

PEP2 Paradigmas de Programación (2020-1)

Junio

NOMBRE: Israel Villagra **RUT:** 16.265.519-1 **Profesor:** Roberto González

Instrucciones

- Duración: 90 minutos (para desarrollar sus respuestas)
- Escriba sus respuestas en hojas separadas
- Puede escribir sus respuestas directamente en un documento electrónico o bien puede hacerlo en papel. Escoja el que más le acomode.
- Identifique cada hoja de respuestas con su nombre y rut, lo que sirve de complemento a la declaración que ha aceptado en Campus Virtual.
- Identifique cada hoja de respuesta con su rut y el profesor de su sección.
- Al terminar la evaluación,
 - Si desarrolló sus respuestas con lápiz y papel, saqué una fotografía o escanee cada una de sus hojas de respuestas. Asegúrese que las imágenes de sus capturas son legibles (abra el archivo y verifique). Luego, agrupe todas las imágenes en un PDF (en Windows basta con seleccionar las imágenes desde el explorador de archivos -> clic derecho -> Imprimir -> Escoger Impresora PDF de Microsoft u otra que disponga) o bien en un archivo comprimido .zip o .rar.
 - Si desarrolló sus respuestas directamente en formato digital (ej: documento Word, Google Docs, LaTeX, SWI Prolog, etc.) guardar el documento como PDF. Se recomienda para evitar problemas en la visualización de estilos.
- Finalmente, subir el archivo al espacio creado para tales efectos en Campus Virtual. Una vez subido el archivo, descárguelo y asegúrese que el archivo se puede abrir (para evitar situaciones de archivos corruptos).
- La fecha y hora de entrega de la prueba es el día 24 de junio de 2020 hasta las 23:59 a través de Campus Virtual en pestaña Evaluaciones de Cátedra.
- Excepción: En la eventualidad que no pueda subir su trabajo a campus virtual, se habilitará una carpeta compartida entre las 23:50 y 23:59. El link a la carpeta se publicará en campus virtual y en este mismo enunciado. Como último recurso, envíe sus respuestas al correo electrónico de su profesor.
- Recomendamos leer atentamente cada pregunta de forma completa, incluyendo las "notas" dispuestas al final de éstas. Procure pensar su solución, diseñar y luego proceder a implementarla.
- Las consultas se pueden hacer vía correo electrónico a roberto.gonzalez.i@usach.cl las que serán respondidas durante el transcurso de la tarde (entre 14:00 y 18:30).. Procure poner en encabezado (PREGUNTA PEP2 Paradigmas)
- También se habilitará un espacio de videoconferencia breve de 20 minutos entre las 21:00 y 21:20 hrs.
- Se reitera la importancia de su integridad académica (código de honor al que usted se ha suscrito) al momento de realizar estas evaluaciones no presenciales. Copias, solicitar ayuda a terceros o buscar información en fuentes bibliográficas que sean identificadas serán sancionadas en base al reglamento institucional. Sus respuestas deben basarse únicamente en base a su estudio de la materia.

Pregunta 1 (30 pts):

En una empresa financiera comúnmente requieren disponer de fuentes de información comercial de los potenciales clientes, la más conocida de todas estas fuentes es Ticom Aquifax (nombre de fantasía), sin embargo existen muchas otras de uso interno entre las empresas financieras.

A continuación se expresan los datos de estas fuentes para 5 personas distintas, considere que hay casos en que no se tiene información de algunas personas en todas las fuentes. También, hay fuentes de información que tienen datos que otras no tienen, por ejemplo Ticom tiene la dirección de la persona, pero Rimacofi no, sin embargo Rimacofi dispone del teléfono.

Fuente 1: SII

Rut	Nombre	Fecha inicio actividades	¿Termino giro?	Tipo persona
12.344.100-8	Cisarro Covid	No tiene	No	Natural
16.500.122-5	Nicole Rojas	20/11/2010	No	Natural
76.540.100-6	Guantes LTDA	30/10/2012	Si	Jurídica
64.990.657-1	Empresa1 S.A	12/07/2015	No	Jurídica
5.680.123-4	Ana Durán	12/2/2020	No	Natural

Fuente 2: Ticom Aquifax

Rut	Nombre	Dirección	Comuna	Morosidades	Infracciones previsionales	Protestos	Monto créditos
16.500.122-5	Nicole Rojas	Calle Sur #454	Recoleta	2.500.000	500.590	0	60.100.000
64.990.657-1	Empresa1 S.A	Poniente 1 #121	Santiago	52.000.000	4.960.010	399.890	101.890.410
5.680.123-4	Ana Durán	Rosales #31123	Maipú	4.500	0	0	300.000

Fuente 3: Gesintel

Rut	N° demandas	N° detenciones	¿Buscado por PDI?
16.500.122-5	5	1	No
12.344.100-8	1	0	Si

Fuente 4: Rimacofi

Rut	Morosidades	Infracciones previsionales	Protestos	Teléfono
16.500.122-5	2.500.000	500.590	0	600421777
64.990.657-1	6.400.000	4.960.010	0	944581110

- Expresa esta información en una base de conocimientos de prolog (5 pts).
- Basándose en los hechos expresados en la pregunta anterior, realice un predicado en prolog que permita consultar las direcciones de una persona según su rut (5 pts).

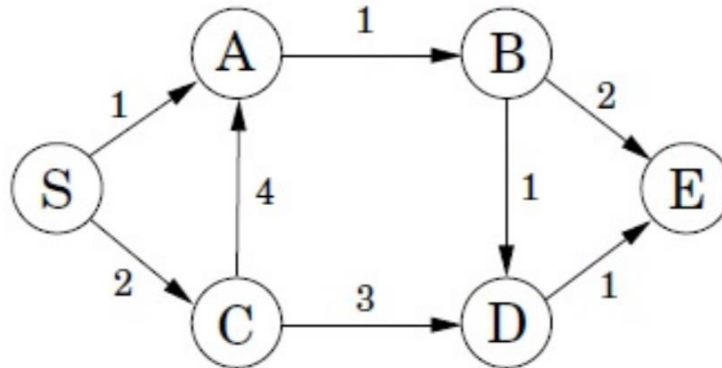
- c) Realice un predicado en prolog que permita obtener el total de deudas vencidas de una persona natural. La deuda total vencida corresponde a morosidades + infracciones previsionales + protestos. En caso que la persona tenga deudas declaradas en varias fuentes de información debe entregar los datos de la mayor de estas (Los datos de la fuente en que tenga mayor monto de deuda) **(8 pts)**.
- d) Antes de realizar negocios con estas personas, las empresas del rubro financiero disponen de un equipo de analistas de riesgos que realizan una evaluación previa, la tendencia en el rubro es intentar automatizar la tarea de la evaluación de riesgos a la mayor cantidad de clientes y que el equipo de analistas de riesgos sólo evalúen ciertos casos. Para esto, en una empresa financiera X se han establecido las siguientes políticas/criterios a ser evaluadas de forma automática. Todo aquello que no sea considerado en estas políticas deben ser evaluadas por algún analista de riesgos.
- Cuando una persona tiene protestos, inmediatamente debe ser rechazado
 - Si tiene una fecha de inicio de actividades menor a 6 meses atrás siendo persona jurídica, inmediatamente debe ser rechazado.
 - Si tiene detenciones o es buscado por la PDI, inmediatamente debe ser rechazado.
 - Si ha terminado su giro, inmediatamente debe ser rechazado.
 - Si sus deudas totales vencidas superan el 25% de sus créditos solicitados, inmediatamente debe ser rechazado.

Realice un predicado en prolog que permita saber si un deudor debe ser rechazado de forma automática sin siquiera pasar por la evaluación de un analista de riesgos. Su predicado debiese funcionar con cualquier otra persona que no necesariamente esté definida en la base de conocimientos indicada anteriormente, por lo que debe procurar que la implementación de su predicado en base a cláusulas debe ser genérico **(8 pts)**.

- e) Documente adecuadamente toda su solución procurando destacar dominios, predicados, metas (primarias y secundarias) y cláusulas. **(4 pts)**.

Pregunta 2 (30 pts):

El siguiente esquema representa los estados (A, B, C, D, E y S) de un sistema que opera maquinaria industrial, además de incluir los costos para pasar de un estado a otro (por ejemplo, para ir del estado S a C el costo es de 2).



En base al diagrama recién indicado, implemente en pseudo-Prolog lo siguiente:

1. Expresar a través de una serie de hechos el esquema de la figura anterior. Procure representar las transiciones entre los estados indicando su costo y dirección. Escoja un nombre de predicado acorde al contexto del problema. **(8 pts)**
2. Implementar las cláusulas para el predicado `verificar(origen,destino,costo)` que permitan consultar si se puede ir desde un estado a otro especificando un costo exacto determinado **(15 pts)**

Por

ejemplo:

<code>verificar(s, d, 6)</code>	<code>=></code>	<code>false</code>
<code>verificar(s, d, 5)</code>	<code>=></code>	<code>true</code>
<code>verificar(s, e, 4)</code>	<code>=></code>	<code>true</code>

3. Basado en la lista de hechos de su base de conocimientos del ítem 1 y el predicado del ítem 2, realice la traza de la consulta `verificar(s, d, 3)`. **(3 pts)**.
 - a. Con traza no es necesariamente lo que hacer SWI-Prolog con trazes. Nos interesa ver su desarrollo de lo que hace Prolog en el proceso de búsqueda de soluciones
4. Documente adecuadamente toda su solución procurando destacar dominios, predicados, metas (primarias y secundarias) y cláusulas. **(4 pts)**.

Pregunta:

%Israel Villagra

%16265519-1

%Dominios

% sii(Rut,Nombre,Fecha_inicio_actividades,Termino_giro, Tipo_persona) symbol.

% aquifax(RUT, NOMBRE,DIRECCION ,COMUNA, Morosidades, Infracciones_previsionales, Protestos, MontoCreditos) Symbol.

% gesintel(Rut,Ndemandas,Ndetenciones,BuscadoPDI) symbol.

%Predicados

% direccion(RUT,DIRECCION) aridad = 2

% deudadVencidadPersonaNatural(RUT,S) aridad = 2

% clienteSinProtestos(RUT,PROTESTOS) aridad = 2

% noBuscadoPDI(RUT,BUSCADO) aridad = 2

% giroNoDetenido(RUT,DETENIDO) aridad = 2

% deudadMenoresA(RUT,X) aridad = 2

% evaluaCliente(RUT,Y) aridad = 2

%Metas

% Primarias

% evaluaCliente(RUT,Y)

% deudadVencidadPersonaNatural(RUT,S)

% evaluaCliente

% Secundarias

% direccion(RUT,DIRECCION) aridad = 2

% clienteSinProtestos(RUT,PROTESTOS) aridad = 2

% noBuscadoPDI(RUT,BUSCADO) aridad = 2

% giroNoDetenido(RUT,DETENIDO) aridad = 2

% deudadMenoresA(RUT,X) aridad = 2

%Cláusulas

% hechos

% Rut	Nombre	Fecha inicio actividades	¿Termino giro?	Tipo persona
-------	--------	--------------------------	----------------	--------------

sii('12.344.100-8', 'Cisarro Covid', 'No tiene'			, 'No'	, 'Natural').
---	--	--	--------	---------------

sii('16.500.122-5', 'Nicole Rojas', '20/11/2010'			, 'No'	, 'Natural').
--	--	--	--------	---------------

sii('76.540.100-6', 'Guantes LTDA', '30/10/2012'			, 'Si'	, 'Jurídica').
--	--	--	--------	----------------

sii('64.990.657-1', 'Empresa1 S.A', '12/07/2015'			, 'No'	, 'Jurídica').
--	--	--	--------	----------------

sii('5.680.123-4', 'Ana Durán', '12/2/2020'			, 'No'	, 'Natural').
---	--	--	--------	---------------

% Rut	Nombre	Morosidades	Infracciones	previsionales
-------	--------	-------------	--------------	---------------

	Protestos	Monto créditos		
--	-----------	----------------	--	--

aquifax('16.500.122-5', 'Nicole Rojas', 'Calle Sur #454.'		, 'Recoleta', 2500000	, 500590	, 0
		, 60100000).		

```

aquifax('64.990.657-1', 'Empresa1 S.A', 'Poniente 1 #121.', 'Santiago', 52000000, 4960010,
399890, 101890410).
aquifax('5.680.123-4', 'Ana Durán', 'Rosales #31123.', 'Maipú', 4500, 0, 0, 300000).

```

```

% Rut N° demandas N° detenciones ¿Buscado por PDI?
gesintel('16.500.122-5', '5', '1', 'No').
gesintel('12.344.100-8', '1', '0', 'Si').

```

```

% Rut Morosidades Infracciones previsionales Protestos Teléfono
sinacofi('16.500.122-5', 2500000, 500590, 0, '600421777').
sinacofi('64.990.657-1', 6400000, 4960010, 0, '944581110').%11.360.010

```

```
%Reglas
```

```
%Dirección de la persona
```

```

direccion(RUT,DIRECCION):- aquifax(RUT, _,DI,COMUNA, _, _,_,_), atom_concat(DI,
COMUNA,DIRECCION);DIRECCION = 'No Encontrado, Compruebe RUT'.

```

```
%Deuda Persona Natural.
```

```

deudadVencidadPersonaNatural(RUT,S):- sii(RUT,_,_,_, 'Natural'),
sinacofi(RUT,MS,IPS,PS,_),
aquifax(RUT,_,_,_,ME,IPE,PSE,_),
(MS+IPS+PS)> (ME+IPE+PSE),
S is (MS+IPS+PS);
aquifax(RUT,_,_,_,ME,IPE,PSE,_),S is (ME+IPE+PSE).

```

```
%*****Región Clienés Sin Protestos*****
```

```
%Busca clientes con Protestos
```

```

clienteSinProtestos(RUT,PROTESTOS):- sinacofi(RUT,_,_,PS,_), aquifax(RUT,_,_,_,_,PSE,_), (PS
= 0, PSE = 0 -> PROTESTOS = true ; PROTESTOS = false);PROTESTOS = true.

```

```
%clienteSinProtestos('64.990.657-1',X).
```

```
%clienteSinProtestos('5.680.123-4',X).
```

```
%*****Región No Buscado por PDI*****
```

```

noBuscadoPDI(RUT,BUSCADO) :-gesintel(RUT, _ ,_,PDI), (PDI =@= 'No' -> BUSCADO = true ;
BUSCADO=false).

```

```
%noBuscadoPDI('16.500.122-5',X).
```

```
%noBuscadoPDI('12.344.100-8',X).
```

```
%*****Región Giro No Detenido*****
```

```

giroNoDetenido(RUT,DETENIDO) :-sii(RUT,_,_, GIRODETENIDO, _), (GIRODETENIDO =@= 'No' ->
DETENIDO = true ; DETENIDO=false).

```

```
%giroDetenido('12.344.100-8', X).
```

```
%*****Región Persona Juridica*****
```

personaJuridica(RUT,X) :- sii(RUT,_,II,_, 'Jurídica'), X is II.

%*****Región Giro No Detenido*****

```
deudadMenoresA(RUT,X):- sinacofi(RUT,DEUDAS,_,_,_),
aquifax(RUT,_,_,_,DEUDAE,_,_,_),
aquifax(RUT,_,_,_,_,_,_,CREDITO),
    ( (
        (
            (DEUDAS+DEUDAE)*100
        ) /CREDITO
    ) > 25 -> X = false ; X = true).
```

%*****Región Evalua Cliente*****

%Evalua posible cliente

```
evaluaCliente(RUT,Y):- clienteSinProtestos(RUT,PROTESTOSCLIENTE),
noBuscadoPDI(RUT,NO_PDI),
giroNoDetenido(RUT, GIRODETENIDO),
deudadMenoresA(RUT, DEUDACREDITO),
(
    (PROTESTOSCLIENTE = true
    , GIRODETENIDO = true
    , NO_PDI = true
    , DEUDACREDITO = true) -> Y = true ; Y = false
); Y =false.
%evaluaCliente('16.500.122-5',X).
%evaluaCliente('64.990.657-1',X).
```

Respuestas

Pregunta 2:

%Israel Villagra

%16265519-1

%Dominios

% estados(s,a,derecha,1). symbol.

%Predicados

% verificar(INICIO,FINAL,VALOR)

aridad = 3

%Metas

% Primarias

% verificar(INICIO,FINAL,VALOR)

%Cláusulas

% hechos

estados(s,a,derecha,1).

estados(s,c,derecha,2).

estados(c,a,arriba,4).

estados(c,d,derecha,3).

estados(a,b, derecha,1).

estados(b,d, abajo, 1).

estados(b,e, derecha, 2).

estados(d,e, derecha,1).

%Reglas

% Se verifica si el origen con el destino final contiene el valor que se ingresa como parametro

% el INICIO se considera como el ORIGEN

% FINAL se referencia al destino

% El VALOR referencia al COSTO

% Este es una de muchas las soluciones.

verificar(INICIO,FINAL,VALOR):- estados(INICIO,FINAL,_,VALOR1)
, VALOR is VALOR1.

verificar(INICIO,FINAL,VALOR):- estados(INICIO,INICIO1,_,VALOR1)
, estados(INICIO1,FINAL,_,VALOR2)
, VALOR is VALOR1+VALOR2.

verificar(INICIO,FINAL,VALOR):- estados(INICIO,INICIO1,_,VALOR1)
, estados(INICIO1,INICIO2,_,VALOR2)
, estados(INICIO2,FINAL,_,VALOR3)
, VALOR is VALOR1+VALOR2+VALOR3.

verificar(INICIO,FINAL,VALOR):- estados(INICIO,INICIO1,_,VALOR1)
, estados(INICIO1,INICIO2,_,VALOR2)


```
, estados(INICIO2,INICIO3,_,VALOR3)
, estados(INICIO3,FINAL,_,VALOR4)
, VALOR is VALOR1+VALOR2+VALOR3+VALOR4.
```

```
verificar(INICIO,FINAL,VALOR):- estados(INICIO,INICIO1,_,VALOR1)
, estados(INICIO1,INICIO2,_,VALOR2)
, estados(INICIO2,INICIO3,_,VALOR3)
, estados(INICIO3,INICIO4,_,VALOR4)
, estados(INICIO4,FINAL,_,VALOR5)
, VALOR is VALOR1+VALOR2+VALOR3+VALOR4+VALOR5.
```

```
%Israel Villagra
%16265519-1
```

```
[trace] ?- verificar(s, d, 3).
Call: (10) verificar(s, d, 3) ? creep
Call: (11) estados(s, d, _8226, _8228) ? creep
Fail: (11) estados(s, d, _8270, _8272) ? creep
Redo: (10) verificar(s, d, 3) ? creep
Call: (11) estados(s, _8356, _8358, _8360) ? creep
Exit: (11) estados(s, a, derecha, 1) ? creep
Call: (11) estados(a, d, _8446, _8448) ? creep
Fail: (11) estados(a, d, _8490, _8492) ? creep
Redo: (11) estados(s, _8532, _8534, _8536) ? creep
Exit: (11) estados(s, c, derecha, 2) ? creep
Call: (11) estados(c, d, _8622, _8624) ? creep
Exit: (11) estados(c, d, derecha, 3) ? creep
Call: (11) 3 is 2+3 ? creep
Fail: (11) 3 is 2+3 ? creep
Redo: (10) verificar(s, d, 3) ? creep
Call: (11) estados(s, _8846, _8848, _8850) ? creep
Exit: (11) estados(s, a, derecha, 1) ? creep
Call: (11) estados(a, _8934, _8936, _8938) ? creep
Exit: (11) estados(a, b, derecha, 1) ? creep
Call: (11) estados(b, d, _9024, _9026) ? creep
Exit: (11) estados(b, d, abajo, 1) ? creep
Call: (11) 3 is 1+1+1 ? creep
Exit: (11) 3 is 1+1+1 ? creep
Exit: (10) verificar(s, d, 3) ? creep
true ;
Redo: (11) estados(b, d, _9876, _9878) ? creep
```

Fail: (11) estados(b, d, _9920, _9922) ? creep
Redo: (11) estados(s, _9962, _9964, _9966) ? creep
Exit: (11) estados(s, c, derecha, 2) ? creep
Call: (11) estados(c, _10050, _10052, _10054) ? creep
Exit: (11) estados(c, a, arriba, 4) ? creep
Call: (11) estados(a, d, _10140, _10142) ? creep
Fail: (11) estados(a, d, _10184, _10186) ? creep
Redo: (11) estados(c, _10226, _10228, _10230) ? creep
Exit: (11) estados(c, d, derecha, 3) ? creep
Call: (11) estados(d, d, _10316, _10318) ? creep
Fail: (11) estados(d, d, _10360, _10362) ? creep
Redo: (10) verificar(s, d, 3) ? creep
Call: (11) estados(s, _10446, _10448, _10450) ? creep
Exit: (11) estados(s, a, derecha, 1) ? creep
Call: (11) estados(a, _10534, _10536, _10538) ? creep
Exit: (11) estados(a, b, derecha, 1) ? creep
Call: (11) estados(b, _10622, _10624, _10626) ? creep
Exit: (11) estados(b, d, abajo, 1) ? creep
Call: (11) estados(d, d, _10712, _10714) ? creep
Fail: (11) estados(d, d, _10756, _10758) ? creep
Redo: (11) estados(b, _10798, _10800, _10802) ? creep
Exit: (11) estados(b, e, derecha, 2) ? creep
Call: (11) estados(e, d, _10888, _10890) ? creep
Fail: (11) estados(e, d, _10932, _10934) ? creep
Redo: (11) estados(s, _10974, _10976, _10978) ? creep
Exit: (11) estados(s, c, derecha, 2) ? creep
Call: (11) estados(c, _11062, _11064, _11066) ? creep
Exit: (11) estados(c, a, arriba, 4) ? creep
Call: (11) estados(a, _11150, _11152, _11154) ? creep
Exit: (11) estados(a, b, derecha, 1) ? creep
Call: (11) estados(b, d, _11240, _11242) ? creep
Exit: (11) estados(b, d, abajo, 1) ? creep
Call: (11) 3 is 2+4+1+1 ? creep
Fail: (11) 3 is 2+4+1+1 ? creep
Redo: (11) estados(b, d, _11434, _11436) ? creep
Fail: (11) estados(b, d, _11478, _11480) ? creep
Redo: (11) estados(c, _11520, _11522, _11524) ? creep
Exit: (11) estados(c, d, derecha, 3) ? creep
Call: (11) estados(d, _11608, _11610, _11612) ? creep
Exit: (11) estados(d, e, derecha, 1) ? creep
Call: (11) estados(e, d, _11698, _11700) ? creep
Fail: (11) estados(e, d, _11742, _11744) ? creep
Redo: (10) verificar(s, d, 3) ? creep
Call: (11) estados(s, _11828, _11830, _11832) ? creep
Exit: (11) estados(s, a, derecha, 1) ? creep
Call: (11) estados(a, _11916, _11918, _11920) ? creep

Exit: (11) estados(a, b, derecha, 1) ? creep
Call: (11) estados(b, _12004, _12006, _12008) ? creep
Exit: (11) estados(b, d, abajo, 1) ? creep
Call: (11) estados(d, _12092, _12094, _12096) ? creep
Exit: (11) estados(d, e, derecha, 1) ? creep
Call: (11) estados(e, d, _12182, _12184) ? creep
Fail: (11) estados(e, d, _12226, _12228) ? creep
Redo: (11) estados(b, _12268, _12270, _12272) ? creep
Exit: (11) estados(b, e, derecha, 2) ? creep
Call: (11) estados(e, _12356, _12358, _12360) ? creep
Fail: (11) estados(e, _12400, _12402, _12404) ? creep
Redo: (11) estados(s, _12444, _12446, _12448) ? creep
Exit: (11) estados(s, c, derecha, 2) ? creep
Call: (11) estados(c, _12532, _12534, _12536) ? creep
Exit: (11) estados(c, a, arriba, 4) ? creep
Call: (11) estados(a, _12620, _12622, _12624) ? creep
Exit: (11) estados(a, b, derecha, 1) ? creep
Call: (11) estados(b, _12708, _12710, _12712) ? creep
Exit: (11) estados(b, d, abajo, 1) ? creep
Call: (11) estados(d, d, _12798, _12800) ? creep
Fail: (11) estados(d, d, _12842, _12844) ? creep
Redo: (11) estados(b, _12884, _12886, _12888) ? creep
Exit: (11) estados(b, e, derecha, 2) ? creep
Call: (11) estados(e, d, _12974, _12976) ? creep
Fail: (11) estados(e, d, _13018, _13020) ? creep
Redo: (11) estados(c, _13060, _13062, _13064) ? creep
Exit: (11) estados(c, d, derecha, 3) ? creep
Call: (11) estados(d, _13148, _13150, _13152) ? creep
Exit: (11) estados(d, e, derecha, 1) ? creep
Call: (11) estados(e, _13236, _13238, _13240) ? creep
Fail: (11) estados(e, _13280, _13282, _13284) ? creep
Fail: (10) verificar(s, d, 3) ? creep

false.

[trace] ?-