



## 1. Sorteio dos Inscritos

Todo o processo será efetuado utilizando o Python na versão 3 e o pacote Numpy versão 1.19.4, responsável por receber a semente de sorteio (*seed*) e executar a função *randint*, a qual escolherá, com base na *seed*, os números sorteados para cada candidato.

Para garantir que o sorteio seja determinístico, ou seja, sempre dê o mesmo resultado, será definida e utilizada a *seed* e a lista de inscrições ordenada, informadas nas sessões posteriores, pois ambas juntas procuram garantir que, encerrado as inscrições de um edital de processo seletivo, sempre se obtenha o mesmo resultado de sorteio.

Serão sorteados números entre 1 e  $(2^{32} - 1)$  para cada candidato inscrito no processo seletivo, esses números serão atribuídos a cada candidato na ordem da lista ordenada. Caso, ao ser sorteado algum número a um candidato e esse número já tenha sido selecionado anteriormente, um novo número será sorteado até se conseguir um que ainda não tenha sido utilizado, para então prosseguir atribuindo os números sorteados aos próximos candidatos.

Com os números sorteados gerados pelo sistema e atribuídos a cada candidato, o sistema ordenará cada registro por ordem decrescente desse número sorteado. Após a ordenação do resultado do sorteio cada candidato será agrupado em seu respectivo campus, curso e cota para a geração da lista de classificados. Essa lista é descrita conforme dicionário de dados a seguir:

Coluna	Descriptor	Valor de Exemplo
campus	Nome do Campus	CAMPUS ARACAJU
curso	Nome do Curso	Técnico de Nível Médio em Alimentos
cota	Descrição da Cota ofertada	Grupo A - Ampla Concorrência
numero_inscricao	Número de inscrição gerado pelo SIGAA após inscrição de um candidato em uma vaga do processo seletivo	25086
nome	Nome do candidato	FULANO DE TAL
numero_sorteado	Número sorteado para o candidato	419326372

## 2. Construção da Seed



A *seed* do processo seletivo será gerada após o encerramento das inscrições do processo seletivo, onde ela será gerada a partir de uma fórmula que leva em consideração o número de candidatos inscritos, os cursos cadastrados e as ofertas de cotas para os cursos.

Para construção da possível seed e da seed definitiva, é gerado a listagem de todos os cursos, conforme dicionário de dados a seguir:

Coluna	Descriptor	Valor de Exemplo
campus	Nome do Campus	CAMPUS ARACAJU
curso	Nome do Curso	Técnico de Nível Médio em Alimentos
cota	Descrição da Cota ofertada	Grupo A - Ampla Concorrência
quantidadelnscricoes	Total de Inscrições para a cota de um curso de um determinado campus	394
blend	Índice da combinação entre as colunas, campus, curso e cota ofertada	1
acumulador	Valor resultado da fórmula do acumulador	12574116

Os acumuladores são gerados a partir da soma entre número 8 e o índice da combinação do campus, curso e cota, sendo o resultado desta soma multiplicado pelo número de inscritos neste campus, curso e cota, sendo que o produto dessa operação é elevado ao quadrado.

Após a construção do *acumulador*, é realizada a soma de todos os acumuladores na qual retorna um valor, conhecido como: possível *seed* (*possível Seed*). O valor da *possível Seed* pode chegar a ser superior a  $2^{32} - 1$ , o que acarretará um erro na função *seed* em sistemas de 32 bits.

Para evitar que seja disparado o erro em sistemas 32 bits, o valor da possível *Seed* é dividido por  $2^{32}$  e o resto dessa divisão será a *seed* utilizada no processo seletivo.

### 3. Ordenação da Lista de Inscrições

Neste passo, é extraída uma lista com todos os inscritos no Processo Seletivo (edital). Essa lista é ordenada pelo número de inscrição de forma ascendente, sendo descrita conforme dicionário de dados a seguir:



Coluna	Descriptor	Valor de Exemplo
campus	Nome do Campus	CAMPUS ARACAJU
curso	Nome do Curso	Técnico de Nível Médio em Alimentos
cota	Descrição da Cota ofertada	Grupo A - Ampla Concorrência
numero_inscricao	Número de inscrição gerado pelo SIGAA após inscrição de um candidato em uma vaga do processo seletivo	25086
nome	Nome do candidato	FULANO DE TAL

Diretoria de Tecnologia da Informação IFS