**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA**



**Ingeniería en Software y Tecnologías Emergentes**

Paradigmas de la Programación

Taller U5-1: Ejemplos de prolog

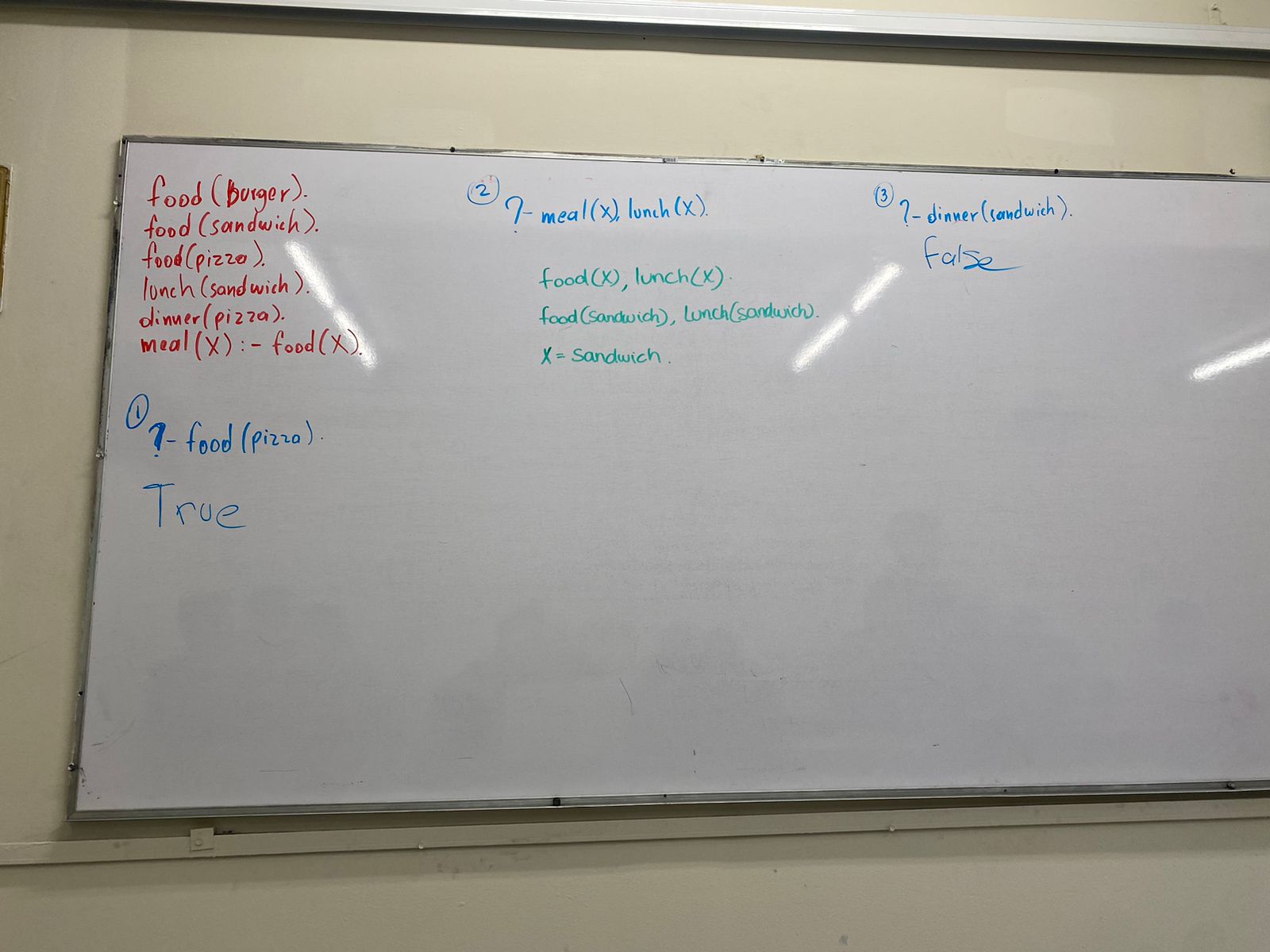
ALUMNOS:

Eliel Alfonso Ontiveros Ojeda

GRUPO: 942

PROFESOR: Carlos Gallegos

30 de mayo del 2024

**Reglas:**

food(burguer).

food(sandwich).

food(pizza).

lunch(sandwich).

dinner(pizza).

meal(X) :- food(X).

**Consulta:**

?- food(pizza).

**Explicación:**

* food(burguer).: Declara que una hamburguesa es comida.
* food(sandwich).: Declara que un sándwich es comida.
* food(pizza).: Declara que una pizza es comida.
* lunch(sandwich).: Declara que el sándwich es un almuerzo.
* dinner(pizza).: Declara que la pizza es una cena.

**Definición de la regla:**

* meal(X) :- food(X).: Declara que X es una comida (meal) si X es comida (food).

**Consulta y resultado:**

* ?- food(pizza).: La consulta pregunta si la pizza es comida.
* El resultado es true porque food(pizza) está definido como un hecho.

**Consulta:**

?- meal(X), lunch(X).

**Evaluación de la regla:**

* La consulta meal(X) usa la regla meal(X) :- food(X).
* Entonces, meal(X) se convierte en food(X).

**Búsqueda de X:**

Se busca un X tal que food(X) y lunch(X) sean verdaderos.

**Evaluación de los hechos:**

* food(sandwich) es verdadero.
* lunch(sandwich) es verdadero.
* Por lo tanto, X = sandwich cumple ambas condiciones.

**Consulta:**

?- dinner(sandwich).

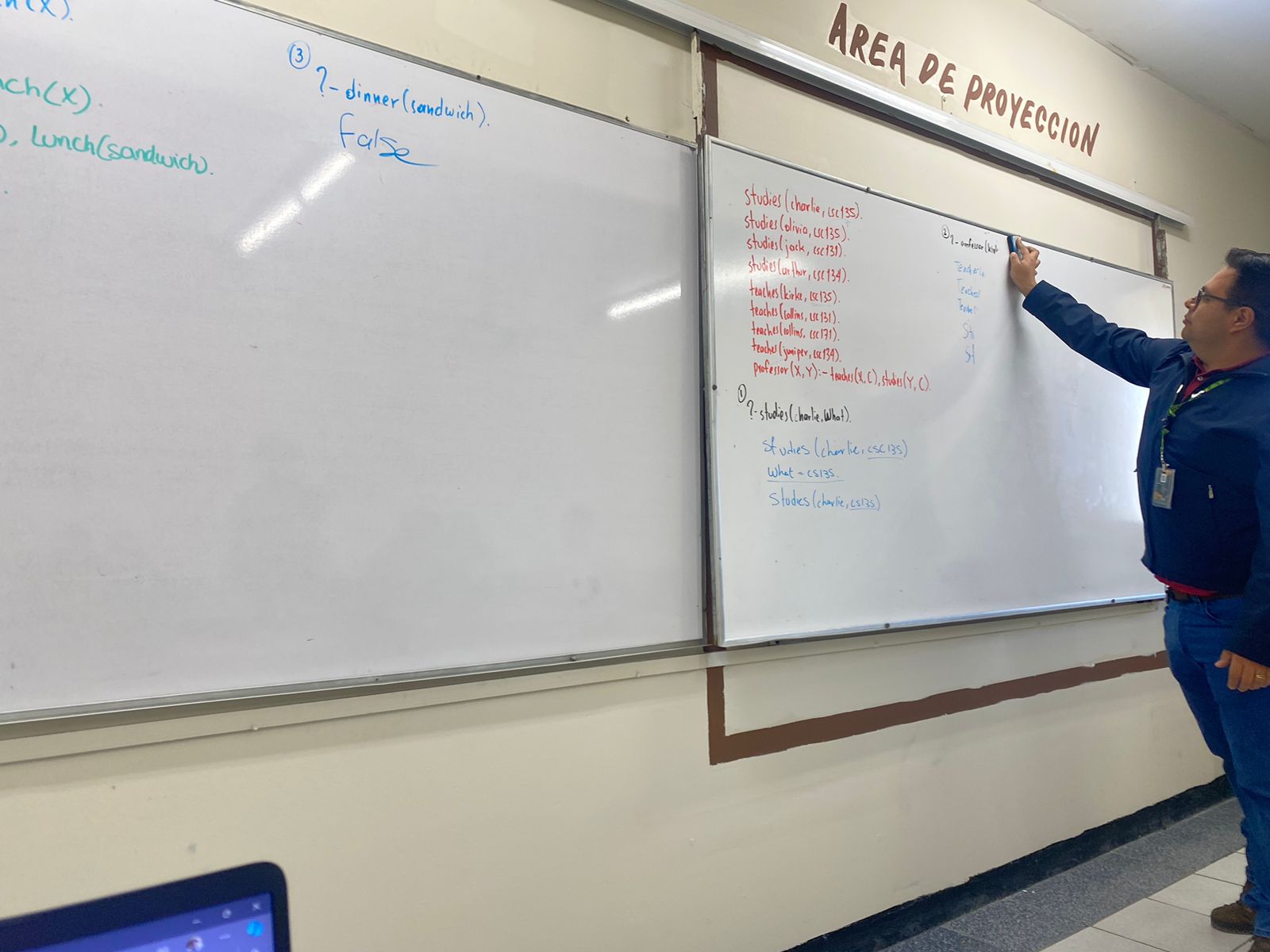
**Explicación:**

1. **Evaluación de la consulta:**

* La consulta pregunta si dinner(sandwich) es verdadero.
* Sin embargo, en los hechos dados, dinner(sandwich) no está declarado como verdadero.

1. **Resultado:**

* La consulta resulta en false porque no hay ninguna declaración que indique que un sándwich es una cena (dinner(sandwich)).

**Reglas:**

studies(charlie,csc135)

studies(olivia,csc135)

studies(jack,csc131)

studies(arthur,csc134)

teaches(kirke,csc135)

teaches(collins,csc131)

teaches(collins,csc171)

teaches(juniper,csc134)

professor(X,Y):-teaches(X,C),studies(Y,C)

**Consulta:**

1.?-studies(charlie,What) studies(charlie,csc135) What = csc135

**Definición de la regla:**

* professor(X, Y) :- teaches(X, C), studies(Y, C).: Declara que X es profesor de Y si X enseña un curso C y Y estudia ese curso C.

**Consulta y resultado:**

* ?- studies(charlie, What).: La consulta pregunta qué curso estudia Charlie.
* El resultado es What = csc135 porque está definido como un hecho que Charlie estudia el curso CSC135.

**Consulta:**

?- professor(kirke, C).

**Evaluación de la regla:**

* La consulta professor(kirke, C) usa la regla professor(X, Y) :- teaches(X, C), studies(Y, C).
* Aquí, X es kirke, entonces la consulta se convierte en teaches(kirke, Course), studies(C, Course).

**Búsqueda de C:**

* Primero, se busca un curso que Kirke enseñe. De los hechos, teaches(kirke, csc135) es verdadero.
* Luego, se busca un estudiante C que estudie el curso csc135.

**Evaluación de los hechos:**

* studies(charlie, csc135) es verdadero.
* studies(olivia, csc135) es verdadero.
* Entonces, tanto charlie como olivia cumplen las condiciones.

**Resultado:**

C = charlie ;

C = olivia.