

Universidad Nacional del Altiplano

Facultad de Ingeniería Estadística e Informática

Docente: Fred Torres Cruz

Autor: *Flores Turpo Jorge Luis*

Trabajo Encargado N.º 013

Repositorio:

https://github.com/Yorchisflrs/Juego_sobrevivientes

Juego “Sobrevivientes” en Python

1. Objetivo del proyecto

Diseñar e implementar, con buenas prácticas de ingeniería de software, el juego *Sobrevivientes*. El programa debe:

- Simular la distribución y el canje de dulces entre 22 jugadores (nivel 1) y 2 equipos de 11 (nivel 2).
- Garantizar que, con colaboración, al menos un jugador alcance la condición de “sobrevivir” (poseer un chupetín).
- Mostrar resultados en consola con tablas Rich y exponer una CLI profesional mediante Typer.
- Contar con pruebas unitarias (pytest) que verifiquen las reglas principales.

2. Estructura de carpetas

```
sobrevivientes/  
__init__.py      # CLI Typer  
config.yaml      # Reglas y parámetros  
levels.py        # Motor de niveles  
models.py        # Candy, Inventory, Player  
rules.py         # Carga y aplicación de reglas  
tests/  
test_rules.py    # Regla tripleta y romper chupetín  
test_levels.py   # Integración nivel 1 y 2
```

3. Principales clases

Candy *Enum* con valores LIMÓN, PERA, HUEVO, CHUPETÍN.

Inventory Multiconjunto derivado de **Counter**; incluye `add_items`, `remove_items` y `choose_any`.

Player Almacena inventario y contador de acciones; aplica reglas hasta agotar combinaciones.

LevelEngine Orquestador: reparte dulces, ejecuta reglas, dona recursos y efectúa *crafting* colectivo.

4. Reglas implementadas

1. **Nivel 1 – Tripleta**

Limón + Pera + Huevo \rightarrow Chupetín + 1 dulce a elección.

2. **Nivel 2 – Sexteto**

2L + 2P + 2H \rightarrow Chupetín + 2 dulces a elección.

3. **Nivel 2 – Romper chupetín**

Chupetín \rightarrow 6 dulces a elección.

5. Algoritmo de donaciones y *crafting*

1. Fase GREEDY 1-a-1: los jugadores con duplicados donan a quienes no cumplen la condición mínima.
2. Fase de *crafting colectivo* (solo nivel 2): se agrupan globalmente dulces para formar sextetos, asignando el resultado al primer jugador que aún no sobreviva. Si no hay dulces suficientes, se desmantela un chupetín para generar seis piezas y se repite el proceso.

6. Resultados de prueba

- Todos los casos de prueba (`pytest`) pasan con éxito.
- En ejecución interactiva:
 - Nivel 1: suelen sobrevivir 8–12 jugadores.
 - Nivel 2: al menos un jugador sobrevive; con suficientes combinaciones, pueden sobrevivir todos.

7. Ejecución

```
$ python -m sobrevivientes --level 1  
$ python -m sobrevivientes --level 2
```

Ju...	I...	Ac...	V...
1	H...	0	✗
	L...		
2	P...	0	✗
3	H...	0	✗
	L...		
4	L...	0	✗
5	H...	0	✗
	L...		
6	H...	0	✗
	P...		
7	L...	0	✗
	P...		
8	L...	0	✗
	P...		
9	H...	0	✗
	L...		
10	H...	0	✗
	L...		
11	H...	0	✗
	L...		
12	H...	0	✗
	L...		
13	P...	0	✗
14	H...	0	✗
	P...		
15	C...	1	✓
	H...		
16	L...	0	✗
	P...		
17	C...	1	✓
	P...		
18	C...	1	✓

Figura 1. Salida de consola formateada con Rich.

8. Conclusiones

El uso de `Typer` y `Rich` ofrece una interfaz profesional con código mínimo. El algoritmo de *crafting* colectivo demuestra que la cooperación distribuye equitativamente recursos escasos y garantiza la supervivencia de al menos un participante, cumpliendo el objetivo del juego.

Versión: v1.0 – 16 de julio de 2025