Sorteio SySDEA-Prova

# Descrição geral

O sorteio de questões no SySDEA-Prova foi projetado com o objetivo de permitir ao elaborador a maior flexibilidade possível no momento de compor os modelos de prova. Entidades e mecanismos se combinam para permitir diversas possibilidades de ajuste macro e micro, tanto no nível do stack quanto no nível das tags, que marcam as questões.

Os modelos de prova conterão slots, que são como se fossem gavetas que o sistema preenche com o conteúdo – as questões de fato sorteadas. Cada uma dessas gavetas aponta para uma ‘pilha de folhas’, de onde o sistema sorteará uma ‘carta’ – a questão escolhida. O processo de pegar essa carta, metaforicamente, é o sorteio. O preenchimento do slot de instância, para continuar a metáfora, é o processo de ‘colocar cada carta na respectiva gaveta’.

O sorteio de questões tem três fases:

1. Preparação;
2. Sorteio, que contém;
3. Consolidação.

Em linhas gerais, a preparação é a fase em que o sistema busca no banco as informações necessárias para instanciar a prova conforme as diretrizes de modelo. A partir desses dados, são geradas tabelas temporárias de controle, que organizam a informação necessária.

A segunda fase, o *Sorteio*, é um loop que se repete uma vez para cada slot – 'ou gaveta'. Nessa rotina, o sistema escolhe randomicamente uma questão no *pool* (a 'pilha de cartas') a questão que vai de fato aparecer na tela do examinador para aplicação daquela instância. O sorteio tem três partes. A primeira parte se chama *Geração das* *Listas de rodada*. Nela o sistema filtra o pool, excluindo do conjunto de questões aquelas que são interditadas ao candidato (por exemplo, porque ele já as fez antes). Por exemplo, o examinador pode ter incluído no slot das questões 1, 2 e 3 um pool contendo 100 questões sorteáveis. Obviamente, uma vez que o candidato já teve contato com uma determinada questão, digamos a 57, esta não deve ser repetida depois. O sistema faz essa exclusão por meio da *Lista de questões interditadas ao candidato*. Quando for rodar a rotina de sorteio de novo, no momento de gerar a Lista de Rodada, a questão 57 será filtrada.

A segunda etapa é a aplicação das tags. As questões serão escolhidas randomicamente, mas haverá pesos e contrapesos a fim de tornar a escolha de algumas mais ou menos provável. As tags controlam esse ajuste fino, e durante a aplicação delas, o sistema calibra as chances. Calibradas as chances, excluídas as questões interditadas, o sorteio finalmente passa para a etapa de escolha, sorteando efetivamente a questão.

Uma vez escolhidas questões para todos os slots, o sistema terá a ‘receita’ para compor a prova: a Tabela de controle de sorteio completamente preenchida. Então tem início a terceira fase, a consolidação, na qual parte das informações geradas no sorteio, incluindo-se aí a 'receita' para a criação da prova, bem como a Lista de questões interditadas ao candidato atualizada, se consolidam.

O que segue é uma descrição das regras que esses sistema deve respeitar. Entretanto, essa descrição não tem por objetivo guiar o desenvolvimento do sistema em termos de classes, entidades, ou mesmo tabelas – afinal, as soluções que podem ser encontradas são diversas e fogem ao escopo desse documento. O que vem a seguir pretende apenas descrever as 'regras do jogo' do ponto de vista lógico ou matemático, se fosse feito por um ser humano de carne e osso fazendo anotações. É como se estivéssemos explicando as regras de um jogo de cartas a ser digitalizado por um software. O alvo do desenvolvimento do SySDEA será automatizar esse processo, sem dúvida, mas de forma alguma a existência de uma 'tabela' ou variável, conforme descritos aqui, implicam numa tabela ou variável real a ser desenvolvida. Desde que sejam alcançados os objetivos e sejam respeitadas as regras matemáticas e lógicas do sorteio, o sistema terá tido êxito.

# Fases do Sorteio

## Fase 1 – Preparação

O objetivo da preparação está em buscar nas tabelas as informações para a administração do sorteio de versão e gerar as tabelas temporárias de sorteio. Isso é feito em cinco etapas:

01 – Criar tabelas e pool;

02 – Criar Tabela de controle de sorteio;

02 – Preencher id dos slots;

03 – Preencher coluna de stacks;

04 – Criar tabelas de controle dos stacks;

05 – Preencher pool;

06 – Criar registro local do candidato.

## Detalhamento da preparação

* + - 1. **Criar tabelas de pool**: Para cada slot utilizado no sorteio, o sistema cria uma tabela temporária chamada de Tabela de pool, listando as questões e as tags do pool relacionado àquele slot. A função da tabela de pool está em promover o acesso rápido às tags relativas à cada questão.

|  |  |
| --- | --- |
| **Tabela de pool** | |
| *Cabeçalho* | |
| Id Pool | 2 |
| Slot | 2 |
| *Registros* | |
| Id Questão | Lista Tags |
| 1 | {1, 2, 3} |
| 2 | {2,3,4} |
| 3 | {1,2,3} |
| ... | ... |
| N | {2,5} |

1 – Tabela de pool

* + - 1. **Criar tabelas de tags**: O sistema faz uma varredura nas tabelas de pool e, para cada tag, copia as informações relevantes do banco para uma linha na *Tabela de tags* local. A função dessa tabela está em promover o acesso rápido às informações de cada tag.

|  |  |  |
| --- | --- | --- |
| **Tabela de tags** | | |
| *Cabeçalho* | | |
| Modelo de prova | xxxx | |
| Id Instância | xxxx |  |
| *Registros* | | |
| Id Tag | Peso | Tipo |
| 1 | 1 | 1 |
| 2 | 1 | 2 |
| 3 | 1 | 3 |
| N | n | n |

2 – Tabela de tags

* + - 1. **Criar tabela de controle de sorteio**: A *Tabela de controle do sorteio* faz o gerenciamento do loop de sorteio a cada rodada. Essa tabela deve durar até a consolidação da nova instância de prova. Durante a preparação, o sistema preenche o ‘cabeçalho’ dessa tabela, definindo o candidato, o modelo de base para a instanciação, bem como o regime da prova (teste, validação ou produção, por um lado, e sorteio vs. composição manual, por outro).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela de controle de sorteio** | | | | | |
| *Cabeçalho* | | | | | |
| Modelo | | | 22 | | |
| Regime | | | Produção | | |
| Id Instância | | | Xxxx | | |
| Candidato | | | Xxxx | | |
| *Registros* | | | | | |
| **Slot** | **Nome** | **Stacks** | **Ocorrência** | **Lista de rodada** | **Conteúdo** |
| 0 | P1Q1 | {1,3} |  |  |  |
| 1 | P1Q2 | {1,3} |  |  |  |
| 2 | P1Q3 | {1,3} |  |  |  |
| 3 | P2Q1 | {1,2} |  |  |  |
| 4 | P2Q2 | {1,2} |  |  |  |
| 5 | P2Q3 | {1,2} |  |  |  |
| 6 | P2Q4 | {1,2} |  |  |  |
| 7 | P2Q5 | {1,2} |  |  |  |
| 8 | P3Q1 | {1,2} |  |  |  |
| 9 | P3Q2 | {1,2} |  |  |  |
| 10 | P3Q3 | {1,2} |  |  |  |
| 11 | P3Q4 | { } |  |  |  |
| 12 | P4Q1 | {1,4} |  |  |  |
| 13 | P4Q2 | {4} |  |  |  |
| 14 | P4Q3 | {4} |  |  |  |
| 15 | P4Q4 | {4} |  |  |  |

3 – Tabela de controle de sorteio

* + - 1. **Preencher id dos slots:** As primeiras colunas a serem preenchidas são a de slots e a de nome de cada questão – por exemplo, slot id '0', e nome 'parte 01, questão 01'. Cada linha da tabela corresponde a um slot daquele modelo sendo instanciado, e portanto, a uma questão que será sorteada. No modelo de prova atual (e no nosso exemplo) isso significa 16 slots, numerados de 0 a 15, cada um correspondente a uma questão de partes distintas, conforme consta na tabela abaixo.
      2. **Preencher coluna de stacks:** A segunda coluna a ser preenchida é a de stacks. O sistema cria para cada slot um conjunto com o id dos stacks a serem considerados naquela instância, que são os stacks apropriados para aquele modelo.
      3. **Criar tabelas de controle de stacks:** Além de preencher a coluna de stacks, o sistema deve criar, para cada stack presente no sorteio uma *Tabela de controle de stacks*. Essa tabela visa a auxiliar a aplicação das tags durante as rodadas. No caso do nosso sorteio de exemplo isso significa criar quatro tabelas de controle de stack, uma para Tema (1), para Sotaque(2), para Sintaxe(3) e para Bloco (4):

|  |  |  |
| --- | --- | --- |
| **Tabela de Stack** | | |
| *Cabeçalho* | | |
| Stack | 1 | |
| Tipo | Sotaque | |
| Regime | Gradual | |
| Onset | 1 | |
| Cumulativa | Sim | |
| Magnitude | 0,8 | |
| *Registros* | | |
| Tag | # Ocorr. | Peso |
| 1 | 1 | 1 |
| 2 | 0 | 1 |
| ... | ... |  |
| n | 0 | 1 |

4 – Exemplo de tabela de controle de stack

O sistema faz uma varredura nos registros de stack e compila os dados relevantes para o sorteio no cabeçalho: tipo, regime, onset, cumulativa, magnitude, etc. Além disso, varre os registros de tag, buscando todas as tags daquele *tipo* (no exemplo, todas as tags de sotaque), criando uma linha para cada uma delas. Nesses registros, anota também o número de ocorrências de cada tag no sorteio da instância atual (na primeira rodada, 0 para todas as tags, afinal, nenhuma questão foi sorteada ainda), bem como o Peso da tag (Pt), outro elemento importante no cálculo.

* + - 1. **Criar registro local do candidato**: O sistema busca as informações relevantes no banco, preenchendo a Tabela local de candidato:

|  |  |
| --- | --- |
| **Tabela local de candidato** | |
| Cabeçalho | |
| Id Candidato | Xxxx |
| Registros | |
| Id Questões interditadas | |
| 1 | |
| ... | |
| n | |

5 – Exemplo de Tabela local de candidato

* 1. Fase 2 – Sorteio

O Sorteio é uma rotina que supervisionará o preenchimento da *Tabela de controle de sorteio* a cada rodada. O output desse processo são as tabelas temporárias de *Lista interditada de candidato*, a *Tabela de controle de sorteio* preenchidas, bem como as *Tabelas de controle de tag*. Também são geradas nessa fase, para cada rodada, as *Tabelas de controle da rodada*. A fase de Sorteio é a mais complexa da geração de prova, e contém 5 etapas, etapas que também podem conter subprocessos. São essas etapas:

1. Gerar tabela de controle da rodada
   1. Compor lista preliminar
   2. Verificar questões interditadas
   3. Acrescentar questões à lista de sorteio
   4. Verificar se a Lista da ocorrência chegou ao fim
   5. Consolidar ocorrências
   6. Consolidar Lista de sorteio
2. Atribuir Quantidade básica de bilhetes a cada questão
3. Aplicar as tags
   1. Aplicar stack
      1. Verificar se a questão contêm tags aplicáveis na Tabela de controle de stack
      2. Calcular a nova Quantidade Final de Bilhetes (Qf), consolidando-a na Tabela de controle de rodada
      3. Verificar se há mais stacks aplicáveis – se sim, voltar para iii.a.i
4. Atribuir os intervalos
5. Gerar um número aleatório
6. Verificar a questão escolhida
7. Registar ocorrência, tipo 2
   1. Em havendo problema, retornar à Qbb
8. Consolidar sorteio

## Detalhamento do sorteio

* + - 1. **Realizar rodada:** para cada slot, o sistema realizará uma *rodada*. A rodada terá como output o preenchimento da coluna *Lista da rodada* naquele registro, um id de questão, que preencherá o campo 'conteúdo' daquele slot, um id de ocorrência, bem como uma *Tabela de controle de rodada*. Também a cada rodada, a Tabela de controle local do candidato receberá um novo registro de questão interditada.

Se tudo correr bem, o campo de ocorrências será preenchido com um 0. Eventualmente, porém, o sistema pode sinalizar algum tipo de problema no sorteio. Esse problema pode ocorrer durante porque o candidato já entrou em contato com todas as questões do pool (código 1) ou porque a combinação de tags foi restritiva demais e o sistema foi obrigado a sortear de forma totalmente randômica (código 2), sem considerar as tags.

A realização da rodada compreende nove passos:

* + - * 1. *Preencher a Lista de rodada*, *na tabela de controle de sorteio:* Primeiro, com base na Tabela de pool daquela rodada, o sistema gera uma Lista preliminar da rodada. Em seguida, busca na Tabela local do candidato as questões interditadas e as subtrai da Lista preliminar. O resultado é a Lista da rodada definitiva, que ele consolida no campo Lista de rodada. Essa será a lista do sorteio. Se por não houver questões para o sorteio, o sistema deve preencher o campo de ocorrência com o código 1.
        2. *Gerar a tabela de controle de rodada(i)*: Trata-se de um subprocesso, detalhado adiante, e que produz como output a *Tabela de controle da rodada*. Esta tabela tem uma linha para cada questão da Lista da rodada, e relaciona para cada uma delas a quantidade de bilhetes, o total de multiplicadores, e o intervalo do sorteio.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela de controle da rodada** | | | | | | | |
| *Cabeçalho* | | | | | | | |
| Id da instância | xxxx | Id slot | 2 | | Id Mod. | | 23 |
| Número natur. | 8989 | Id Questão | 99 | | Occ | | 0 |
| *Registros* | | | | | | | |
| Questão | Qbb | T | | Qbf | | Range | |
| 1 | 10000 | 1 | | 10000 | | 1-10000 | |
| 37 | 0,5 | | 5000 | | 10001-15000 | |
| ... | ... | | ... | | ... | |
| 178 | 0,5 | | 5000 | | 990001-100000 | |

1. – Tabela de controle de rodada

A geração dessa tabela consiste dos passos detalhados abaixo:

1. **Compor lista preliminar:** O sistema acessa a Lista da rodada e preenche um registro para cada id de questão.
2. **Atribuir a cada questão a Quantidade Básica de Bilhetes(ii)**: Para cada questão da Lista da Rodada, o sistema atribui uma quantidade padrão de bilhetes, normalmente 100, que depois poderá aumentar ou diminuir a fim de calibrar o sorteio.
3. **Aplicar as tags**: Para cada questão, o sistema aplica tags a fim de aumentar ou diminuir as chances de uma questão específica ser escolhida, chegando à Quantidade de Bilhetes Final (Qbf). A chance efetiva de cada questão ser sorteada será a quantidade final de bilhetes dela dividido pelo total de 'bilhetes em circulação'. As fórmulas para aplicação de tags encontram-se explicadas no item 4. A aplicação de tags se compõe dos seguintes passos:
   * 1. Verificar na Tabela de controle de sorteio se o slot está vinculado a um stack. Caso não, terminar a aplicação de tag.
     2. Fazer uma varredura em todas as linhas da Tabela de controle da rodada. Para cada linha:
        1. Buscar o id da questão na Tabela de pool para aquele slot, verificando se aquela questão tem tags;
        2. Se a questão não tiver tiver tags, ir para a linha seguinte;
        3. Se a questão tiver tags, verificar as informações relevantes para a aplicação na Tabela de tag e na Tabela de stack correspondente;
        4. Aplicar a tag;
        5. Registrar a Quantidade final de bilhetes para aquela questão.
        6. Passar para a linha seguinte.
4. **Atribuir intervalos(iv):** Com as quantidades finais distribuídas, o sistema pode enfim atribuir intervalos para cada questão, e preencher, na Tabela de controle de rodada, a coluna correspondente. O intervalo é o conjunto de números efetivamente atribuído àquela questão, por exemplo, a questão 1 pode ficar com os bilhetes de 1 a 100, a questão 2 com os números de 101 a 150, e assim por diante. O último número do último intervalo será o *Limite do sorteio*.
5. **Gerar um número aleatório(v):**O sistema número aleatório natural entre 1 e o *Limite do sorteio*. Esse número será equivalente ao 'sorteio da loteria federal', quando se procede ao sorteio de rifas, e determinará a questão escolhida.
6. **Verificar questão escolhida(vi):** O sistema verifica a qual questão pertence o número (‘bilhete’) e consolida o número no topo dos registros da Tabela de controle da rodada. Por exemplo, se o número for igual a 25, a questão número 1 (intervalo 1 a 100) será a sorteada.
7. **Registrar ocorrência, tipo 2:** Caso haja problema no sorteio de uma questão – por exemplo, as tags produziram resultados inconsistentes, o sistema deve registrar uma ocorrência, do tipo 2.
8. **Atribuir Qbb (opcional):** Em havendo problema, o sistema repete a atribuição básica dos bilhetes, e realiza novo sorteio, pulando a etapa de tags.
9. **Consolidar o sorteio:** O sorteio da questão se consolida com o registro das informações no cabeçalho.

Esses dois tipos de tabelas, a de controle de sorteio e as de controle de stack, realizam a implementação da memória macro do sorteio.

* + - * 1. *Atualizar Tabela de controle de sorteio*: Nessa etapa, o sistema consolida na Tabela de controle de sorteio o id da questão escolhida, o tipo de ocorrência de erro (0 = sem erro, 1 = erro de questões indisponíveis; ou 2 = erro de tag).
        2. *Atualizar tabelas de controle de stacks:* Conforme a questão escolhida para o slot, o sistema atualiza as Tabelas de controle de stack correspondentes, acrescendo em 1 o número de ocorrências no registro de cada tag que a questão sorteada possuir. Por exemplo, se a questão possuir a tag 'Americano', do tipo 'sotaque', o sistema vai na tabela do stack correspondente e aumenta o número de ocorrências da tag 'Americano' em um.
        3. *Atualizar Tabela local do candidato:* Acrescenta a questão escolhida à Lista de interditadas do candidato nos registros locais. Se ela já for repetida (código ocorrência 1), a questão não é acrescentada.
        4. *Verificar fim*: Se os slots chegaram ao fim, passa para a fase de consolidação da instância de prova. Caso contrário, o ciclo se repete.

# Fase 3 – Consolidação da instância de prova

Uma vez concluído o sorteio, o sistema deve registrar no banco uma nova instância de prova, bem como as informações da Tabela de controle de sorteio preenchida. Esta é a 'receita' para a produção da nova versão de prova, que será utilizada posteriormente numa aplicação online, ou quando o Software Local de Aplicação requisitar o download criptografado da instância.

# Formulas de aplicação de tag

Durante a aplicação da tag, o sistema faz uma varredura na tabela de rodada

Para Para cada questão:

01 ) Verifica se ela está marcada com a tag

02 ) Calcula a nova quantidade de bilhetes, conforme o regime.

Há dois regimes de cálculo para o stack: o binário e o gradual.

**Stack no regime binário**

O regime binário serve para operações extremas com tags. Se a tag for binária do tipo ‘Não’, uma vez que uma questão contendo aquela tag for escolhida, todas as outras são excluídas nas rodadas seguintes do stack. Se a tag for do tipo ‘Sim’ acontece o oposto: todas as outras, as que não estiverem marcadas com a tag, são excluídas.

Formalizando em termos de lógica, o cálculo com o stack binário uma fórmula para cada cenário (sim ou não, M = 1 ou M = 0) e duas variáveis de controle: 1) o Limiar (L) e 2) a Magnitude (M). Como já foi dito, a M controla se a tag é do tipo Sim (1) ou Não (0). A variável de limiar conta quantas vezes o ciclo de variáveis foi completado, isto é, cada vez que todas a tags são utilizadas, ela a acrescida em 1.

A fórmula para o controle de tag do stack binário é:

*Se 1) a magnitude do stack (M) for igual a ‘Sim’ (1) e; 2) o número de ocorrências daquela tag específica for maior que o limiar e 3) Pt não for igual a zero, então todas as questões marcadas com a tag T (pertencentes ao conjunto) manterão seus bilhetes, enquanto as outras, que não pertencem ao conjunto, terão seus bilhetes reduzidos a zero.*

*Se 1) a magnitude do stack(M) for igual a ‘Não’(0) e; 2) o número de ocorrências daquela tag específica for maior que o limiar, e 3) Pt não for igual a zero, o oposto acontece: todas as questões marcadas com a tag T (pertencentes ao conjunto) terão seus bilhetes reduzidos a zero, enquanto as outras mantêm seus bilhetes.*

*Se Pt for igual a zero, aquela tag deve ser ignorada*.

Em linguagem de conjuntos:

sendo

T = Conjunto de questões marcadas com a tag

Qbb = Quantidade básica de bilhetes

Qbf = Quantidade de bilhetes final

q = questão

M = Magnitude na rodada, variável do tipo sim(1)/não(0)

L = Limiar para aplicação da tag

As fórmulas acima podem ser traduzidas como:

Segue um exemplo concreto, que também detalha a função do limiar. Vamos supor que haja 4 tags do tipo ‘Não’, A, B, C e D num sorteio. A ordem do sorteio é C, B, D A. Se o sistema não tivesse um controle de limiar, a partir da quarta rodada chegaria numa situação em que todas as tags teriam sido aplicadas, e não conseguiria sortear mais questões válidas (todas são não). Entretanto, a variável L é acrescida de 1 no final da quarta rodada. Isso significa que as tags só serão aplicadas quando tiverem tido duas ocorrências. Na quinta rodada, isso significa que não haverá aplicação de nenhuma. O sorteio escolheu A. Na sexta, só ela será considerada.

**Stack no regime gradual**

O stack gradual se aplica para qualquer questão nesse regime. A fórmula de aplicação de tag no contexto de um stack é:

Onde,

Qb = a quantidade de bilhetes a ser alocada para uma questão específica

Qbb = quantidade básica de bilhetes a ser alocada para cada questão

M = Magnitude

Pt = Peso da tag

n = Número de ocorrências da tag no stack

c = Se a tag for cumulativa, c é 1; caso contrário, c é 0

**Exemplo**:

Para realizar os cálculos de tag expostos, o sistema usa as tabelas de controle de stack, a tabela de controle de rodada e uma variável de Limiar, no caso do regime binário. São esses dados que indicam os parâmetros da aplicação. Vamos supor que o sistema vá aplicar o stack 1 durante o sorteio do slot 3 de uma prova qualquer. O stack faz referência a uma tag do tipo tema, intensidade -0,8, tipo gradual. Essa é a tabela de controle do stack:

|  |  |  |
| --- | --- | --- |
| Stack 1, -0,8 | Tema |  |
| Tag | Quantidade | Peso da Tag |
| Derrapagem | 1 | 1 |
| Terrorismo | 0 | 1 |
| Neblina | 2 | 1 |

Como o regime é gradual, não existe variável de limiar – as tags podem ser aplicadas sempre.

Exemplos:

Questões marcadas com Derrapagem:

Questões marcadas com Terrorismo:

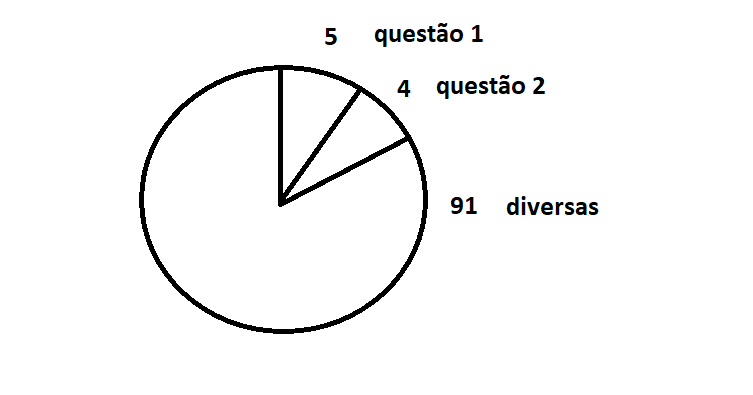
Questões marcadas com Neblina:

Script

# Estudo de caso

Quando usar uma tag desativada?

Suponhamos que a gente tenha um pool com 100 questões, compartilhadas por 3 slots, divididas conforme esquema abaixo:



A questão 1 aparece no pool repetida 5 vezes, gravadas em cinco sotaques diferentes. A questão 2 aparece 4 vezes, gravadas em 4 sotaques diferentes. As outras são questões diversas, que não se repetem. Como dizer para o sistema que ele deve não repetir nenhuma questão 1, ou nenhuma questão 2 se elas forem sorteadas para algum slot?

A gente cria um stack com uma tag do tipo 'Questão individual' para realizar esse controle. Então, cria uma tag do tipo binária diferente chamada 'questão 1' e marca as cinco primeiras questões. Se alguma delas for sorteada, não irá mais se repetir. Faz o mesmo com a questão 2, cria uma tag do tipo binária diferente chamada 'questão 2' e a aplica às questões de 6 a 9. E quanto às questões restantes? Devido às regras do stack, somos obrigados a marcar todas as questões do pool com uma tag desse tipo, então seríamos obrigados a criar uma tag 'questão individual' para cada questão do pool, não fosse a possibilidade de desativar a tag. Essa variável da tag nos dá um caminho diferente, e muito mais prático.

Basta que a gente crie uma única tag – 'questões diversas', e ajuste seu 'Peso' para 0. Isso faz com que a questão cumpra os requisitos de stack e pool – todas elas serão marcadas com a tag do tipo 'questão individual', e ao mesmo tempo, podemos marcar todas as outras 91 com a certeza de que se uma delas for sorteada, a tag não será aplicada.

Quando devo usar uma tag não-cumulativa? Ou com onset 0?

Suponhamos que um banco contenha 100 questões antigas e 100 questões novas. Devido ao estudo estatístico realizado, chega-se à conclusão de que é seguro que até 20% das questões (em média) possa ser utilizado sem que a segurança do teste seja afetada. Nesse caso, marcamos as questões antigas com a tag 'antiga', que deve ser não-cumulativa e onset 0, com intensidade -0,8.

Onset 0 significa que desde a primeira rodada será aplicada. Se o onset fosse igual a 1, as questões antigas nunca seriam escolhidas para o primeiro slot. O fato de ser não-cumulativa significa que a intensidade será constante ao longo da prova, não importa quantas questões tenham sido sorteadas numa mesma instância. Assim, para cada slot, existe 20% de chance de uma questão antiga ser escolhida. Se ela fosse cumulativa, essa chance cairia para 4% tão logo uma questão antiga fosse escolhida, e para 0,8% assim que duas fossem.

Glossário

Slot – Unidade no modelo de prova que serve de placeholder indicando um espaço a ser preenchido por uma questão sorteada. Cada modelo de prova contém um número de slots, que também se articulam com o stack.

Stack – Controle de eventos que se estende por um número de slots.

Tag – Mecanismo unificado para calibrar o sorteio randômico da prova. A tag tem os seguintes parâmetros