

Examen Sustitutorio de Inteligencia Artificial 2023-I

Apellidos y nombres:

Vargas Gonzales Jorge Antony

Ambrosio Quispe Jose Enrique

En el caso del árbol genealógico y el bus que recorre varios paraderos, agregue las siguientes consideraciones

- Una persona nace en un país (debe contener por lo menos 8 países. Por lo menos 4 parejas de países diferentes)
- Una persona habla el idioma del país en el que nace y adicionalmente los idiomas de los progenitores.
- Una persona también habla los idiomas que adicionalmente puede aprender. Considere que todas las personas aprendieron mínimo un idioma
- **Considere que en cada paradero deben subir por lo menos 3 personas**
- **GENERE la REGLA que los que bajan en el ultimo paradero son todos aquellos que subieron y que no bajaron antes del último paradero**

Elaborar un sistema experto para lo siguiente.

1. Listar los nombres de los familiares de ZZ que subieron antes del paradero XX y que no hablan inglés.
2. Listar los nombres de los descendientes de ZZ que bajaron después del paradero XX y que no hablan inglés.
3. Calcular el monto total que pagaron los descendientes de ZZ que bajaron en el último paradero y que hablan inglés (asumiendo que las mujeres pagan 1 sol y los hombres pagan 2 soles)
4. Calcular el porcentaje de descendientes de ZZ que no subieron al bus.
5. Calcular el porcentaje de pasajeros que bajaron en el último paradero. (especto al total de pasajeros)

Entiéndase por familiar el **progenitor, hijo, hermano, tío, sobrino y primo (hombres y mujeres)**

Ejecute el programa 3 veces y presente los resultados con datos diferentes para cada caso

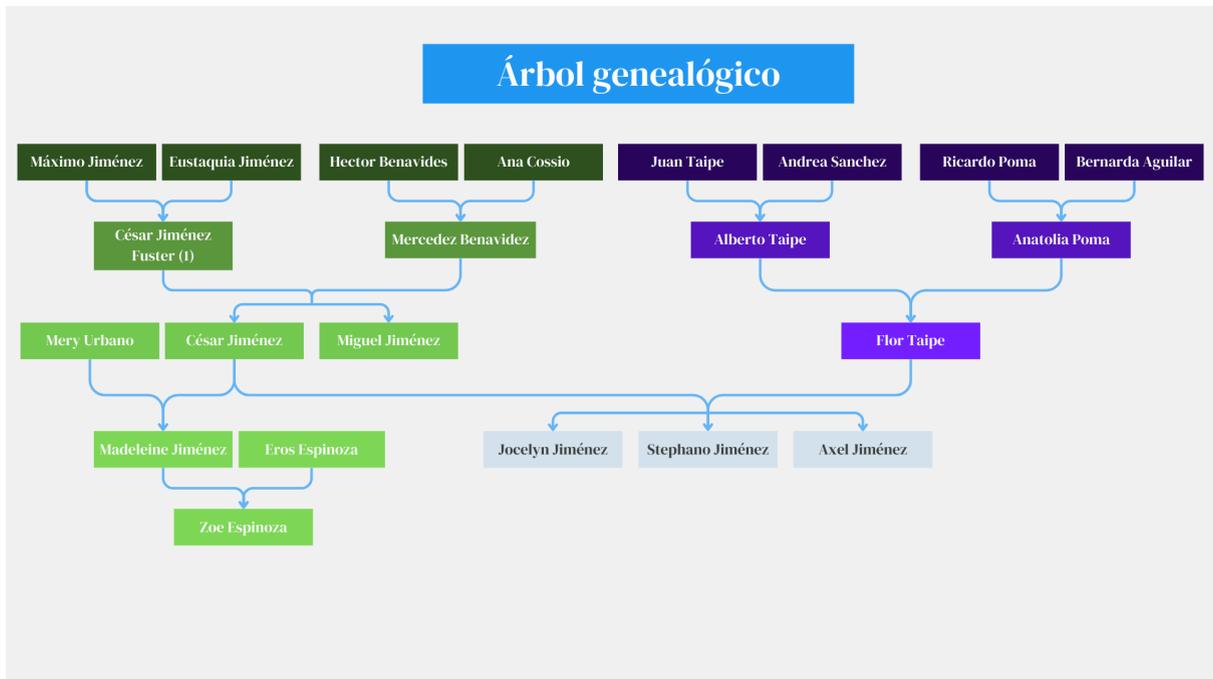
El examen debe contener

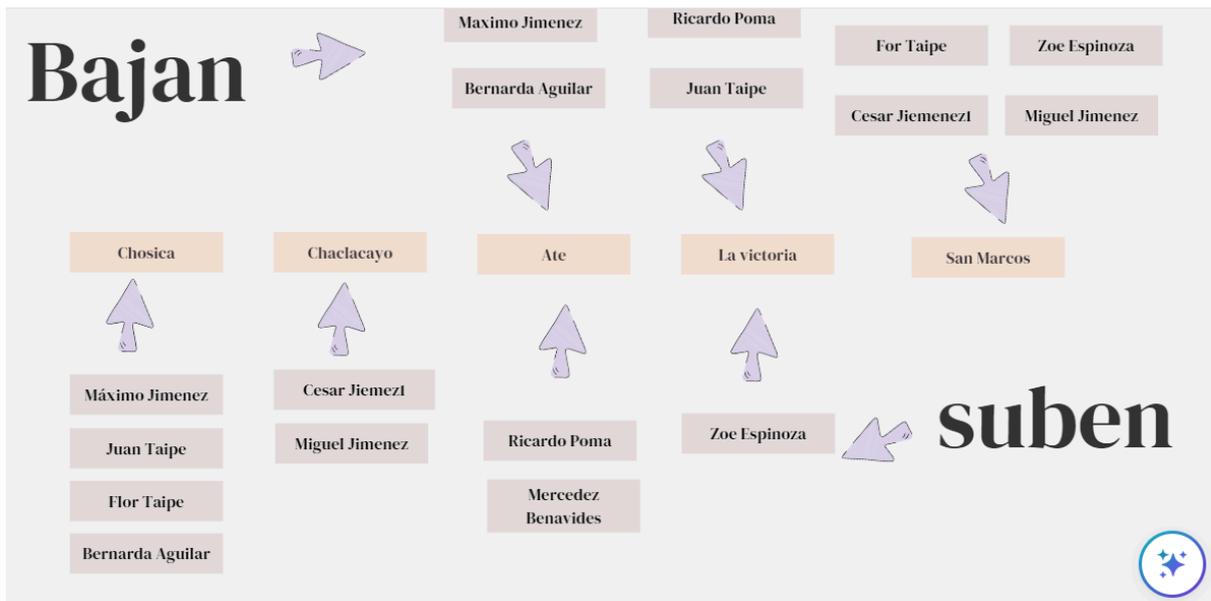
- a. Árbol genealógico que indique en que país nació cada persona
- b. El gráfico Mapa de paraderos indicando quienes suben y bajan en cada paradero
- c. Código fuente
- d. Resultados solicitados

Calificación

	Hasta 2 hrs	2-3 hrs	+3hrs
a. De 2 alumnos	14	10	5
b. Individual	16	12	8

Enviar el examen a mi correo en archivo doc o pdf





CODIGO

```

varon(maximoJimenez).
varon(hectorBenavides).
varon(juanTaipe).
varon(ricardoPoma).
varon(cesarJimenez1).
varon(albertoTaipe).
varon(cesarJimenez).
varon(stephanoJimenez).
varon(axelJimenez).
varon(erosEspinoza).
%-----
varon(miguelJimenez).

mujer(eustaquiaFuster).
mujer(anaCossio).
mujer(andreaSanchez).
mujer(bernardaAguilar).
mujer(mercedesBenavides).
mujer(anatoliaPoma).
mujer(floraTaipe).
mujer(jocelynJimenez).
mujer(madeleineJimenez).
mujer(zoeEspinoza).
mujer(merySanchez).

%---- Paradero -----

paradero(chosica).
paradero(chaclacayo).
paradero(ate).
paradero(lavictoria).
paradero(sanmarcos).

sube(maximoJimenez,chosica).
sube(juanTaipe,chosica).

```

sube(florTaipe,chosica).
sube(bernardaAguilar,chaclacayo).
sube(ricardoPoma,ate).
sube(mercedezBenavides,ate).
sube(cesarJimenez1,chaclacayo).
sube(miguelJimenez,chaclacayo).
sube(zoeEspinoza,lavictoria).

baja(maximoJimenez,ate).
baja(bernardaAguilar,ate).
baja(ricardoPoma,lavictoria).
baja(mercedezBenavides,sanmarcos).
baja(juanTaipe,lavictoria).
baja(florTaipe,sanmarcos).
baja(cesarJimenez1,sanmarcos).
baja(miguelJimenez,sanmarcos).
baja(zoeEspinoza,sanmarcos).

%--relaciones de orden --

antes(chosica,chaclacayo).
antes(chaclacayo,ate).
antes(ate,lavictoria).
antes(lavictoria,sanmarcos).

progenitor(maximoJimenez,cesarJimenez1).
progenitor(eustaquiaFuster,cesarJimenez1).

progenitor(hectorBenavides,mercedezBenavides).
progenitor(anaCossio,mercedezBenavides).

progenitor(juanTaipe,albertoTaipe).
progenitor(andreaSanchez,albertoTaipe).

progenitor(ricardoPoma,anatoliaPoma).
progenitor(bernardaAguilar,anatoliaPoma).

progenitor(cesarJimenez1,cesarJimenez).
progenitor(mercedezBenavides,cesarJimenez).

progenitor(albertoTaipe,florTaipe).
progenitor(anatoliaPoma,florTaipe).

progenitor(cesarJimenez,jocelynJimenez).
progenitor(cesarJimenez,stephanoJimenez).
progenitor(cesarJimenez,axelJimenez).
progenitor(florTaipe,jocelynJimenez).
progenitor(florTaipe,stephanoJimenez).
progenitor(florTaipe,axelJimenez).

%----progenitores para madeleine
progenitor(merySanchez,madeleineJimenez).
progenitor(cesarJimenez,madeleineJimenez).

progenitor(madeleineJimenez,zoeEspinoza).
progenitor(erosEspinoza,zoeEspinoza).

%----cesar hnro
progenitor(cesarJimenez1,miguelJimenez).
progenitor(mercedezBenavides,miguelJimenez).

pareja(maximoJimenez,eustaquiaFuster).
pareja(hectorBenavides,anaCossio).
pareja(juanTaipe,andreaSanchez).
pareja(ricardoPoma,bernardaAguilar).
pareja(cesarJimenez1,mercedezBenavides).
pareja(albertoTaipe,anatoliaPoma).
pareja(cesarJimenez,florTaipe).

pareja(erosEspinoza,madeleineJimenez).

pareja(eustaquiaFuster,maximoJimenez).
pareja(anaCossio,hectorBenavides).
pareja(andreaSanchez,juanTaipe).
pareja(bernardaAguilar,ricardoPoma).
pareja(mercedezBenavides,cesarJimenez1).
pareja(anatoliaPoma,albertoTaipe).
pareja(florTaipe,cesarJimenez).
pareja(madeleineJimenez,erosEspinoza).
%pareja con cesar y carmen

pareja(merySanchez,cesarJimenez).
pareja(cesarJimenez,carmen).

%REGLAS

padre(X,Y):-varon(X),progenitor(X,Y).
madre(X,Y):-mujer(X),progenitor(X,Y).

hermanos(X,Y):-progenitor(Z,X),progenitor(Z,Y), not(X==Y).
hermano(X,Y):-varon(X),hermanos(X,Y).
hermana(X,Y):-mujer(X),hermanos(X,Y).

%-----

esposo(X,Y):-pareja(X,Y),varon(X).
esposa(X,Y):-pareja(X,Y),mujer(X).

%-----

suegro(X,Y):-padre(X,Z),pareja(Y,Z).
suegra(X,Y):-madre(X,Z),pareja(Y,Z).

yerno(X,Y):-suegro(Y,X);suegra(Y,X),varon(X).
nuera(X,Y):-suegro(Y,X);suegra(Y,X),mujer(X).

cuñados(X,Y):-((pareja(X,Z),hermanos(Z,Y));(pareja(Y,Z),hermanos(Z,X))).
cuñado(X,Y):-cuñados(X,Y),varon(X).
cuñada(X,Y):-cuñados(X,Y),mujer(X).

abuelo(X,Y):-progenitor(Z,Y),padre(X,Z).
abuela(X,Y):-progenitor(Z,Y),madre(X,Z).

nieto(X,Y):-progenitor(Y,Z),progenitor(Z,X),varon(X).
nieta(X,Y):-progenitor(Y,Z),progenitor(Z,X),mujer(X).

tio(X,Y):-progenitor(Z,Y),(hermano(X,Z);cuñado(X,Z)).
tia(X,Y):-progenitor(Z,Y),(hermana(X,Z);cuñada(X,Z)).

primo(X,Y):-progenitor(Z,X),progenitor(W,Y),hermanos(Z,W),varon(X).
prima(X,Y):-progenitor(Z,X),progenitor(W,Y),hermanos(Z,W),mujer(X).

hijo(X,Y) :- progenitor(Y,X),varon(X).
hija(X,Y) :- progenitor(Y,X),mujer(X).

bisabuelo(X,Y) :- padre(X,W),progenitor(W,Z),progenitor(Z,Y).
bisabuela(X,Y) :- madre(X,W),progenitor(W,Z),progenitor(Z,Y).

bisnieto(X,Y) :- bisabuela(Y,X),varon(X); bisabuelo(Y,X),varon(X).
bisnieta(X,Y) :- bisabuelo(Y,X),mujer(X); bisabuela(Y,X),mujer(X).

tatarabuelo(X,Y) :- padre(X,W),progenitor(W,Z),progenitor(Z,V),progenitor(V,Y).
tatarabuela(X,Y) :- madre(X,W),progenitor(W,Z),progenitor(Z,V),progenitor(V,Y).

%-----

tataranieto(X,Y) :- tatarabuela(Y,X),varon(X); tatarabuelo(Y,X),varon(X).
tataranieta(X,Y) :- tatarabuelo(Y,X),mujer(X); tatarabuela(Y,X),mujer(X).

%-----

sobrino(X,Y) :- tia(Y,X),varon(X); tio(Y,X),varon(X).
sobrina(X,Y) :- tio(Y,X),mujer(X); tia(Y,X),mujer(X).

mediohermanos(X,Y) :- pareja(B,A),pareja(A,C),progenitor(B,X),progenitor(C,Y),not(B==C),not(X==Y).

primohermanos(X,Y) :- progenitor(Z,X),tio(Z,Y),varon(X).

%-----

tio_politico(X,Y) :- pareja(Y,Z), tio(X,Z); esposo(X,Z),tia(Z,Y).
tia_politica(X,Y) :- pareja(Y,Z), tia(X,Z); esposa(X,Z),tio(Z,Y).

tios_abuelos(X,Y):- abuela(Z,Y), hermanos(X,Z); abuelo(Z,Y),hermanos(X,Z).
primo_segundo(X,Y) :- tios_abuelos(Z,Y),nieta(X,Z).
prima_segunda(X,Y) :- tios_abuelos(Z,Y),nieta(X,Z).
%REGLA 1-----

% Regla para determinar si una persona es familiar de otra

es_familiar(Persona, Familiar) :-
 padre(Persona, Familiar).
es_familiar(Persona, Familiar) :-
 madre(Persona, Familiar).
es_familiar(Persona, Familiar) :-
 hermano(Persona, Familiar).
es_familiar(Persona, Familiar) :-
 hermana(Persona, Familiar).
es_familiar(Persona, Familiar) :-
 hijo(Persona, Familiar).
es_familiar(Persona, Familiar) :-
 hija(Persona, Familiar).
es_familiar(Persona, Familiar) :-
 tio(Persona, Familiar).
es_familiar(Persona, Familiar) :-
 tia(Persona, Familiar).
es_familiar(Persona, Familiar) :-
 sobrino(Persona, Familiar).
es_familiar(Persona, Familiar) :-
 sobrina(Persona, Familiar).
es_familiar(Persona, Familiar) :-
 primohermanos(Persona, Familiar).

%descendientes

descendiente(Persona, Descendiente) :- progenitor(Persona, Descendiente).

descendiente(Persona, Descendiente) :- progenitor(Persona, Hijo),descendiente(Hijo, Descendiente).

%REGLAS

despues(X,Y) :- antes(Y,X).
todos_despues(X, Y) :- despues(X, Y).
todos_despues(X, Y) :- despues(X, Z), todos_despues(Z, Y).
todos_antes(X, Y) :- antes(X, Y).
todos_antes(X, Y) :- antes(X, Z), todos_antes(Z, Y).

nombres_suben(Paradero, Individuos) :-
 findall(Individuo, sube(Individuo, Paradero), Individuos).

comparar_nombres(Lista, Nombre, Resultado) :-
 findall(X, (member(X, Lista), es_familiar(Nombre, X)), Resultado).

comparar_nombres_descendientes(Lista, Nombre, Resultado) :-
 findall(X, (member(X, Lista), descendiente(Nombre, X)), Resultado).

nacionalidad(maximoJimenez, [peru]).
nacionalidad(eustaquiaFuster, [peru]).
nacionalidad(hectorBenavides, [peru]).
nacionalidad(anaCossio, [peru]).
nacionalidad(juanTaipe, [peru]).
nacionalidad(andreaSanchez, [peru]).
nacionalidad(ricardoPoma, [peru]).
nacionalidad(bernardaAguilar, [peru]).
nacionalidad(erosEspinoza, [peru]).
nacionalidad(cesarJimenez1, [peru]).
nacionalidad(albertoTaipe, [canada, japon]).
nacionalidad(cesarJimenez, [peru, japon]).
nacionalidad(stephanoJimenez, [japon, peru, canada, brasil, francia]).
nacionalidad(axelJimenez, [japon, peru, canada, brasil, francia]).

```
nacionalidad(miguelJimenez, [mexico, japon]).
nacionalidad(mercedezBenavides, [mexico, japon]).
nacionalidad(anatoliaPoma, [francia]).
nacionalidad(florTaipe, [canada, brasil]).
nacionalidad(jocelynJimenez, [japon, mexico, canada, brasil, francia]).
nacionalidad(madeleineJimenez, [mexico, japon]).
nacionalidad(zoeEspinoza, [estadosunidos, japon, mexico]).
nacionalidad(merySanchez, [japon]).
```

```
idioma(peru,espanol).
idioma(mexico,espanol).
idioma(chile,espanol).
idioma(japon,japones).
idioma(canada,ingles).
idioma(francia,frances).
idioma(brasil,portugues).
idioma(estadosunidos,ingles).
hereda_idioma(Descendiente, Idioma) :- progenitor(Progenitor, Descendiente), idioma(Pais, Idioma),
nacionalidad(Progenitor, Pais).
```

```
idioma_nacionalidad(Persona, Idioma) :-
    nacionalidad(Persona, Nacionalidad),
    idioma(Nacionalidad, Idioma).
```

```
idiomas_persona(Persona, Idiomas) :-
    nacionalidad(Persona, Nacionalidad),
    findall(Idioma, (member(Nacionalidad, Nacionalidades), idioma(Nacionalidad, Idioma) ; hereda_idioma(Persona, Idioma)), TodosLosIdiomas),
    list_to_set(TodosLosIdiomas, Idiomas).
```

%PROBLEM 1

```
familiar_subieron_antes_de(Paradero, Individuos, Persona) :-
    findall(Individuo, (sube(Individuo, Subida), todos_despues(Paradero, Subida),
not(idiomas_persona(Individuo,ingles))), ListaA),
    nombres_suben(Paradero, ListaB),
    append(ListaB, ListaA, ListaCombinada),
    comparar_nombres(ListaCombinada, Persona, Resultado),
    list_to_set(Resultado, Individuos).
```

%Falta la union con el idioma

%PROBLEM 2

```
descendiente_bajaron_despues_de(Paradero, Individuos, Persona) :-
    findall(Individuo, (baja(Individuo, Subida), todos_antes(Paradero, Subida)), ListaA),
    nombres_suben(Paradero, ListaB),
    append(ListaB, ListaA, ListaCombinada),
    comparar_nombres_descendientes(ListaCombinada, Persona, Resultado),
    list_to_set(Resultado, Individuos).
```

%Falta la union con el idioma

%PROBLEM 4

```
subieron_todos(Individuos) :-
    findall(Individuo, (sube(Individuo, _)), ListaA),
    list_to_set(Resultado, Individuos).
```

```
subieron_descendientes( Individuos, Persona) :-
    findall(Individuo, (sube(Individuo, _)), ListaA),
    comparar_nombres_descendientes(ListaA, Persona, Resultado),
    list_to_set(Resultado, Individuos).
```

```
porcentaje_de_descendientes_que_no_subieron(Persona,Individuos) :-
    findall(Individuo, (sube(Individuo, _)), ListaA),
    comparar_nombres_descendientes(ListaA, Persona, Resultado),
    length(Resultado, Cdesce),
```

```
findall(Individuo, (sube(Individuo, _)), Resultado2),  
length(Resultado2, Ctot),
```

```
Individuos is (1 - (Cdesce / Ctot)) * 100.
```

1. Listar los nombres de los familiares de ZZ que subieron antes del paradero XX y que no hablan inglés.

```
familiar_subieron_antes_de(ate, Individuos, miguelJimenez).  
Individuos = [mercedezBenavides, cesarJimenez1]  
?- familiar_subieron_antes_de(ate, Individuos, miguelJimenez).
```

```
familiar_subieron_antes_de(ate, Individuos, zoeEspinoza).  
Individuos = []  
?- familiar_subieron_antes_de(ate, Individuos, zoeEspinoza).
```

```
familiar_subieron_antes_de(lavictoria, Individuos, erosEspinoza).  
Individuos = [zoeEspinoza]  
?- familiar_subieron_antes_de(lavictoria, Individuos, erosEspinoza).
```

2. Listar los nombres de los descendientes de ZZ que bajaron después del paradero XX y que no hablan inglés.

```
descendiente_bajaron_despues_de(ate, Individuos, erosEspinoza).  
Individuos = [zoeEspinoza]  
?- descendiente_bajaron_despues_de(ate, Individuos, erosEspinoza).
```

```
descendiente_bajaron_despues_de(ate, Individuos, zoeEspinoza).  
Individuos = []  
?- descendiente_bajaron_despues_de(ate, Individuos, zoeEspinoza).
```

```
descendiente_bajaron_despues_de(ate, Individuos, cesarJimenez1).
Individuos = [miguelJimenez, zoeEspinoza]
?- descendiente_bajaron_despues_de(ate, Individuos, cesarJimenez1).
```

3. Calcular el monto total que pagaron los descendientes de ZZ que bajaron en el último paradero y que hablan inglés (asumiendo que las mujeres pagan 1 sol y los hombres pagan 2 soles)

4. Calcular el porcentaje de descendientes de ZZ que no subieron al bus.

Individuos == Porcentaje

```
porcentaje_de_descendientes_que_no_subieron(cesarJimenez1,Individuos).
Individuos = 77.77777777777779
?- porcentaje_de_descendientes_que_no_subieron(cesarJimenez1,Individuos).
```

```
porcentaje_de_descendientes_que_no_subieron(zoeEspinoza,Individuos).
Individuos = 100
?- porcentaje_de_descendientes_que_no_subieron(zoeEspinoza,Individuos).
```

```
porcentaje_de_descendientes_que_no_subieron(florTaipe,Individuos).
Individuos = 100
?- porcentaje_de_descendientes_que_no_subieron(florTaipe,Individuos).
```

5. Calcular el porcentaje de pasajeros que bajaron en el último paradero. (especto al total de pasajeros)