

**ADDA**

## 1. Introducción

Problema de Pociones	
Técnica	<b>Programación Dinámica</b>
Tamaño	<b>despensa.size() - i</b>
Propiedades Compartidas	<ul style="list-style-type: none"> <li>List&lt;Potion&gt;<b>despensa</b>: Guarda todas las pociones</li> <li>Integer <b>nIni</b>: nivel inicial oponente</li> <li>TipoPersonaje <b>tp</b>: dirá si es nigromante o no</li> </ul>
Propiedades Individuales	<ul style="list-style-type: none"> <li>Integer <b>i</b> en [0, despensa.size]: posición actual de la lista</li> <li>Integer <b>nAct</b>: nivel actual</li> <li>Integer <b>cAcu</b>: coste acumulado</li> </ul>
Solución	$s = (as, ct, dt)$ ; siendo: <ul style="list-style-type: none"> <li>as: MultiSet de alternativas seleccionadas en el paso</li> <li>ct: coste total</li> <li>dt: daño total</li> </ul>
Objetivo	Encontrar "s" tal que "dt >= nIni" y "ct" tenga el menor valor posible
Solución Parcial	<b>Sp = (a, ct)</b>
Alternativas	$A_{cAcu}, i, despensa = \{a:k...0\};$ $k = \min(can, nAct/can);$ $can = despensa.get(i).getCant();$ Falta filtro de si es nigromante o no
Instanciación	$pp(despensa, nAct, tipo) = ppg(despensa, nIni, tipo, 0, nIni, 0)$
Problema Generalizado	$ppg(X, p) = \begin{cases} (null, 0) & i > d.length() \text{ or } nAct <= 0 \\ cA_{a \in A_{co,i,d}}(c(X, a, ppg(sp))) & \text{ecoc} \end{cases}$ <p> <math>co=cAcu; d=despensa;</math>  <math>p=(i, nAct, cAcu)</math>  <math>X=(despensa, nIni, tipo)</math>  <math>sp=(i++, d.get(i).getDano()*a + nAct,</math>  <math>d.get(i).getCoste()*a)</math> </p>
sA	$(X, a)=(a, cAcu + pociones.get(i).getCoste()*a)$
Solución reconstruida	? $\hat{}$