



Instituto Politécnico Nacional

INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE COMPUTO

PRÁCTICA 2: NÚMEROS PRIMOS

Teoría de la Computación

Autor:
Hernández Vergara Eduardo

Marzo 2022

Índice

1. Introducción	2
1.1. Objetivo	2
1.2. Números Primos	2
1.3. ¿Por qué usar la definición?	2
2. Solución	3
2.1. Escritura de Datos	3
2.2. Lectura de Datos	3
3. Código	7
3.1. Escritura de Datos	7
3.2. Grafica/Lectura Datos	9

1. Introducción

1.1. Objetivo

En la práctica desarrollada tuvimos que desarrollar un algoritmo que nos de n números primos, por ello, se usó la definición de los números primos para realizar la práctica

1.2. Números Primos

Los números primos son todos los números naturales mayores a 1 que tienen exactamente dos factores: la unidad y ellos mismos.

El término primo deriva del latín "primus" que significa primero (protos en griego). Todo número entero se expresa de forma única como producto de números primos. Por eso se les considera los "primeros", porque a partir de ellos se obtienen todos los demás números enteros.

El teorema fundamental de la Aritmética establece que cualquier número natural mayor que 1 siempre puede representarse como un producto de números primos, y esta representación (factorización) es única en el orden de los factores, razón por la cual por consenso los matemáticos no consideran al 1 como primo.

1.3. ¿Por qué usar la definición?

En el programa desarrollado usamos la definición de número primo porque no hay una manera, o por lo menos no fui capaz de verla/encontrarla, ya que, si hay maneras de encontrar un número primo de manera lineal, pero hay varios inconvenientes, si lo ves desde el punto de vista binario todos los números que tienen todos los bits encendidos, digase (11, 111, 1111, 11111), etc; son números primos, pero ahí el problema es que ahí no están contemplados los demás números primos, por lo que podemos decir que no es viable, otra manera de encontrar números primos es: para todo número primo mayor a 5 se puede expresar como, $n+1$ y $n-1$, el problema de esto es que además de devolverte números primos te devuelve algunos complejos, por lo que no es viable.

2. Solución

Para la implementación de la solución se uso C#, con .Net 6.0(para las gráficas se uso .Net 4.8), con el IDE Visual Studio 2022, como veremos a continuación con las capturas de resultado, veremos como salio.

2.1. Escritura de Datos

```
>>Menu<<
Eliga una opcion
1. Escoger Numero:
2. Al azar:
3.Salir.
1
Ingresa k:
400000
Se tardo en escribir: 01:19:21.485
Exito
```

Figura 1: Escritura de datos

En esta escritura usare un $n = 100\,000$, porque por error cerre el que me desplegaba el $n = 1M$ y tardo 8 horas, asi que no quiero que se tarde más. En esta imagen pongo 4 archivos, los





 primos.txt	13/03/2022 06:38 a. m.	Documento de te...	8,053 KB
 primosBinario.txt	13/03/2022 06:38 a. m.	Documento de te...	23,252 KB
 primosBinarioConjunto.txt	13/03/2022 06:38 a. m.	Documento de te...	24,228 KB
 primosConjunto.txt	13/03/2022 06:38 a. m.	Documento de te...	9,030 KB

Figura 2: El tamaño del archivo

que dicen Conjunto son los abro, los otros estan de manera que me facilitan la lectura

2.2. Lectura de Datos

Ahora bien, mi programa por si solo, si llega a hacer millones de numeros primos, pero no graficarlos, para $n = 1M$ me mando un error de memoria, siendo este el siguiente, use para las graficas un $n = 100\,000$ por la falta de tiempo, la escritura de esta grafica es colosal, entonces incluso con ese numero que me tarda en escribir 5 minutos, se tardo 30 minutos en graficarla.

Los primos son : {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 397, 401, 409, 419, 421, 431, 439, 443, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541, 547, 551, 557, 563, 569, 577, 581, 587, 593, 599, 607, 613, 617, 619, 623, 629, 631, 637, 641, 643, 647, 653, 659, 661, 667, 671, 673, 677, 683, 687, 691, 697, 701, 703, 707, 709, 713, 719, 727, 729, 733, 737, 743, 751, 757, 761, 763, 767, 773, 779, 781, 787, 791, 793, 797, 803, 809, 811, 817, 821, 823, 827, 833, 839, 847, 851, 853, 857, 859, 863, 867, 871, 877, 881, 883, 887, 893, 897, 901, 907, 911, 913, 917, 919, 923, 929, 931, 937, 941, 943, 947, 953, 959, 961, 967, 971, 973, 977, 983, 989, 991, 993, 997, 1001, 1003, 1007, 1009, 1013, 1017, 1021, 1031, 1033, 1039, 1041, 1043, 1047, 1051, 1053, 1057, 1061, 1063, 1067, 1069, 1073, 1077, 1081, 1083, 1087, 1091, 1093, 1097, 1101, 1103, 1107, 1109, 1111, 1113, 1117, 1121, 1123, 1127, 1129, 1133, 1137, 1141, 1143, 1147, 1151, 1153, 1157, 1161, 1163, 1167, 1171, 1173, 1177, 1181, 1183, 1187, 1191, 1193, 1197, 1201, 1203, 1207, 1211, 1213, 1217, 1219, 1223, 1227, 1229, 1231, 1233, 1237, 1241, 1243, 1247, 1251, 1253, 1257, 1259, 1261, 1263, 1267, 1271, 1273, 1277, 1279, 1281, 1283, 1287, 1291, 1293, 1297, 1301, 1303, 1307, 1311, 1313, 1317, 1321, 1323, 1327, 1331, 1333, 1337, 1341, 1343, 1347, 1351, 1353, 1357, 1361, 1363, 1367, 1371, 1373, 1377, 1381, 1383, 1387, 1391, 1393, 1397, 1401, 1403, 1407, 1411, 1413, 1417, 1421, 1423, 1427, 1431, 1433, 1437, 1441, 1443, 1447, 1451, 1453, 1457, 1461, 1463, 1467, 1471, 1473, 1477, 1481, 1483, 1487, 1491, 1493, 1497, 1501, 1503, 1507, 1511, 1513, 1517, 1521, 1523, 1527, 1531, 1533, 1537, 1541, 1543, 1547, 1551, 1553, 1557, 1561, 1563, 1567, 1571, 1573, 1577, 1581, 1583, 1587, 1591, 1593, 1597, 1601, 1603, 1607, 1611, 1613, 1617, 1621, 1623, 1627, 1631, 1633, 1637, 1641, 1643, 1647, 1651, 1653, 1657, 1661, 1663, 1667, 1671, 1673, 1677, 1681, 1683, 1687, 1691, 1693, 1697, 1701, 1703, 1707, 1711, 1713, 1717, 1721, 1723, 1727, 1731, 1733, 1737, 1741, 1743, 1747, 1751, 1753, 1757, 1761, 1763, 1767, 1771, 1773, 1777, 1781, 1783, 1787, 1791, 1793, 1797, 1801, 1803, 1807, 1811, 1813, 1817, 1821, 1823, 1827, 1831, 1833, 1837, 1841, 1843, 1847, 1851, 1853, 1857, 1861, 1863, 1867, 1871, 1873, 1877, 1881, 1883, 1887, 1891, 1893, 1897, 1901, 1903, 1907, 1911, 1913, 1917, 1921, 1923, 1927, 1931, 1933, 1937, 1941, 1943, 1947, 1951, 1953, 1957, 1961, 1963, 1967, 1971, 1973, 1977, 1981, 1983, 1987, 1991, 1993, 1997, 2001, 2003, 2007, 2011, 2013, 2017, 2021, 2023, 2027, 2031, 2033, 2037, 2041, 2043, 2047, 2051, 2053, 2057, 2061, 2063, 2067, 2071, 2073, 2077, 2081, 2083, 2087, 2091, 2093, 2097, 2101, 2103, 2107, 2111, 2113, 2117, 2121, 2123, 2127, 2131, 2133, 2137, 2141, 2143, 2147, 2151, 2153, 2157, 2161, 2163, 2167, 2171, 2173, 2177, 2181, 2183, 2187, 2191, 2193, 2197, 2201, 2203, 2207, 2211, 2213, 2217, 2221, 2223, 2227, 2231, 2233, 2237, 2241, 2243, 2247, 2251, 2253, 2257, 2261, 2263, 2267, 2271, 2273, 2277, 2281, 2283, 2287, 2291, 2293, 2297, 2301, 2303, 2307, 2311, 2313, 2317, 2321, 2323, 2327, 2331, 2333, 2337, 2341, 2343, 2347, 2351, 2353, 2357, 2361, 2363, 2367, 2371, 2373, 2377, 2381, 2383, 2387, 2391, 2393, 2397, 2401, 2403, 2407, 2411, 2413, 2417, 2421, 2423, 2427, 2431, 2433, 2437, 2441, 2443, 2447, 2451, 2453, 2457, 2461, 2463, 2467, 2471, 2473, 2477, 2481, 2483, 2487, 2491, 2493, 2497, 2501, 2503, 2507, 2511, 2513, 2517, 2521, 2523, 2527, 2531, 2533, 2537, 2541, 2543, 2547, 2551, 2553, 2557, 2561, 2563, 2567, 2571, 2573, 2577, 2581, 2583, 2587, 2591, 2593, 2597, 2601, 2603, 2607, 2611, 2613, 2617, 2621, 2623, 2627, 2631, 2633, 2637, 2641, 2643, 2647, 2651, 2653, 2657, 2661, 2663, 2667, 2671, 2673, 2677, 2681, 2683, 2687, 2691, 2693, 2697, 2701, 2703, 2707, 2711, 2713, 2717, 2721, 2723, 2727, 2731, 2733, 2737, 2741, 2743, 2747, 2751, 2753, 2757, 2761, 2763, 2767, 2771, 2773, 2777, 2781, 2783, 2787, 2791, 2793, 2797, 2801, 2803, 2807, 2811, 2813, 2817, 2821, 2823, 2827, 2831, 2833, 2837, 2841, 2843, 2847, 2851, 2853, 2857, 2861, 2863, 2867, 2871, 2873, 2877, 2881, 2883, 2887, 2891, 2893, 2897, 2901, 2903, 2907, 2911, 2913, 2917, 2921, 2923, 2927, 2931, 2933, 2937, 2941, 2943, 2947, 2951, 2953, 2957, 2961, 2963, 2967, 2971, 2973, 2977, 2981, 2983, 2987, 2991, 2993, 2997, 3001, 3003, 3007, 3011, 3013, 3017, 3021, 3023, 3027, 3031, 3033, 3037, 3041, 3043, 3047, 3051, 3053, 3057, 3061, 3063, 3067, 3071, 3073, 3077, 3081, 3083, 3087, 3091, 3093, 3097, 3101, 3103, 3107, 3111, 3113, 3117, 3121, 3123, 3127, 3131, 3133, 3137, 3141, 3143, 3147, 3151, 3153, 3157, 3161, 3163, 3167, 3171, 3173, 3177, 3181, 3183, 3187, 3191, 3193, 3197, 3201, 3203, 3207, 3211, 3213, 3217, 3221, 3223, 3227, 3231, 3233, 3237, 3241, 3243, 3247, 3251, 3253, 3257, 3261, 3263, 3267, 3271, 3273, 3277, 3281, 3283, 3287, 3291, 3293, 3297, 3301, 3303, 3307, 3311, 3313, 3317, 3321, 3323, 3327, 3331, 3333, 3337, 3341, 3343, 3347, 3351, 3353, 3357, 3361, 3363, 3367, 3371, 3373, 3377, 3381, 3383, 3387, 3391, 3393, 3397, 3401, 3403, 3407, 3411, 3413, 3417, 3421, 3423, 3427, 3431, 3433, 3437, 3441, 3443, 3447, 3451, 3453, 3457, 3461, 3463, 3467, 3471, 3473, 3477, 3481, 3483, 3487, 3491, 3493, 3497, 3501, 3503, 3507, 3511, 3513, 3517, 3521, 3523, 3527, 3531, 3533, 3537, 3541, 3543, 3547, 3551, 3553, 3557, 3561, 3563, 3567, 3571, 3573, 3577, 3581, 3583, 3587, 3591, 3593, 3597, 3601, 3603, 3607, 3611, 3613, 3617, 3621, 3623, 3627, 3631, 3633, 3637, 3641, 3643, 3647, 3651, 3653, 3657, 3661, 3663, 3667, 3671, 3673, 3677, 3681, 3683, 3687, 3691, 3693, 3697, 3701, 3703, 3707, 3711, 3713, 3717, 3721, 3723, 3727, 3731, 3733, 3737, 3741, 3743, 3747, 3751, 3753, 3757, 3761, 3763, 3767, 3771, 3773, 3777, 3781, 3783, 3787, 3791, 3793, 3797, 3801, 3803, 3807, 3811, 3813, 3817, 3821, 3823, 3827, 3831, 3833, 3837, 3841, 3843, 3847, 3851, 3853, 3857, 3861, 3863, 3867, 3871, 3873, 3877, 3881, 3883, 3887, 3891, 3893, 3897, 3901, 3903, 3907, 3911, 3913, 3917, 3921, 3923, 3927, 3931, 3933, 3937, 3941, 3943, 3947, 3951, 3953, 3957, 3961, 3963, 3967, 3971, 3973, 3977, 3981, 3983, 3987, 3991, 3993, 3997, 4001, 4003, 4007, 4011, 4013, 4017, 4021, 4023, 4027, 4031, 4033, 4037, 4041, 4043, 4047, 4051, 4053, 4057, 4061, 4063, 4067, 4071, 4073, 4077, 4081, 4083, 4087, 4091, 4093, 4097, 4101, 4103, 4107, 4111, 4113, 4117, 4121, 4123, 4127, 4131, 4133, 4137, 4141, 4143, 4147, 4151, 4153, 4157, 4161, 4163, 4167, 4171, 4173, 4177, 4181, 4183, 4187, 4191, 4193, 4197, 4201, 4203, 4207, 4211, 4213, 4217, 4221, 4223, 4227, 4231, 4233, 4237, 4241, 4243, 4247, 4251, 4253, 4257, 4261, 4263, 4267, 4271, 4273, 4277, 4281, 4283, 4287, 4291, 4293, 4297, 4301, 4303, 4307, 4311, 4313, 4317, 4321, 4323, 4327, 4331, 4333, 4337, 4341, 4343, 4347, 4351, 4353, 4357, 4361, 4363, 4367, 4371, 4373, 4377, 4381, 4383, 4387, 4391, 4393, 4397, 4401, 4403, 4407, 4411, 4413, 4417, 4421, 4423, 4427, 4431, 4433, 4437, 4441, 4443, 4447, 4451, 4453, 4457, 4461, 4463, 4467, 4471, 4473, 4477, 4481, 4483, 4487, 4491, 4493, 4497, 4501, 4503, 4507, 4511, 4513, 4517, 4521, 4523, 4527, 4531, 4533, 4537, 4541, 4543, 4547, 4551, 4553, 4557, 4561, 4563, 4567, 4571, 4573, 4577, 4581, 4583, 4587, 4591, 4593, 4597, 4601, 4603, 4607, 4611, 4613, 4617, 4621, 4623, 4627, 4631, 4633, 4637, 4641, 4643, 4647, 4651, 4653, 4657, 4661, 4663, 4667, 4671, 4673, 4677, 4681, 4683, 4687, 4691, 4693, 4697, 4701, 4703, 4707, 4711, 4713, 4717, 4721, 4723, 4727, 4731, 4733, 4737, 4741, 4743, 4747, 4751, 4753, 4757, 4761, 4763, 4767, 4771, 4773, 4777, 4781, 4783, 4787, 4791, 4793, 4797, 4801, 4803, 4807, 4811, 4813, 4817, 4821, 4823, 4827, 4831, 4833, 4837, 4841, 4843, 4847, 4851, 4853, 4857, 4861, 4863, 4867, 4871, 4873, 4877, 4881, 4883, 4887, 4891, 4893, 4897, 4901, 4903, 4907, 4911, 4913, 4917, 4921, 4923, 4927, 4931, 4933, 4937, 4941, 4943, 4947, 4951, 4953, 4957, 4961, 4963, 4967, 4971, 4973, 4977, 4981, 4983, 4987, 4991, 4993, 4997, 5001, 5003, 5007, 5011, 5013, 5017, 5021, 5023, 5027, 5031, 5033, 5037, 5041, 5043, 5047, 5051, 5053, 5057, 5061, 5063, 5067, 5071, 5073, 5077, 5081, 5083, 5087, 5091, 5093, 5097, 5101, 5103, 5107, 5111, 5113, 5117, 5121, 5123, 5127, 5131, 5133, 5137, 5141, 5143, 5147, 5151, 5153, 5157, 5161, 5163, 5167, 5171, 5173, 5177, 5181, 5183, 5187, 5191, 5193, 5197, 5201, 5203, 5207, 5211, 5213, 5217, 5221, 5223, 5227, 5231, 5233, 5237, 5241, 5243, 5247, 5251, 5253, 5257, 5261, 5263, 5267, 5271, 5273, 5277, 5281, 5283, 5287, 5291, 5293, 5297, 5301, 5303, 5307, 5311, 5313, 5317, 5321, 5323, 5327, 5331, 5333, 5337, 5341, 5343, 5347, 5351, 5353, 5357, 5361, 5363, 5367, 5371, 5373, 5377, 5381, 5383, 5387, 5391, 5393, 5397, 5401, 5403, 5407, 5411, 5413, 5417, 5421, 5423, 5427, 5431, 5433, 5437, 5441, 5443, 5447, 5451, 5453, 5457, 5461, 5463, 5467, 5471, 5473, 5477, 5481, 5483, 5487, 5491, 5493, 5497, 5501, 5503, 5507, 5511, 5513, 5517, 5521, 5523, 5527, 5531, 5533, 5537, 5541, 5543, 5547, 5551, 5553, 5557, 5561, 5563, 5567, 5571, 5573, 5577, 5581, 5583, 5587, 5591, 5593, 5597, 5601, 5603, 5607, 5611, 5613, 5617, 5621, 5623, 5627, 5631, 5633, 5637, 5641, 5643, 5647, 5651, 5653, 5657, 5661, 5663, 5667, 5671, 5673, 5677, 5681, 5683, 5687, 5691, 5693, 5697, 5701, 5703, 5707, 5711, 5713, 5717, 5721, 5723, 5727, 5731, 5733, 5737, 5741, 5743, 5747, 5751, 5753, 5757, 5761, 5763, 5767, 5771, 5773, 5777, 5781, 5783, 5787, 5791, 5793, 5797, 5801, 5803, 5807, 5811, 5813, 5817, 5821, 5823, 5827, 5831, 5833, 5837, 5841, 5843, 5847, 5851, 5853, 5857, 5861, 5863, 5867, 5871, 5873, 5877, 5881, 5883, 5887, 5891, 5893, 5897, 5901, 5903, 5907, 5911, 5913, 5917, 5921, 5923, 5927, 5931, 5933, 5937, 5941, 5943, 5947, 5951, 5953, 5957, 5961, 5963, 5967, 5971, 5973, 5977, 5981, 5983, 5987, 5991, 5993, 5997, 6001, 6003, 6007, 6011, 6013, 6017, 6021, 6023, 6027, 6031, 6033, 6037, 6041, 6043, 6047, 6051, 6053, 6057, 6061, 6063, 6067, 6071, 6073, 6077, 6081, 6083, 6087, 6091, 6093, 6097, 6101, 6103, 6107, 6111, 6113, 6117, 6121, 6123, 6127, 6131, 6133, 6137, 6141, 6143, 6147, 6151, 6153, 6157, 6161, 6163, 6167, 6171, 6173, 6177, 6181, 6183, 6187, 6191, 6193, 6197, 6201, 6203, 6207, 6211, 6213, 6217, 6221, 6223, 6227, 6231, 6233, 6237, 6241, 6243, 6247, 6251, 6253, 6257, 6261, 6263, 6267, 6271, 6273, 6277, 6281, 6283, 6287, 6291, 6293, 6297, 6301, 6303, 6307, 6311, 6313, 6317, 6321, 6323, 6327, 6331, 6333, 6337, 6341, 6343, 6347, 6351, 6353, 6357, 6361, 6363, 6367, 6371, 6373, 6377, 6381, 6383, 6387, 6391, 6393, 6397, 6401, 6403, 6407, 6411, 6413, 6417, 6421, 6423, 6427, 6431, 6433, 6437, 6441, 6443, 6447, 6451, 6453, 6457, 6461, 6463, 6467, 6471, 6473, 6477, 6481, 6483, 6487, 6491, 6493, 6497, 6501, 6503, 6507, 6511, 6513, 6517, 6521, 6523, 6527, 6531, 6533, 6537, 6541, 6543, 6547, 6551, 6553, 6557, 6561, 6563, 6567, 6571, 6573, 6577, 6581, 6583, 6587, 6591, 6593, 6597, 6601, 6603, 6607, 6611, 6613, 6617, 6621, 6623, 6627, 6631, 6633, 6637, 6641, 6643, 6647, 6651, 6653, 6657, 6661, 6663, 6667, 6671, 6673, 6677, 6681, 6683, 6687, 6691, 6693, 6697, 6701, 6703, 6707, 6711, 6713, 6717, 6721, 6723, 6727, 6731, 6733, 6737, 6741, 6743, 6747, 6751, 6753, 6757, 6761, 6763, 6767, 6771, 6773, 6777, 6781, 6783, 6787, 6791, 6793, 6797, 6801, 6803, 6807, 6811, 6813, 6817, 6821, 6823, 6827, 6831, 6833, 6837, 6841, 6843, 6847, 6851, 6853, 6857, 6861, 6863, 6867, 6871, 6873, 6877, 6881, 6883, 6887, 6891, 6893, 6897, 6901, 6903, 6907, 6911, 6913, 6917, 6921, 6923, 6927, 6931, 6933, 6937, 6941, 6943, 6947, 6951, 6953, 6957, 6961, 6963, 6967, 6971, 6973, 6977, 6981, 6983, 6987, 6991, 6993, 6997, 7001, 7003, 7007, 7011, 7013, 7017, 7021, 7023, 7027, 7031, 7033, 7037, 7041, 7043, 7047, 7051, 7053, 7057, 7061, 7063, 7067, 7071, 7073, 7077, 7081, 7083, 7087, 7091, 7093, 7097, 7101, 7103, 7107, 7111, 7113, 7117, 7121, 7123, 7127, 7131, 7133, 7137, 7141, 7143, 7147, 7151, 7153, 7157, 7161, 7163, 7167, 7171, 7173, 7177, 7181, 7183, 7187, 7191, 7193, 7197, 7201, 7203, 7207, 7211, 7213, 7217, 7221, 7223, 7227, 7231, 7233, 7237, 7241, 7243, 7247, 7251, 7253, 7257, 7261, 7263, 7267, 7271, 7273, 7277, 7281, 7283, 7287, 7291, 7293, 7297, 7301, 7303, 7307, 7311, 7313, 731



Figura 5: Prueba de que llego a n = 1 millon

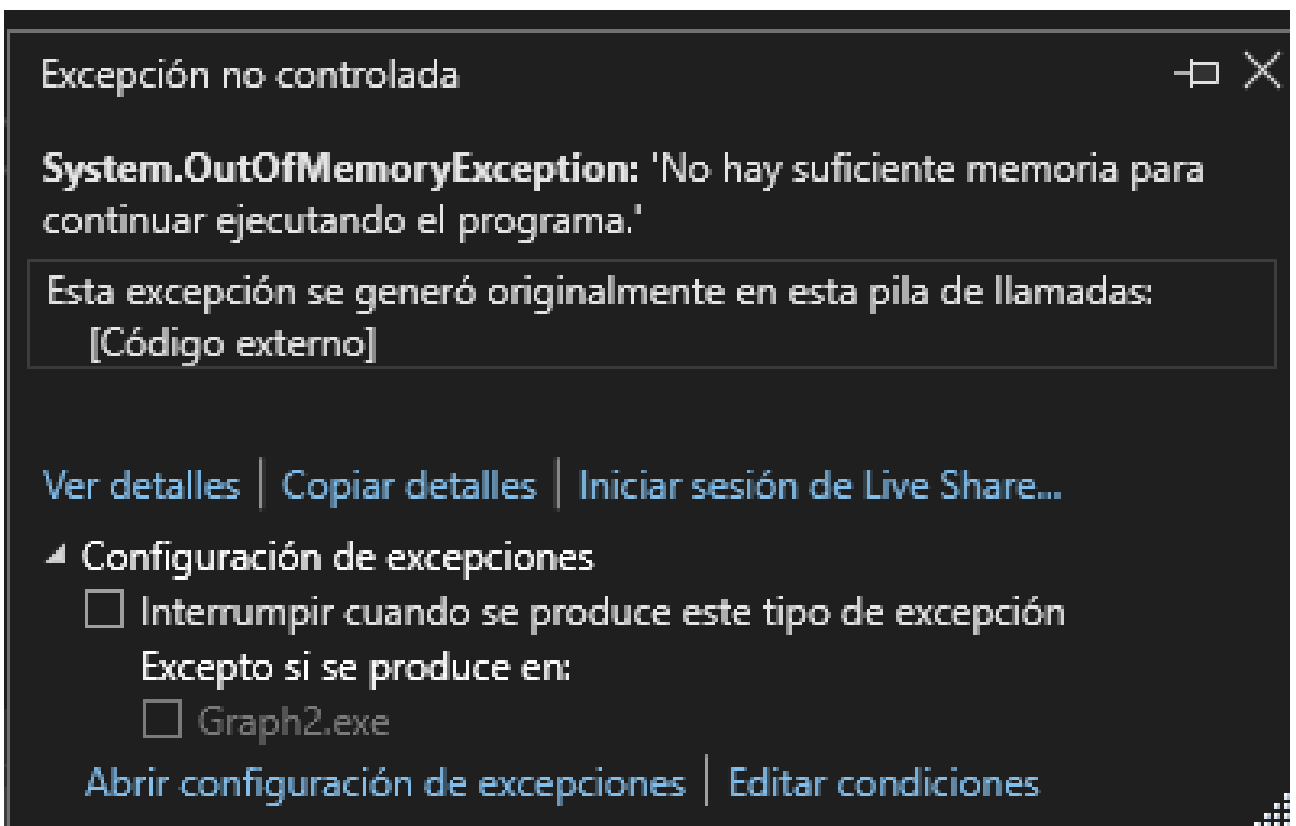


Figura 6: Error por ser demasiados Datos

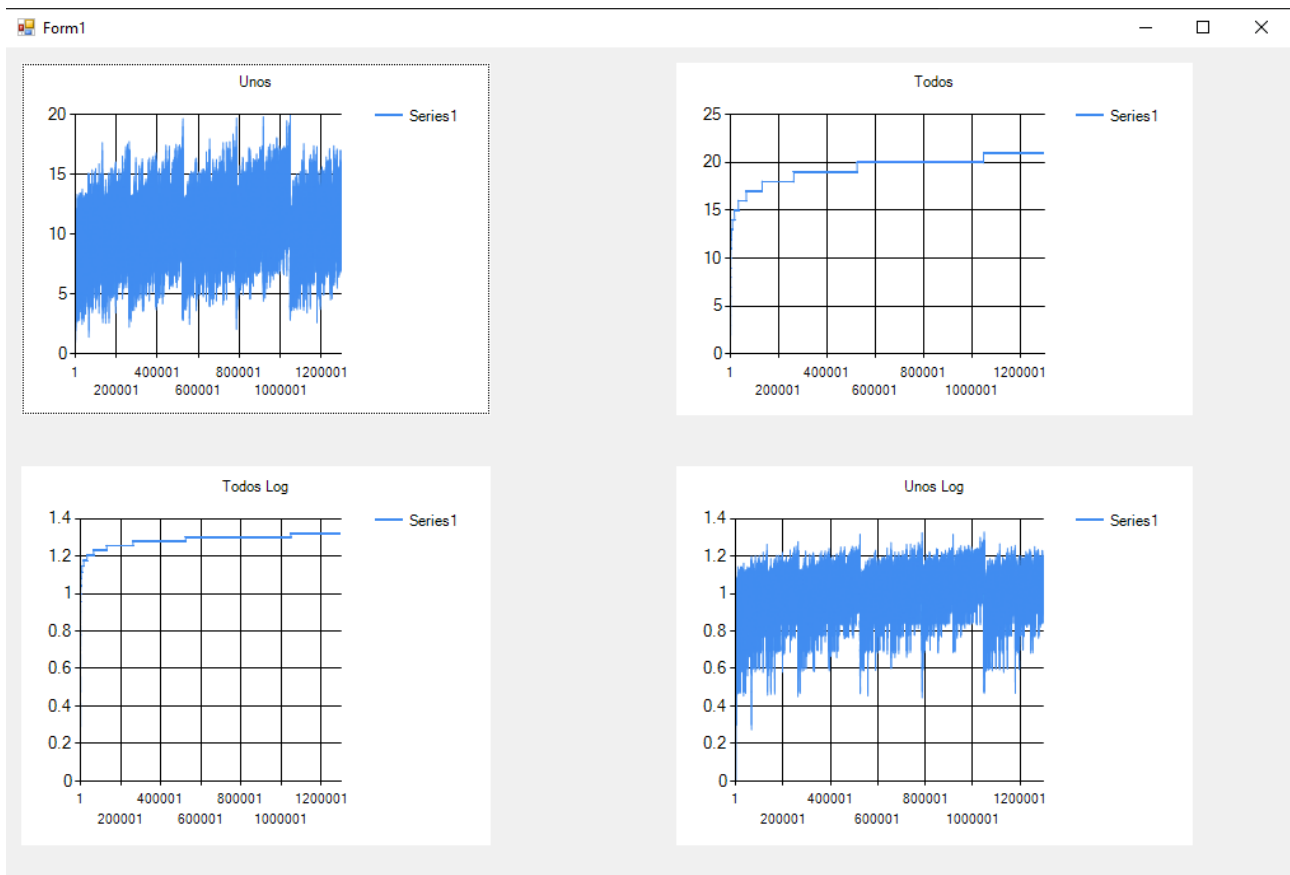


Figura 7: Graficas con $n = 100\,000$

3. Código

3.1. Escritura de Datos

Listing 1: Escritura de Datos C# .net 6.0

```
1 using System;
2 using System.Collections.Generic;
3 using System.Diagnostics;
4 using System.Linq;
5
6 namespace Programa2
7 {
8     public class Programa2
9     {
10         public static void Main(string[] args)
11         {
12             Menu();
13         }
14         public static void Menu()
15         {
16             int num = 0;
17             const string Menus = "1. Escoger Numero:\n2. Al azar:\n3.Salir
18             .";
19             Console.WriteLine(">>Menu<<");
20             Console.WriteLine("Eliga una opcion");
21             Console.WriteLine(Menus);
22             int opc = Convert.ToInt32(Console.ReadLine());
23             switch (opc)
24             {
25                 case 1:
26                     Console.WriteLine("Ingresa k: ");
27                     num = Convert.ToInt32(Console.ReadLine() ?? "0");
28                     Procesos(num);
29                     break;
30                 case 2:
31                     Random rnd = new Random();
32                     num = rnd.Next(2, 10000);
33                     Console.WriteLine($"Se escogio un numero al azar: {num
34                     }");
35                     Procesos(num);
36                     break;
37                 case 3:
38                     Console.WriteLine("Adios");
39                     break;
40                 default:
41                     Menu();
42                     break;
43             }
44         }
45         public static void Procesos(int nume)
46         {
47             Stopwatch stopwatch = new Stopwatch();
48             stopwatch.Start();
49             bool siono = File.Exists("D:\\Documentos\\ESCOM\\Teoria
50             Computacional\\DatosP2\\primosBinario.txt");
51             bool siono2 = File.Exists("D:\\Documentos\\ESCOM\\Teoria
52             Computacional\\DatosP2\\primos.txt");
```



```

49     bool siono3 = File.Exists("D:\\Documentos\\ESCOM\\Teoria
      Computacional\\DatosP2\\primosConjunto.txt");
50     bool siono4 = File.Exists("D:\\Documentos\\ESCOM\\Teoria
      Computacional\\DatosP2\\primosBinarioConjunto.txt");
51     if (siono == true || siono2 == true || siono3 || true)
52     {
53         File.Delete("D:\\Documentos\\ESCOM\\Teoria Computacional\\
          DatosP2\\primosBinario.txt");
54         File.Delete("D:\\Documentos\\ESCOM\\Teoria Computacional\\
          DatosP2\\primos.txt");
55         File.Delete("D:\\Documentos\\ESCOM\\Teoria Computacional\\
          DatosP2\\primosConjunto.txt");
56         File.Delete("D:\\Documentos\\ESCOM\\Teoria Computacional\\
          DatosP2\\primosBinarioConjunto.txt");
57     }
58     File.AppendAllText("D:\\Documentos\\ESCOM\\Teoria
      Computacional\\DatosP2\\primosBinarioConjunto.txt", "Los
      primos en Binario son: {");
59     File.AppendAllText("D:\\Documentos\\ESCOM\\Teoria
      Computacional\\DatosP2\\primosConjunto.txt", "Los primos
      son : {");
60     int n = 2, total = 1;
61     while (total <= nume)
62     {
63         bool esPrimo = true;
64         for (int i = 2; i < n; i++)
65         {
66             if (n % i == 0)
67             {
68                 esPrimo = false;
69                 break;
70             }
71         }
72         if (esPrimo)
73         {
74             string binario = Convert.ToString(n, 2);
75             File.AppendAllText("D:\\Documentos\\ESCOM\\Teoria
              Computacional\\DatosP2\\primosBinario.txt",
              binario + "\\n");
76             File.AppendAllText("D:\\Documentos\\ESCOM\\Teoria
              Computacional\\DatosP2\\primos.txt", n.ToString()
              + "\\n");
77             File.AppendAllText("D:\\Documentos\\ESCOM\\Teoria
              Computacional\\DatosP2\\primosBinarioConjunto.txt"
              , binario + ", ");
78             File.AppendAllText("D:\\Documentos\\ESCOM\\Teoria
              Computacional\\DatosP2\\primosConjunto.txt", n.
              ToString() + ", ");
79             total++;
80         }
81         n++;
82     }
83     stopwatch.Stop();
84     Console.WriteLine("Se tardo en escribir: {0}", stopwatch.
      Elapsed.ToString("hh\\:mm\\:ss\\.fff"));
85     Console.WriteLine("Exito");
86     Console.Beep(200, 5000);
87     Menu();
88 }

```

3.2. Grafica/Lectura Datos

Listing 2: Grafica/Lectura Datos C# (.net 4.8)

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Diagnostics;
6 using System.Drawing;
7 using System.IO;
8 using System.Linq;
9 using System.Text;
10 using System.Threading.Tasks;
11 using System.Windows.Forms;
12 using System.Windows.Forms.DataVisualization.Charting;
13
14 namespace Graph2
15 {
16     public partial class Form1 : Form
17     {
18         static List<UInt64> cuentas;
19         static List<UInt64> cuentasp;
20         static List<UInt64> cusas;
21         static List<Double> cuentasLog;
22         static List<Double> cusasLog;
23         public Form1()
24         {
25             InitializeComponent();
26         }
27         public static void leer()
28         {
29             Stopwatch stopwatch = new Stopwatch();
30             stopwatch.Start();
31             StreamReader lectura = new StreamReader("D:\\Documentos\\ESCOM
32             \\Teoria Computacional\\DatosP2\\primosBinario.txt");
33             StreamReader lectura2 = new StreamReader("D:\\Documentos\\
34             ESCOM\\Teoria Computacional\\DatosP2\\primos.txt");
35             string linea, linea2;
36             int cuenta = 0, usass = 0;
37             Double cuentaLog = 0, usassLog = 0;
38             cuentas = new List<UInt64>();
39             cuentasp = new List<UInt64>();
40             cusas = new List<UInt64>();
41             cuentasLog = new List<Double>();
42             cusasLog = new List<Double>();
43             while ((linea = lectura.ReadLine()) != null)
44             {
45                 cuenta = linea.Count(character => character.Equals('1'));
46                 usass = linea.Count();
47                 cuentas.Add(Convert.ToUInt64(cuenta));
48                 cusas.Add(Convert.ToUInt64(usass));
49             }
50             while ((linea2 = lectura2.ReadLine()) != null)
51             {
52                 cuentasp.Add(Convert.ToUInt64(linea2));
53             }
54             for (int i = 0; i < cuentas.Count(); i++)
55             {
56                 cuentaLog = Math.Log(Convert.ToDouble(cuentas[i]), 10);
```

```

55         cuentasLog.Add(cuentaLog);
56         usassLog = Math.Log(Convert.ToDouble(cusas[i]), 10);
57         cusasLog.Add(Convert.ToDouble(usassLog));
58         Console.WriteLine(cuentaLog);
59     }
60     stopwatch.Stop();
61     Console.WriteLine("Se tardo en leer: {0}", stopwatch.Elapsed.
        ToString("hh\\:mm\\:ss\\.fff"));
62     Console.Beep(200, 1000);
63     lectura.Close();
64     lectura2.Close();
65 }
66 private void Form1_Load(object sender, EventArgs e)
67 {
68     leer();
69     chart1.Palette = ChartColorPalette.BrightPastel;
70     chart2.Palette = ChartColorPalette.BrightPastel;
71     chart3.Palette = ChartColorPalette.BrightPastel;
72     chart4.Palette = ChartColorPalette.BrightPastel;
73     chart1.Titles.Add("Unos");
74     chart2.Titles.Add("Todos");
75     chart3.Titles.Add("Unos Log");
76     chart4.Titles.Add("Todos Log");
77     for (int i = 0; i < cuentas.Count(); i++)
78     {
79         chart1.Series["Series1"].Points.AddXY(Convert.ToInt32(
            cuentasp[i]), cuentas[i]);
80         chart2.Series["Series1"].Points.AddXY(Convert.ToInt32(
            cuentasp[i]), cusas[i]);
81         chart3.Series["Series1"].Points.AddXY(Convert.ToInt32(
            cuentasp[i]), cuentasLog[i]);
82         chart4.Series["Series1"].Points.AddXY(Convert.ToInt32(
            cuentasp[i]), cusasLog[i]);
83     }
84 }
85 }
86 }

```
