



Materia:	Programación Lógica y Funcional		
Práctica:	Ejercicio: 99 botellas de cerveza en la pared		
Alumno(s):	Luis Manuel Cárdenas Ibarra	Fecha:	01/04/2025

Objetivo

El objetivo de este trabajo fue implementar en Prolog la canción tradicional estadounidense "99 Bottles of Beer", utilizando recursión para manejar el conteo regresivo desde 99 hasta 0 botellas. Además, se buscó:

- Analizar la estructura de la canción para modelarla en reglas lógicas.
- Aplicar el concepto de recursión para evitar repetición de código.
- Permitir flexibilidad para iniciar la canción desde cualquier número de botellas.

Desarrollo

Lo primero que podemos notar es que la canción sigue un patrón repetitivo con la siguiente estructura:

[N] bottles of beer on the wall, [N] bottles of beer.

Take one down, pass it around, [N-1] bottles of beer on the wall.

Ademas de algunos casos especiales, como los siguientes:

- 0 botellas: Cambia la última línea ("Go to the store...").
- 1 botella: Cambia "bottles" → "bottle".
- 2 botellas: Transición correcta a "1 bottle".

Lo primero fue desarrollar esos casos especiales, con las siguientes reglas:

sing(0) :-

write('No more bottles of beer on the wall, no more bottles of beer. '), nl,

write('Go to the store and buy some more, 99 bottles of beer on the wall. '), nl, nl.



sing(1) :-

```
write('1 bottle of beer on the wall, 1 bottle of beer. '), nl,  
write('Take one down and pass it around, no more bottles of beer on the wall. '), nl, nl,  
Next is 0,  
sing(Next).
```

sing(2) :-

```
write('2 bottles of beer on the wall, 2 bottles of beer. '), nl,  
write('Take one down and pass it around, 1 bottle of beer on the wall. '), nl, nl,  
Next is 1,  
sing(Next).
```

Una vez definidos los casos especiales, podemos continuar con el resto de números que si siguen el patrón consecutivo sin cambios.

sing(N) :-

```
N > 2,  
write(N), write(' bottles of beer on the wall, '), write(N), write(' bottles of beer. '), nl,  
write('Take one down and pass it around, '),  
Next is N - 1,  
write(Next), write(' bottles of beer on the wall. '), nl, nl,  
sing(Next).
```

Y el predicado principal, con el que daremos inicio al programa

ninety_nine_bottles :-

```
sing(99).
```



Resultado

- El programa genera correctamente todas las estrofas desde **N** hasta **0**.

```
SWI-Prolog -- d:/ProgLogyFunc/prolog/99BottlesOfBeer/99BottleOfBeer.pl
File Edit Settings Run Debug Help

65 bottles of beer on the wall, 65 bottles of beer.
Take one down and pass it around, 64 bottles of beer on the wall.

64 bottles of beer on the wall, 64 bottles of beer.
Take one down and pass it around, 63 bottles of beer on the wall.

63 bottles of beer on the wall, 63 bottles of beer.
Take one down and pass it around, 62 bottles of beer on the wall.

62 bottles of beer on the wall, 62 bottles of beer.
Take one down and pass it around, 61 bottles of beer on the wall.

61 bottles of beer on the wall, 61 bottles of beer.
Take one down and pass it around, 60 bottles of beer on the wall.

60 bottles of beer on the wall, 60 bottles of beer.
Take one down and pass it around, 59 bottles of beer on the wall.
```

- Maneja adecuadamente los casos especiales (0, 1 y 2 botellas).

```
4 bottles of beer on the wall, 4 bottles of beer.
Take one down and pass it around, 3 bottles of beer on the wall.

3 bottles of beer on the wall, 3 bottles of beer.
Take one down and pass it around, 2 bottles of beer on the wall.

2 bottles of beer on the wall, 2 bottles of beer.
Take one down and pass it around, 1 bottle of beer on the wall.

1 bottle of beer on the wall, 1 bottle of beer.
Take one down and pass it around, no more bottles of beer on the wall.

No more bottles of beer on the wall, no more bottles of beer.
Go to the store and buy some more, 99 bottles of beer on the wall.
```



- Permite iniciar desde cualquier número con sing(N).

```
?- sing(5).  
5 bottles of beer on the wall, 5 bottles of beer.  
Take one down and pass it around, 4 bottles of beer on the wall.  
  
4 bottles of beer on the wall, 4 bottles of beer.  
Take one down and pass it around, 3 bottles of beer on the wall.  
  
3 bottles of beer on the wall, 3 bottles of beer.  
Take one down and pass it around, 2 bottles of beer on the wall.  
  
2 bottles of beer on the wall, 2 bottles of beer.  
Take one down and pass it around, 1 bottle of beer on the wall.  
  
1 bottle of beer on the wall, 1 bottle of beer.  
Take one down and pass it around, no more bottles of beer on the wall.  
  
No more bottles of beer on the wall, no more bottles of beer.  
Go to the store and buy some more, 99 bottles of beer on the wall.
```

Conclusiones

Este ejercicio refuerza la importancia de identificar casos base y reglas generales en la programación lógica, así como la capacidad de Prolog para manejar problemas secuenciales mediante recursión. El resultado cumple con el objetivo principal y sirve como un ejemplo claro de cómo la recursión puede aplicarse en contextos prácticos. En conclusión, el proyecto no solo logra replicar la canción de manera funcional, sino que también ilustra conceptos fundamentales de programación en Prolog de manera didáctica y efectiva.