5\_Mensuracao\_de\_Duracoes.cpp - (~/Documentos/atual\_2023-1/lp/aulas/09\_Aleatorio\_e\_Tempo/execucao) - GVIM <u>A</u>rquivo <u>E</u>ditar <u>F</u>erramentas <u>S</u>intaxe <u>B</u>uffers <u>J</u>anela Ajuda #include <cstdlib> #include <random> #include <random> #include <random> #include <iostream> #include <iostream> #include <iostream> #include <iostream> using namespace std; using namespace std; using namespace std; using namespace std; int main () int muito errado () class MeuGerador int main () srand(time(nullptr)); mt19937 gerador; // Demorado e sempre começa igual! mt19937 gerador; random device disp aleat; uniform int distribution<int> distr; cout << "RAND MAX: " << RAND MAX << '\n'; mt19937 gerador{ disp aleat() }; // Semente Aleatória return gerador(); public: for (int i = 0; i < 5; ++i) uniform int distribution distr{0,10}; int melhor (mt19937 &gerador) MeuGerador () : gerador{}, distr{100,200} { } for (int i = 0; i < 10; ++i) cout << rand() << '\n'; return gerador(); // Ok, mas podia chamar logo diretamente. int operator() () // int numero () cout << ' ' << distr(gerador);</pre> for (int i = 0; i < 5; ++i) return distr(gerador); lass MeuGerador cout << rand() % 100 << '\n'; cout << '\n'; mt19937 gerador; int main () for (int i = 0; i < 5; ++i) public: mt19937 gerador; 4 Sementes.cpp 8,1 #include <chrono> cout << rand() / (double) RAND MAX << '\n';</pre> MeuGerador () /\* : gerador{} \*/ { } uniform int distribution distr(100,200); #include <iostream> int numero () // Não precisa de argumento. using namespace std; using namespace chrono; for (int i = 0; i < 10; ++i) return gerador(); cout << ' ' << distr(gerador); **}**; int main () cout << "\n\n"; int main () int n; cout << "n: "; uniform real distribution dr(5.0,10.0); auto inicio = steady clock::now(); mt19937 gerador; for (int i = 0; i < 10; ++i) for (int i = 0; i < 10; ++i) cout << ' ' << gerador(); cin >> n; cout << "\n\n"; cout << ' ' << dr(gerador); auto fim = steady clock::now(); for (int i = 0; i < 10; ++i) cout << ' ' << muito errado();</pre> cout << "\n\n"; duration<double> d = fim - inicio; cout << "\n\n"; MeuGerador g; for (int i = 0; i < 10; ++i) cout << ' ' << melhor(gerador);</pre> cout << "Você demorou " << d.count()</pre> for (int i = 0; i < 10; ++i) << " segundos para digitar.\n"; cout << "\n\n"; cout << ' ' << g(); // g.numero(); duration<double, ratio<60>> min = d; MeuGerador meu; for (int i = 0; i < 10; ++i) cout << ' ' << meu.numero(); cout << "Você demorou " << min.count()</pre> cout << "\n\n"; cout << "\n\n"; << " minutos para digitar.\n";</pre> cout << "Observação: min\*60 = " << min.count() \* 60 << '\n';

34,1

Tudo 3 Distribuicoes Uniformes.cpp

25,1

Tudo 5 Mensuracao de Duracoes.cpp

11,1

1 Aleatorios via C.cpp

:set noma

14,1

Tudo 2 Gerador MT.cpp