

**ADDA**



## 1. Introducción

Problema de Pociones	
Técnica	Programación Dinámica
Tamaño	<code>despensa.size() - i</code>
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> <li>¿Necesaria?? List&lt;Integer&gt;<b>multiplicidades</b>: Guarda todas las multiplicidades de las pociones</li> </ul>
Propiedades Individuales	<ul style="list-style-type: none"> <li>Integer <b>i</b> en [0, <code>despensa.size()</code>]: 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: 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	$Sp = (a, ct)$
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	<code>ProbPociones(0, 0, 0)</code>
Instanciación	<code>ProbPociones(?i, ?i, ?i)</code>
Problema Generalizado	$ppg(p) = \begin{cases} (p[1],) & \text{sin is even} \\ () & i > d.length() \text{ or } nAct >= 0 \\ cA_{a \in A_{co,i,d}}(c(jejeje)) & \text{ecoc} \end{cases}$ <p> <math>co=cAcu</math>; <math>d=despensa</math>;  <math>p=(i, nAct, costeAcumulado)</math>  <math>Sp=(?i,?i)</math> </p>
?i	?i
sA	?i
Solución reconstruida	?i