

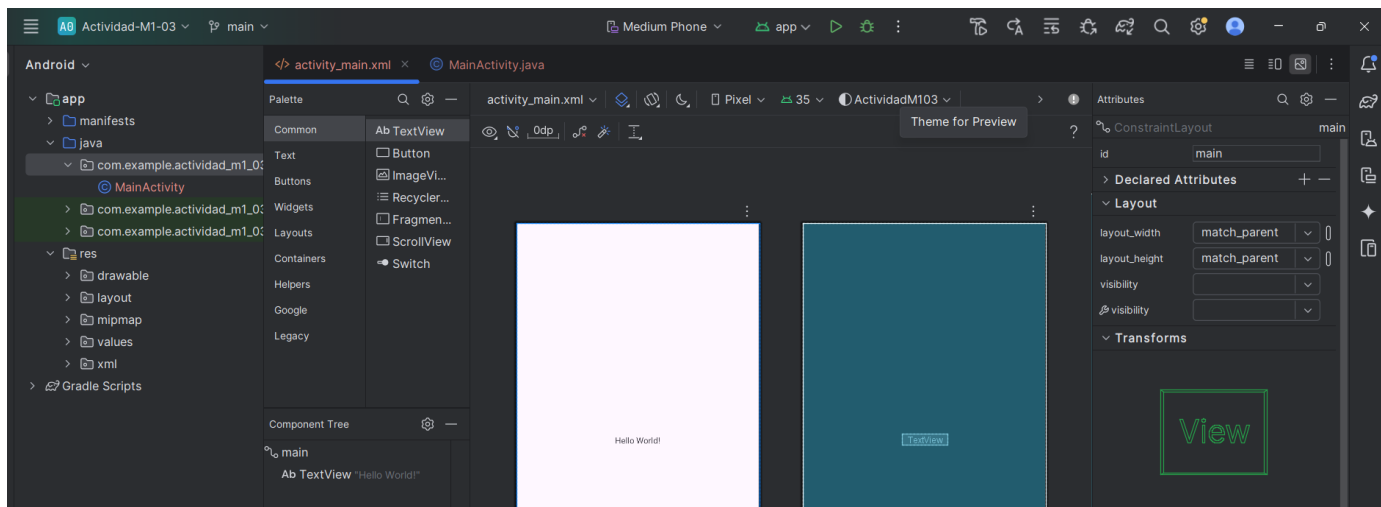
ACTIVIDAD M1-03

Arreglos y tiempo de ejecución



Jorge Ernesto Ceballos Guzman

1. Crear tu primer activity
2. En lenguaje de programación JAVA dentro de Android Studio, crea una app vacía.



3. Crear un arreglo con 1 millón de elementos enteros de manera aleatoria.

Código que solo llega a 805:

```
//Declaramos nuestro arreglo que contendra los numeros aleatorios
int[] numeracion;

//Inicializamos el arreglo con 1000000 numeros aleatorios
numeracion = new int [1000000];
//Inicializamos el generador de numeros aleatorios
Random random = new Random();

//Llenamos el arreglo con numeros aleatorios
for (int i = 0; i < numeracion.length; i++)
{
    numeracion[i] = random.nextInt( bound: 1000000);
}

//Ordenamos el arreglo
Arrays.sort(numeracion);

//Imprimimos el arreglo con Log
Log.d( tag: "MainActivity", msg: "Numero: " + Arrays.toString(Arrays.copyOf(numeracion, newLength: 1000000)));
```

Código con el que podría llegar hasta el millón:

```
//Declaramos nuestro arreglo que contendra los numeros aleatorios
int[] numeracion;

//Inicializamos el arreglo con 1000000 numeros aleatorios
numeracion = new int [1000000];
//Inicializamos el generador de numeros aleatorios
Random random = new Random();

//Llenar el arreglo con números aleatorios
for (int i = 0; i < numeracion.length; i++) {
    numeracion[i] = random.nextInt( bound: 1000000);
}

//Ordenar el arreglo
Arrays.sort(numeracion);

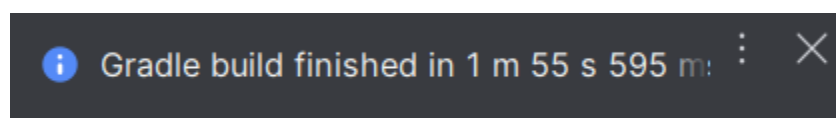
//Imprimir en bloques de 500, ya que sino, solo llega hasta 805
for (int i = 0; i < numeracion.length; i += 500) {
    StringBuilder sb = new StringBuilder();
    for (int j = i; j < i + 500 && j < numeracion.length; j++) {
        sb.append(numeracion[j]).append(", ");
    }
    Log.d( tag: "Numeracion", sb.toString());
}
```

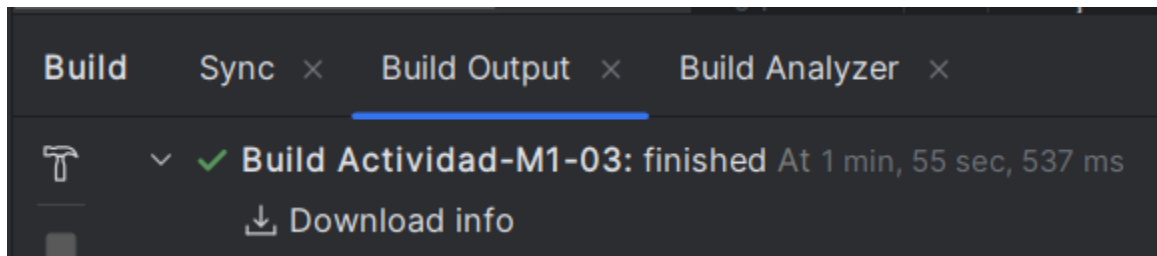
Use 3 fors, uno genera el arreglo del millón de números, el segundo recorre al arreglo en bloques de 500, en ese mismo for, se crea un `StringBuilder`, en donde se acumularan los números, además es mucho más eficiente que una simple concatenación

El tercer for también recorre el arreglo, pero lo hace comprobando de no pasarse con la parte de `j < numeracion.length`, además, va agregando el numero `j` a `StringBuilder`, siguiendo el formato de una coma y un espacio.

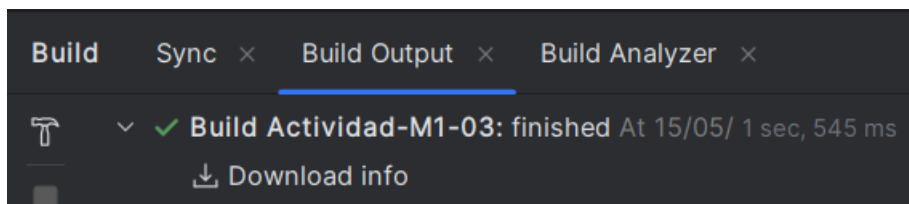
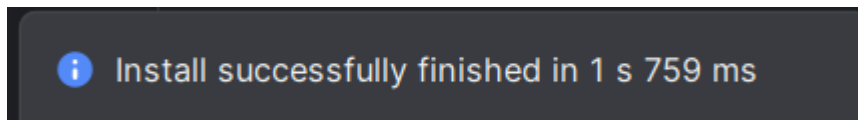
Finalmente, se convierte el `StringBuilder`, a un `String` y luego se imprime en el log con la tag de numeración

4. Ordenar los elementos.
5. Medir el tiempo de inicio a fin.
6. Código que solo llega a 805:

 Gradle build finished in 1 m 55 s 595 ms



Código con el que podría llegar hasta el millón:



7. Imprimir en el logcat los resultados obtenidos.

Aunque al investigar un `Int` si acepta un millón de números, al parecer el como funciona log cat no, por lo que únicamente imprimia hasta 805:

Numero: [0, 2, 3, 4, 4, 5, 6, 7, 9, 12, 13, 15, 15, 16, 16, 17, 17, 19, 19, 19, 21, 22, 23, 23, 23, 25, 25, 25, 26, 28, 29, 30, 31, 31, 31, 31, 32, 32, 33, 34, 35, 40, 41, 41, 42, 44, 45, 48, 50, 51, 52, 53, 53, 53, 54, 55, 55, 56, 57, 57, 58, 58, 59, 61, 61, 61, 62, 62, 68, 68, 70, 72, 74, 75, 78, 79, 80, 80, 82, 83, 84, 85, 85, 86, 87, 91, 92, 92, 92, 93, 94, 95, 95, 96, 96, 98, 100, 100, 102, 103, 103, 104, 104, 104, 106, 107, 108, 110, 111, 111, 112, 112, 113, 114, 115, 116, 118, 118, 119, 119, 121, 121, 122, 123, 125, 126, 129, 129, 129, 130, 131, 132, 132, 132, 133, 135, 136, 136, 141, 142, 142, 144, 145, 145, 147, 148, 149, 153, 154, 154, 155, 159, 159, 159, 160, 160, 160, 162, 164, 165, 166, 166, 167, 171, 171, 173, 174, 178, 179, 180, 180, 181, 181, 181, 182, 182, 184, 187, 189, 189, 190, 190, 191, 191, 191, 191, 193, 194, 194, 194, 196, 198, 198, 198, 200, 201, 202, 202, 205, 206, 206, 207, 208, 209, 211, 211, 212, 213, 215, 215, 215, 216, 218, 218, 218, 221, 222, 225, 225, 225, 225, 226, 227, 227, 227, 229, 230, 230, 230, 231, 232, 232, 232, 233, 234, 236, 236, 236, 237, 239, 239, 239, 240, 241, 241, 242, 242, 242, 242, 243, 243, 244, 244, 244, 245, 247, 248, 249, 249, 250, 251, 251, 253, 255, 255, 256, 256, 258, 258, 259, 259, 259, 260, 265, 265, 266, 266, 269, 269, 269, 270, 270, 272, 272, 272, 273, 273, 274, 274, 276, 277, 281, 283, 283, 286, 287, 287, 289, 289, 290, 291, 291, 292, 292, 294, 294, 296, 296, 298, 298, 299, 299, 299, 300, 300, 300, 301, 301, 303, 304, 304, 305, 305, 305, 306, 306, 306, 307, 310, 311, 312, 313, 313, 313, 313, 314, 315, 319, 321, 322, 323, 324, 325, 327, 328, 330, 332, 332, 334, 334, 334, 336, 336, 338, 338, 338, 340, 340, 342, 343, 344, 344, 344, 346, 346, 347, 347, 349, 349, 350, 350, 351, 356, 357, 358, 360, 361, 364, 365, 365, 366, 366, 367, 367, 368, 368, 370, 371, 372, 372, 372, 379, 379, 381, 382, 383, 383, 383, 384, 386, 387, 387, 388, 390, 391, 392, 393, 395, 395, 398, 398, 399, 401, 402, 403, 404, 406, 406, 407, 407, 408, 408, 409, 410,

411, 414, 415, 415, 416, 416, 418, 419, 419, 420, 420, 420, 422, 423, 424, 425, 426, 426,
426, 427, 427, 428, 429, 429, 435, 439, 440, 440, 440, 441, 444, 445, 446, 447, 449, 449,
450, 451, 452, 453, 454, 454, 455, 455, 455, 455, 456, 456, 456, 457, 460, 462, 463, 466,
467, 467, 468, 468, 469, 469, 470, 472, 472, 472, 472, 475, 477, 477, 478, 479, 480, 481,
481, 481, 482, 484, 487, 487, 488, 490, 491, 491, 492, 492, 492, 492, 493, 496, 496, 497,
499, 502, 503, 504, 506, 507, 508, 509, 509, 509, 510, 511, 512, 513, 515, 519, 520, 521,
521, 523, 524, 524, 526, 527, 527, 528, 531, 532, 533, 534, 535, 536, 537, 538, 538, 538,
539, 540, 543, 543, 544, 545, 545, 546, 546, 549, 550, 551, 551, 552, 552, 553, 554, 555,
557, 557, 557, 558, 560, 560, 561, 561, 563, 563, 564, 564, 565, 565, 565, 566, 566, 566,
567, 568, 568, 571, 574, 575, 575, 575, 575, 576, 576, 577, 577, 578, 579, 579, 581, 582,
583, 583, 584, 586, 588, 588, 588, 588, 588, 588, 590, 590, 593, 593, 594, 596, 597, 598, 598,
599, 599, 599, 600, 603, 603, 606, 607, 609, 610, 610, 611, 613, 614, 615, 616, 616, 616,
617, 617, 618, 618, 619, 619, 620, 621, 621, 621, 623, 623, 626, 627, 628, 629, 630, 630,
630, 632, 633, 634, 635, 636, 637, 637, 638, 638, 639, 639, 639, 639, 641, 642, 642, 644,
645, 645, 649, 649, 651, 652, 652, 653, 655, 655, 655, 656, 657, 657, 658, 658, 659, 660,
661, 662, 662, 662, 666, 668, 668, 669, 669, 673, 673, 674, 674, 675, 678, 680, 680, 681,
681, 683, 684, 686, 686, 686, 687, 687, 689, 689, 689, 689, 690, 692, 692, 693, 693, 693,
694, 694, 694, 694, 697, 698, 699, 700, 700, 700, 701, 701, 702, 704, 704, 706, 707, 710,
712, 712, 713, 716, 718, 719, 719, 721, 722, 722, 724, 724, 725, 725, 725, 726, 726, 727,
727, 727, 728, 731, 731, 733, 736, 737, 743, 744, 745, 746, 750, 750, 750, 752, 754, 755,
755, 757, 760, 761, 763, 764, 764, 765, 765, 765, 766, 766, 769, 770, 772, 772, 773, 775,
775, 776, 777, 777, 779, 780, 781, 781, 781, 782, 783, 784, 787, 790, 791, 793, 796, 797,
798, 799, 800, 801, 801, 801, 802, 802, 803, 805, 805,

Segundo código, al parecer si llega al millón, pero el LogCat me lo muestra desde del 873865
(toda la impresión serian más de 360 páginas únicamente empezando desde el 873865)

873865, 873870, 873872, 873875, 873880, 873881, 873882, 873886, 873886, 873890,
873895, 873895, 873895, 873897, 873899, 873899, 873899, 873900, 873902, 873905,
873906, 873906, 873907, 873908, 873909, 873909, 873909, 873910, 873910, 873912,
873914, 873914, 873915, 873918, 873919, 873920, 873924, 873924, 873925, 873925,
873926, 873927, 873928, 873928, 873929, 873929, 873929, 873930, 873931, 873932,
873932, 873932, 873933, 873933, 873933, 873934, 873937, 873937, 873938, 873940,
873940, 873940, 873942, 873942, 873943, 873945, 873946, 873948, 873948, 873949,
873952, 873954, 873954, 873956, 873956, 873957, 873958, 873960, 873960, 873961,
873962, 873963, 873963, 873964, 873965, 873965, 873966, 873966, 873966, 873967,
873967, 873970, 873970, 873971, 873972, 873973, 873973, 873973, 873975, 873977,
873977, 873978, 873979, 873981, 873982, 873982, 873982, 873983, 873983, 873983,
873985, 873985, 873986, 873986, 873988, 873989, 873989, 873989, 873992, 873992,
873993, 873997, 874000, 874000, 874001, 874003, 874004, 874004, 874006, 874006,
874008, 874008, 874010, 874015, 874015, 874015, 874015, 874018, 874020, 874020,
874020, 874021, 874021, 874021, 874023, 874023, 874023, 874024, 874025, 874025,

874026, 874030, 874032, 874033, 874033, 874034, 874036, 874036, 874036, 874039,
874039, 874041, 874042, 874046, 874048, 874049, 874049, 874049, 874050, 874051,
874051, 874051, 874052, 874052, 874052, 874053, 874056, 874056, 874056, 874058,
874058, 874059, 874060, 874061, 874061, 874064, 874065, 874068, 874069, 874072,
874072, 874072, 874072, 874074, 874075, 874077, 874078, 874078, 874078,
874080, 874081, 874082, 874083, 874084, 874085, 874086, 874086, 874088, 874088,
874089, 874089, 874090, 874090, 874091, 874091, 874093, 874093, 874094, 874094,
874094, 874094, 874096, 874097, 874097, 874098, 874098, 874100, 874100,
874100, 874101, 874101, 874103, 874103, 874106, 874107, 874108, 874110, 874111,
874115, 874117, 874118, 874120, 874121, 874124, 874124, 874126, 874127, 874127,
874128, 874128, 874129, 874130, 874130, 874130, 874131, 874132, 874132, 874133,
874133, 874134, 874136, 874137, 874137, 874139, 874140, 874141, 874142, 874143,
874143, 874144, 874144, 874145, 874145, 874145, 874147, 874149, 874149, 874150,
874152, 874153, 874154, 874154, 874154, 874155, 874156, 874156, 874158, 874159,
874159, 874159, 874161, 874163, 874164, 874164, 874165, 874166, 874167, 874167,
874168, 874168, 874169, 874170, 874170, 874171, 874172, 874172, 874172, 874173,
874175, 874175, 874176, 874178, 874179, 874180, 874181, 874182, 874182, 874183,
874183, 874184, 874186, 874186, 874187, 874188, 874189, 874194, 874194, 874198,
874198, 874199, 874200, 874202, 874203, 874204, 874204, 874205, 874207, 874207,
874208, 874210, 874211, 874211, 874211, 874212, 874213, 874213, 874214, 874214,
874215, 874220, 874220, 874223, 874224, 874227, 874227, 874227, 874228, 874229,
874229, 874230, 874231, 874233, 874234, 874234, 874235, 874235, 874236, 874236,
874238, 874238, 874239, 874239, 874240, 874242, 874243, 874244, 874244, 874245,
874246, 874246, 874248, 874248, 874249, 874249, 874250, 874251, 874252, 874253,
874254, 874255, 874255, 874256, 874257, 874258, 874258, 874259, 874260, 874260,
874260, 874260, 874262, 874262, 874263, 874266, 874266, 874268, 874269, 874269,
874269, 874270, 874271, 874272, 874272, 874273, 874273, 874273, 874275, 874275,
874276, 874277, 874277, 874278, 874281, 874282, 874283, 874284, 874284, 874287,
874288, 874290, 874291, 874291, 874293, 874295, 874295, 874299, 874300, 874300,
874301, 874302, 874303, 874303, 874303, 874304, 874311, 874312, 874312, 874313,
874314, 874315, 874315, 874317, 874318, 874318, 874320, 874320, 874322,
874322, 874323, 874324, 874325, 874325, 874325, 874326, 874328, 874329, 874334,
874334, 874335, 874335, 874337, 874337, 874337, 874338, 874338, 874339, 874339,
874339, 874340, 874340, 874341, 874341, 874342, 874343, 874343, 874344,
874345, 874345, 874345, 874346, 874346, 874347, 874349, 874350, 874352, 874353,

