
```

%% Punto 2

```
cruzaGralPaz(Linea):-  
    recorrido(Linea, caba, _),  
    recorrido(Linea, gba(_), _).
```

```
perteneceA(caba, caba).  
perteneceA(gba(_), buenosAires).
```

```
jurisdiccion(Linea, nacional):-  
    cruzaGralPaz(Linea).
```

```
jurisdiccion(Linea, provincial(Provincia)):-  
    recorrido(Linea, Zona, _),  
    perteneceA(Zona, Provincia),  
    not(cruzaGralPaz(Linea)).
```

%%Punto3

```
cuantasLineasPasan(Calle, Zona, Cantidad):-  
    recorrido(_, Zona, Calle),  
    findall(Calle, recorrido(_, Zona, Calle), Calles),  
    length(Calles, Cantidad).
```

```
calleMasTransitada(Calle, Zona):-  
    cuantasLineasPasan(Calle, Zona, Cantidad),  
    forall((recorrido(_, Zona, OtraCalle), Calle \= OtraCalle),  
(cuantasLineasPasan(OtraCalle, Zona, CantidadMenor), Cantidad > CantidadMenor)).
```

%% Punto 4

```
calleDeTrasbordo(CalleTrasbordo, Zona):-  
    recorrido(_, Zona, CalleTrasbordo),  
    forall(recorrido(Linea, Zona, CalleTrasbordo), jurisdiccion(Linea, nacional)),  
    cuantasLineasPasan(CalleTrasbordo, Zona, CantidadLineasCalleTrasbordo),  
    CantidadLineasCalleTrasbordo >= 3.
```

%% Punto 5

```
pasaPorDistintasZonas(Linea):-  
    recorrido(Linea, gba(Zona), _),  
    recorrido(Linea, gba(OtraZona), _),  
    Zona \= OtraZona.
```

```
plus(Linea, 50):-
```

```

    pasaPorDistintasZonas(Linea).
plus(Linea, 0):-
    not(pasaPorDistintasZonas(Linea)).

valorNormal(Linea, 500):-
    jurisdiccion(Linea, nacional).
valorNormal(Linea, 350):-
    jurisdiccion(Linea, provincial(caba)).
valorNormal(Linea, Valor):-
    jurisdiccion(Linea, provincial(buenosAires)),
    findall(Calle, recorrido(Linea, Calle, _), Calles),
    length(Calles, CantidadCalles),
    plus(Linea, Plus),
    Valor is (25*CantidadCalles) + Plus.

beneficiario(pepito, personalCasaParticular(gba(oeste))).
beneficiario(juanita, estudiantil).
beneficiario(marta, jubilado).
beneficiario(marta, personalCasaParticular(caba)).
beneficiario(marta, personalCasaParticular(gba(sur))).

beneficio(estudiantil, _, 50).
beneficio(personalCasaParticular(Zona), Linea, 0):-
    recorrido(Linea, Zona, _).
beneficio(jubilado, Linea, ValorConBeneficio):-
    valorNormal(Linea, ValorNormal),
    ValorConBeneficio is ValorNormal // 2.

posiblesBeneficios(Persona, Linea, ValorConBeneficio):-
    beneficiario(Persona, Beneficio),
    beneficio(Beneficio, Linea, ValorConBeneficio).

costo(Persona, Linea, CostoFinal):-
    beneficiario(Persona, _),
    recorrido(Linea, _, _),
    posiblesBeneficios(Persona, Linea, CostoFinal),
    forall((posiblesBeneficios(Persona, Linea, OtroValorBeneficiado),
    OtroValorBeneficiado \= CostoFinal), CostoFinal < OtroValorBeneficiado).

costo(Persona, Linea, ValorNormal):-
    persona(Persona),
    valorNormal(Linea, ValorNormal),
    not(beneficiario(Persona, _)).

persona(pepito).
persona(juanita).
persona(tito).
persona(marta).

```

%% Punto 5 c)

Si quisiéramos agregar otro beneficio, sea cual sea, sería fácil de implementar gracias al polimorfismo, debido a que sólo deberíamos agregarlo junto con sus condiciones en el predicado beneficio/1, lo cual no nos genera mucha dificultad, por lo tanto, el agregado de estos no nos cambiaría la solución.