

# TRABAJO PRÁCTICO FINAL

CONCEPTOS Y PARADIGMAS DE LENGUAJES DE PROGRAMACIÓN

**Cristian Villafañe**  
**2025**

# Índice

<b>Índice</b>	<b>2</b>
Empresa propuesta	3
Productos Analizados	3
Solicitud del Análisis	3
<b>Trabajo Realizado</b>	<b>4</b>
Funcionalidades del Sistema	4
<b>Conclusiones</b>	<b>5</b>
Comparación con otro Paradigma	5
Reflexión final	5
<b>Código</b>	<b>6</b>
Hechos y relaciones	6
Consultas	12

# Empresa propuesta

## Cafetería UNLa

**Cafetería UNLa** es una cafetería que provee de productos comestibles y se encuentra en proceso de innovar sus productos, por lo que requiere hacer un relevamiento de estos para analizar la viabilidad de la inversión.

## Productos Analizados

Se analizaron **5 productos** (prod1, prod2, prod3, prod4, prod5) donde se buscara clientes frecuentes y se les consultará por:

- Nombre
- Rango etario (18-24 / 25-39 / 40+)
- Género
- Producto en análisis
- Aceptación del producto (sí o no)
- Razón de aceptación o rechazo
- Precio dispuesto a pagar en caso de aceptación.

## Solicitud del Análisis

CafeteríaUNLa necesitaba un sistema que procese las encuestas realizadas y pueda generar reportes sobre:

- Producto con más y menos aceptación.
- Distintos tipos de listados (por ejemplo: listado de productos, encuestas por producto, etc.).
- ¿Cuál es el rango de edad que más acepta cada producto?
- ¿Cuál es el rango de edad que menos acepta cada producto?
- ¿Cuál es el género que más acepta cada producto?
- ¿Cuál es el género que menos acepta cada producto?
- Cantidad de encuestados en general.
- Cantidad de encuestas de aceptación y rechazo.
- Razón principal de aceptación y rechazo (la más mencionada) de cada producto.
- ¿Cuánto estarían dispuestos a pagar los encuestados que aceptan cada producto?

# Trabajo Realizado

Se desarrolló un **sistema en Prolog** que:

- Procesa 50 encuestas de personas diferentes
- Identifica el producto más y menos aceptado
- Analiza preferencias por edad (18-24, 25-39, 40-99 años) y género
- Calcula las razones más comunes de aceptación/rechazo
- Obtiene el precio promedio que pagarían por cada producto

## Funcionalidades del Sistema

El sistema implementa múltiples consultas especializadas:

### Análisis de Participación:

- Cuenta el total de encuestados únicos
- `cantidad_encuestados(C) .`
- Calcula la cantidad de encuestas por tipo de aceptación (si/no)
- `cantidad_de_encuestas_por_aceptacion(si, C) .`

### Evaluación de Productos:

- Determina cuál es el producto con mayor aceptación
- `producto_mas_aceptado(X, C) .`
- Identifica el producto con menor rechazo
- `producto_menos_aceptado(X,C) .`
- Genera listados de encuestas filtradas por producto
- `encuestas_por_producto(prodl, X) .`

### Segmentación Etaria:

- Analiza qué rango etario acepta más cada producto
- `rango_mayor_aceptacion_por_producto(L) .`
- Identifica qué género tiene mayor preferencia por producto
- `genero_mayor_aceptacion_por_producto(L) .`

### Análisis de Motivaciones:

- Encuentra las razones principales de aceptación (sabor, calidad, novedad, marca)
- `principales_razones_por_producto(si, L) .`
- Identifica los motivos más comunes de rechazo (precio, calidad, marca)
- `principales_razones_por_producto(no, L) .`

### Evaluación Económica:

- Calcula precios promedio que están dispuestos a pagar por producto
- `listado_de_precios_promedios_por_producto(L) .`

## Conclusiones

El desarrollo del sistema de análisis de encuestas en **Prolog** demostró ser una elección acertada para este tipo de problema aunque un reto para quienes no manejan este paradigma. El paradigma lógico permitió modelar el conocimiento de manera natural y declarativa, donde cada encuesta se representa como un hecho simple, con relaciones que se consultan entre sí para llegar a las consultas solicitadas.

### Ventajas del paradigma lógico:

- **Flexibilidad:** El sistema permite agregar nuevos tipos de consultas sin modificar la base de datos
- **Mantenimiento simple:** Agregar nuevas encuestas es tan fácil como añadir hechos
- **Consultas complejas:** Permite escalar las consultas haciendo uso de consultas previas ya creadas.

## Comparación con otro Paradigma

### Si se hubiera desarrollado en Programación Orientada a Objetos (Java/C#):

- Habríamos creado una clase Encuesta con sus métodos
- Tendríamos que programar nosotros los algoritmos para buscar y filtrar datos
- Sería más código, pero quizás más fácil de entender para alguien que no conoce Prolog
- Tendríamos más control sobre cómo funciona, pero también más trabajo

## Reflexión final

Elegir Prolog fue acertado porque nuestro problema es básicamente "hacer preguntas sobre datos". Prolog está diseñado exactamente para eso. En lugar de preocuparnos por cómo buscar la información, nos enfocamos en qué información necesitábamos.

Al principio fue raro acostumbrarse a pensar en lógica en lugar de pasos, pero una vez que le agarramos la mano, resultó mucho más simple que otros lenguajes para este tipo de problema.

# Código

## Hechos y relaciones

```
:- dynamic encuesta/7.

% Base de conocimientos
generos([m, f]).
productos([prod1, prod2, prod3, prod4, prod5]).
rango_etario(['18-24', '25-39', '40-99']).

% encuesta(Cliente, RangoEtario, Sexo, Producto, Acepta, Razon, PrecioMaximo).
encuesta(juan, '18-24', m, prod1, si, sabor, 150).
encuesta(maria, '18-24', f, prod1, no, precio, 0).
encuesta(luis, '25-39', m, prod1, si, calidad, 300).
encuesta(ana, '25-39', f, prod1, no, marca, 0).
encuesta(carla, '40-99', f, prod1, si, sabor, 450).
encuesta(ricardo, '40-99', m, prod1, si, calidad, 600).
encuesta(paula, '25-39', f, prod1, no, precio, 0).
encuesta(mateo, '18-24', m, prod1, si, novedad, 200).
encuesta(lucas, '40-99', m, prod1, no, precio, 0).
encuesta(laura, '25-39', f, prod1, si, sabor, 500).

encuesta(juan, '18-24', m, prod2, no, marca, 0).
encuesta(silvia, '40-99', f, prod2, no, calidad, 0).
encuesta(diego, '18-24', m, prod2, no, sabor, 0).
encuesta(julietta, '25-39', f, prod2, si, novedad, 350).
encuesta(marcos, '40-99', m, prod2, no, calidad, 0).
encuesta(celeste, '25-39', f, prod2, no, precio, 0).
encuesta(rodolfo, '25-39', m, prod2, si, sabor, 800).
encuesta(agustina, '18-24', f, prod2, no, marca, 0).
encuesta(esteban, '25-39', m, prod2, si, novedad, 400).
encuesta(monica, '25-39', f, prod2, si, sabor, 550).

encuesta(juan, '18-24', m, prod3, no, precio, 0).
encuesta(andrea, '25-39', f, prod3, si, calidad, 600).
encuesta(leandro, '18-24', m, prod3, si, marca, 300).
encuesta(valeria, '40-99', f, prod3, no, sabor, 0).
encuesta(emanuel, '25-39', m, prod3, no, calidad, 0).
encuesta(romina, '25-39', f, prod3, no, novedad, 0).
encuesta(nicolas, '25-39', m, prod3, si, sabor, 400).
encuesta(vanesa, '25-39', f, prod3, no, precio, 0).
encuesta(tomas, '25-39', m, prod3, si, novedad, 380).
encuesta(sabrina, '40-99', f, prod3, si, marca, 420).

encuesta(juan, '18-24', m, prod4, si, marca, 400).
encuesta(daiana, '18-24', f, prod4, si, sabor, 400).
encuesta(sebastian, '25-39', m, prod4, no, precio, 0).
encuesta(natalia, '25-39', f, prod4, si, calidad, 400).
encuesta(brian, '18-24', m, prod4, si, sabor, 400).
encuesta(evelyn, '25-39', f, prod4, no, marca, 0).
encuesta(pablo, '40-99', m, prod4, si, novedad, 400).
encuesta(camila, '25-39', f, prod4, si, sabor, 400).
encuesta(joaquin, '40-99', m, prod4, si, sabor, 400).
encuesta(sol, '25-39', f, prod4, si, calidad, 400).

encuesta(juan, '18-24', f, prod5, si, calidad, 690).
```

```

encuesta(cynthia, '25-39', f, prod5, no, precio, 0).
encuesta(franco, '18-24', m, prod5, si, novedad, 200).
encuesta(aylen, '25-39', f, prod5, si, sabor, 610).
encuesta(gaston, '25-39', m, prod5, no, precio, 0).
encuesta(noelia, '18-24', f, prod5, si, calidad, 610).
encuesta(pedro, '40-99', m, prod5, no, precio, 0).
encuesta(yanina, '40-99', f, prod5, si, calidad, 730).
encuesta(juliana, '40-99', f, prod5, si, novedad, 560).
encuesta(melina, '25-39', f, prod5, no, precio, 0).

/* ***** ENCUESTADOS ***** */

% CONSULTA: Contar el número total de personas diferentes que participaron en las
encuestas
% MÉTODO: Obtiene todos los nombres de clientes, elimina duplicados y cuenta
% EJEMPLO: Si Juan aparece en 5 encuestas, se cuenta solo una vez
cantidad_encuestados(Cantidad) :-
    findall(Cliente, encuesta(Cliente, _, _, _, _, _), TodosLosClientes),
    list_to_set(TodosLosClientes, ClientesUnicos),
    length(ClientesUnicos, Cantidad).

% CONSULTA: Contar cuántas encuestas tienen una aceptación específica (si/no)
% MÉTODO: Busca todas las encuestas con la aceptación dada y las cuenta
% EJEMPLO: cantidad_de_encuestas_por_aceptacion(si, X) devuelve total de
aceptaciones
cantidad_de_encuestas_por_aceptacion(Aceptacion, Cantidad) :-
    findall(1, encuesta(_, _, _, _, Aceptacion, _, _), ListaEncuestas),
    length(ListaEncuestas, Cantidad).

% CONSULTA: Listar encuestas por producto
% MÉTODO: Filtra encuestas por producto y las lista
% EJEMPLO: encuestas_por_producto(prodl, Lista) devuelve todas las encuestas de
prodl
encuestas_por_producto(Producto, Lista) :-
    findall(
        encuesta(Cliente, RangoEtario, Sexo, Producto, Acepta, Razon,
PrecioMaximo),
        encuesta(Cliente, RangoEtario, Sexo, Producto, Acepta, Razon,
PrecioMaximo),
        Lista
    ).

/* ***** ACEPTACIÓN ***** */

% CONSULTA: Contar aceptaciones de un producto específico
% MÉTODO: Filtra encuestas por producto y tipo de aceptación (si/no), cuenta los
resultados
% EJEMPLO: aceptaciones_producto(prodl, si, X) cuenta cuántos aceptaron prodl
aceptaciones_producto(Producto, Aceptacion, Cantidad) :-
    findall(1, encuesta(_, _, _, Producto, Aceptacion, _, _), Lista),
    length(Lista, Cantidad).

% CONSULTA: Crear lista con aceptaciones de todos los productos
% MÉTODO: Para cada producto, cuenta las aceptaciones y arma tuplas (Producto,
Cantidad)

```

```

% EJEMPLO: aceptaciones_todos_productos(si, L) devuelve [(prod1,5), (prod2,3), ...]
aceptaciones_todos_productos(Aceptacion, Lista) :-
    productos(Productos),
    findall(
        (Producto, Cantidad),
        (
            member(Producto, Productos),
            aceptaciones_producto(Producto, Aceptacion, Cantidad)
        ),
        Lista
    ).

% CONSULTA AUXILIAR: Encontrar el máximo en una lista de tuplas (Producto,
Cantidad)
% MÉTODO: Compara recursivamente las cantidades para encontrar el mayor
max_aceptacion([(Prod, Cant)], (Prod, Cant)).
max_aceptacion([(Prod1, Cant1), (_, Cant2) | Resto], Max) :-
    Cant1 >= Cant2,
    max_aceptacion([(Prod1, Cant1) | Resto], Max).
max_aceptacion([(_, Cant1), (Prod2, Cant2) | Resto], Max) :-
    Cant1 < Cant2,
    max_aceptacion([(Prod2, Cant2) | Resto], Max).

% CONSULTA: Identificar el producto más exitoso
% MÉTODO: Obtiene todas las aceptaciones positivas y encuentra el máximo
% RESULTADO: Devuelve el producto y su cantidad de aceptaciones
producto_mas_aceptado(Producto, Cantidad) :-
    aceptaciones_todos_productos(si, ListaAceptaciones),
    max_aceptacion(ListaAceptaciones, (Producto, Cantidad)).

% CONSULTA AUXILIAR: Encontrar el mínimo en una lista de tuplas (Producto,
Cantidad)
% MÉTODO: Compara recursivamente las cantidades para encontrar el menor
min_aceptacion([(Prod, Cant)], (Prod, Cant)).
min_aceptacion([(Prod1, Cant1), (_, Cant2) | Resto], Min) :-
    Cant1 <= Cant2,
    min_aceptacion([(Prod1, Cant1) | Resto], Min).
min_aceptacion([(_, Cant1), (Prod2, Cant2) | Resto], Min) :-
    Cant1 > Cant2,
    min_aceptacion([(Prod2, Cant2) | Resto], Min).

% CONSULTA: Identificar el producto con menor rechazo
% MÉTODO: Obtiene todos los rechazos y encuentra el mínimo
% RESULTADO: Devuelve el producto con menos rechazos y su cantidad
producto_menos_aceptado(Producto, Cantidad) :-
    aceptaciones_todos_productos(no, ListaAceptaciones),
    min_aceptacion(ListaAceptaciones, (Producto, Cantidad)).

/* ***** RANGOS ETARIOS ***** */

% CONSULTA: Contar aceptaciones por producto, rango de edad y tipo de aceptación
% MÉTODO: Filtra encuestas por los tres criterios y cuenta los resultados
% EJEMPLO: Para saber cuántos jóvenes (18-24) aceptaron prod1
cantidad_aceptaciones_por_rango_de_edad(Producto, Rango, Aceptacion, Cantidad) :-
    findall(1, encuesta(_, Rango, _, Producto, Aceptacion, _, _), Lista),
    length(Lista, Cantidad).

```



```

% CONSULTA: Encontrar el rango etario con mayor aceptación/rechazo para un producto
% MÉTODO: Evalúa todos los rangos etarios y encuentra el que tiene más encuestas
del tipo especificado
% RESULTADO: Devuelve el rango ganador y la cantidad máxima encontrada
rango_y_cantidad_mayor_por_aceptacion_y_producto(Producto, Aceptacion, RangoMayor,
CantidadMaxima) :-
    rango_etario(Rangos),
    findall(
        Cantidad-Rango,
        (
            member(Rango, Rangos),
            cantidad_aceptaciones_por_rango_de_edad(Producto, Rango, Aceptacion,
Cantidad)
        ),
        Lista
    ),
    max_member(CantidadMaxima-RangoMayor, Lista).

% CONSULTA: Crear reporte de rangos etarios con mayor aceptación por cada producto
% MÉTODO: Para cada producto, encuentra su rango etario de mayor aceptación
% RESULTADO: Lista de tuplas (Producto, RangoPreferido, CantidadAceptaciones)
rango_mayor_aceptacion_por_producto(ListaRangos) :-
    findall(
        Producto-RangoMayor-CantidadMaxima,
        ( productos(Productos),
            member(Producto, Productos),
            rango_y_cantidad_mayor_por_aceptacion_y_producto(Producto, si, RangoMayor,
CantidadMaxima)
        ),
        ListaRangos).

% CONSULTA: Crear reporte de rangos etarios con mayor rechazo por cada producto
% MÉTODO: Para cada producto, encuentra su rango etario de mayor rechazo
(aceptacion no)
% RESULTADO: Lista de tuplas (Producto, RangoQueRechaza, CantidadRechazos)
rango_menor_aceptacion_por_producto(ListaRangos) :-
    findall(
        Producto-RangoMayor-CantidadMaxima,
        ( productos(Productos),
            member(Producto, Productos),
            rango_y_cantidad_mayor_por_aceptacion_y_producto(Producto, no, RangoMayor,
CantidadMaxima)
        ),
        ListaRangos).

/* ***** GENERO ***** */

% CONSULTA: Contar aceptaciones por producto, género y tipo de aceptación
% MÉTODO: Filtra encuestas por los tres criterios especificados
% EJEMPLO: Cuántas mujeres aceptaron prod1, cuántos hombres lo rechazaron, etc.
cantidad_aceptaciones_por_producto_y_genero(Producto, Genero, Aceptacion, Cantidad)
:-
    findall(1, encuesta(_, _, Genero, Producto, Aceptacion, _, _), Lista),
    length(Lista, Cantidad).

```

```

% CONSULTA: Encontrar el género con mayor aceptación/rechazo para un producto
% MÉTODO: Compara las cantidades de aceptación entre géneros para un producto
% RESULTADO: Devuelve el género predominante y su cantidad de respuestas
aceptacion_de_producto_por_genero(Producto, Aceptacion, GeneroMayor,
CantidadMaxima) :-
    generos(Generos),
    findall(
        Cantidad-Genero,
        (
            member(Genero, Generos),
            cantidad_aceptaciones_por_producto_y_genero(Producto, Genero, Aceptacion,
Cantidad)
        ),
        Lista
    ),
    % Encontrar el máximo
    max_member(CantidadMaxima-GeneroMayor, Lista).

% CONSULTA: Reporte de género predominante en aceptación para cada producto
% MÉTODO: Para cada producto, identifica el género con más aceptaciones
% RESULTADO: Lista de tuplas (Producto, GeneroQueAcepta, CantidadAceptaciones)
genero_mayor_aceptacion_por_producto(Lista) :-
    findall(
        Producto-GeneroMayor-CantidadMaxima,
        (
            productos(Productos),
            member(Producto, Productos),
            aceptacion_de_producto_por_genero(Producto, si, GeneroMayor,
CantidadMaxima)
        ),
        Lista).

% CONSULTA: Reporte de género predominante en rechazo para cada producto
% MÉTODO: Para cada producto, identifica el género con más rechazos
% RESULTADO: Lista de tuplas (Producto, GeneroQueRechaza, CantidadRechazos)
genero_menor_aceptacion_por_producto(Lista) :-
    findall(
        Producto-GeneroMenor-CantidadMaxima,
        (
            productos(Productos),
            member(Producto, Productos),
            aceptacion_de_producto_por_genero(Producto, no, GeneroMenor,
CantidadMaxima)
        ),
        Lista).

/* ***** RAZONES ***** */

% CONSULTA: Contar encuestas por producto, razón y tipo de aceptación
% MÉTODO: Filtra encuestas que coincidan con los tres criterios
% EJEMPLO: Cuántos aceptaron prod1 por "sabor", cuántos rechazaron prod2 por
"precio"
cantidad_de_encuestas_por_producto_aceptacion_y_razon(Producto, Razon, Aceptacion,
Cantidad) :-
    % Encontrar todas las encuestas donde se evaluó el producto
    findall(1, encuesta(_, _, _, Producto, Aceptacion, Razon, _), Lista),

```

```

% Contar cuántas encuestas encontré
length(Lista, Cantidad).

% CONSULTA: Obtener todas las razones mencionadas para un producto con sus
cantidades
% MÉTODO: Encuentra todas las razones únicas y cuenta cuántas veces aparece cada
una
% RESULTADO: Lista de tuplas (Razon, CantidadVeces) para el producto especificado
todas_las_razones_con_cantidad(Producto, Aceptacion, Lista) :-
    findall(Razon, encuesta(_, _, _, Producto, Aceptacion, Razon, _),
TodasRazones),
    sort(TodasRazones, RazonesUnicas), % sort elimina duplicados automáticamente
    findall(
        (Razon, Cant),
        (member(Razon, RazonesUnicas),
cantidad_de_encuestas_por_producto_aceptacion_y_razon(Producto, Razon, Aceptacion,
Cant)),
        Lista
    ).

% CONSULTA: Identificar la razón principal de aceptación/rechazo para cada producto
% MÉTODO: Para cada producto, obtiene todas sus razones y selecciona la más
frecuente
% RESULTADO: Lista de tuplas (Producto, RazonPrincipal, CantidadVeces)
principales_razones_por_producto(Aceptacion, Lista) :-
    productos(Productos),
    findall(
        (Producto, Razon, Cantidad),
        (
            member(Producto, Productos),
            todas_las_razones_con_cantidad(Producto, Aceptacion, ListaRazones),
            member((Razon, Cantidad), ListaRazones),
            max_razon(ListaRazones, (Razon, Cantidad))
        ),
        Lista
    ).

% CONSULTA AUXILIAR: Encontrar la razón con mayor cantidad en una lista
% MÉTODO: Compara recursivamente las cantidades de cada razón
max_razon([(Razon, Cantidad)], (Razon, Cantidad)).
max_razon([(Razon1, Cantidad1), (_, Cantidad2) | Resto], Max) :-
    Cantidad1 >= Cantidad2,
    max_razon([(Razon1, Cantidad1) | Resto], Max).
max_razon([(_, Cantidad1), (Razon2, Cantidad2) | Resto], Max) :-
    Cantidad1 < Cantidad2,
    max_razon([(Razon2, Cantidad2) | Resto], Max).

/* ***** PRECIO A PAGAR
***** */

% CONSULTA: Calcular el precio promedio que están dispuestos a pagar por un
producto
% MÉTODO: Obtiene todos los precios de quienes aceptaron el producto y calcula el
promedio
% MANEJO DE ERRORES: Si no hay aceptaciones, devuelve 0 para evitar división por
cero
precio_promedio_por_producto_aceptado(Producto, Promedio) :-
    findall(Precio, encuesta(_, _, _, Producto, si, _, Precio), ListaPrecios),

```

```

sum_list(ListaPrecios, Total),
length(ListaPrecios, Cantidad),
(Cantidad > 0 -> Promedio is Total / Cantidad ; Promedio is 0).

% CONSULTA: Crear reporte de precios promedio para todos los productos
% MÉTODO: Para cada producto, calcula su precio promedio de aceptación
% RESULTADO: Lista de tuplas (Producto, PrecioPromedio) ordenada por producto
listado_de_precios_promedios_por_producto(Lista) :-
    productos(Productos),
    findall(
        (Producto, PrecioPromedio),
        (
            member(Producto, Productos),
            precio_promedio_por_producto_aceptado(Producto, PrecioPromedio)
        ),
        Lista
    ).

```

## Consultas

% ✓ Producto con más aceptación.  
 producto\_mas\_aceptado(X, C).

% ✓ Producto con menos aceptación.  
 producto\_menos\_aceptado(X,C).

% ✓ Distintos tipos de listados (por ejemplo: listado de productos, encuestas por producto, etc.).  
 productos(X).  
 encuestas\_por\_producto(prod1, X).

% ✓ ¿Cuál es el rango de edad y género que más acepta cada producto?  
 rango\_mayor\_aceptacion\_por\_producto(L).  
 genero\_mayor\_aceptacion\_por\_producto(L).

% ✓ ¿Cuál es el rango de edad y género que menos acepta cada producto?  
 rango\_menor\_aceptacion\_por\_producto(L).  
 genero\_menor\_aceptacion\_por\_producto(L).

% ✓ Cantidad de encuestados.  
 cantidad\_encuestados(C).

% ✓ Cantidad de encuestas de aceptación, en general.  
 cantidad\_de\_encuestas\_por\_aceptacion(si, C).

% ✓ Cantidad de encuestas de no aceptación, en general.  
 cantidad\_de\_encuestas\_por\_aceptacion(no, C).

% ✓ Razón principal de aceptación (la más mencionada) de cada producto.

principales\_razones\_por\_producto(si, L).

% ✓ Razón principal de no aceptación (la más mencionada) de cada producto.

principales\_razones\_por\_producto(no, L).

% ✓ ¿Cuánto estarían dispuestos a pagar los encuestados que aceptan cada producto?

listado\_de\_precios\_promedios\_por\_producto(L).