



TECNICATURA UNIVERSITARIA EN DISEÑO
INTEGRAL DE VIDEOJUEGOS

FACULTAD DE INGENIERÍA
Universidad Nacional de Jujuy



FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS

Trabajo Practico N°1

Facundo Condori

TUV000667

Profesores:

Mg. Ing. Ariel Alejandro Vega

Ing. Carolina Cecilia Apaza

Año

Indice

Punto 1:	3
Punto 2:	4
Punto 3:	6
Punto 4:	6
Valores asignados a las variables:	6
Valores asignados a las variables:	6
Punto 5:	10
Punto 6:	14
Punto 7:	15
Punto 8:	16
Punto 9:	17
Punto 10:	18
Punto 11:	19
Punto 12:	20
Punto 13:	22
Punto 14:	24
Punto 15:	26
Punto 16:	28
Punto 17:	30
Punto 18:	33
Punto 19:	34
Punto 20:	36
Punto 21:	38
Punto 22:	41



Punto 1:

Ejercicio 1: Evaluar(obtener resultado) la siguiente expresión para

$A = 2$ y $B = 5$

Desarrollo del punto

$3 * A - 4 * B / A ^ 2$

$(3*A)-(4*B/(A^2))$

$3 * A - 4 * B / 4$

$6 - 20 / 4$

$6 - 5$

1

```
ejercicio 1
1 int A = 2;
2 int B = 5;
3 float resultado;
4
5 resultado= 3* A - 4 * B / pow(A, 2);
6 println(resultado);
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

Se ha guardado el sketch.

1.0

Consola Errores



Punto 2:

Ejercicio 2: Evaluar la siguiente expresión

$$4 / 2 * 3 / 6 + 6 / 2 / 1 / 5 ^ 2 / 4 * 2$$

Desarrollo del punto

$$(4 / 2 * 3 / 6) + (6 / 2 / 1 / (5 ^ 2) / 4 * 2)$$

$$(4 / 2 * 3 / 6) + (6 / 2 / 1 / 25 / 4 * 2)$$

$$(2 * 3 / 6) + (6 / 2 / 1 / 25 / 4 * 2)$$

$$(6 / 6) + (6 / 2 / 1 / 25 / 4 * 2)$$

$$1 + (3 / 1 / 25 / 4 * 2)$$

$$1 + (3 / 25 / 4 * 2)$$

$$1 + (3 / 25 / 8)$$

$$1 + (3 / 200)$$

$$1 + 0,015$$

$$1,015$$

$$1,06$$



```
sketch 240412b
1 float resolver;
2 resolver = (4 / 2 * 3 / 6) + (6 / 2 / 1 / pow(5, 2) / 4 * 2);
3 println("resultado ejercicio 2)" + resolver);
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
--
resultado ejercicio 2)1.06
```

Punto 3:

Ejercicio 3: Escribir las siguientes expresiones algebraicas como expresiones algorítmicas (en su forma aritmética dentro del algoritmo). En este caso no se pide evaluarlas ni programarlas.

Punto 4:

Ejercicio 4: Evaluar las siguientes expresiones aritméticas, para lo cual indicar en el caso de las variables, el valor indicado. Luego escribirlas como expresiones algebraicas.

Desarrollo del punto

a) $b^2 - 4 * a * c$

$b^2 - 4ac$

Valores asignados a las variables:

$b=2$

$a=2$

$c=2$

a) $(b^2) - (4 * a * c)$

$4 - (4 * a * c)$

$4 - (8 * c)$

$4 - 16$

-12

b) $3 * X^4 - 5 * X^3 + X^{12} - 17$

$3X^4 - 5X^3 + X^{12} - 17$

Valores asignados a las variables:

$X=2$

b) $(3 * (X^4)) - (5 * (X^3) + X^{12}) - 17$

$(3 * 16) - (5 * (X^3) + X^{12}) - 17$

$$(3 * 16) - (5 * 8 + X 12) - 17$$

$$48 - (5 * 8 + X 12) - 17$$

$$48 - (40 + X 12) - 17$$

$$48 - (40 + 24) - 17$$

$$48 - 64 - 17$$

$$- 16 - 17$$

$$-33$$

```
ejercicio 1  ejercicio 2  ejercicio 4  ▼
1  int b = 2;
2  int a = 2;
3  int c = 2;
4  float resuelve;
5
6  resultado = pow(b, 2) - 4 * a * c;
7  println(resultado);
8
9  int X = 2;
10 float reseuelve;
11 resultado = (3 * pow(X, 4)) - (5 * pow(X, 3) + X * 12) - 17;
12 println(resultado);
13
14
15
16
17
18
19
```

```
1.0
1.06
-12.0
-33.0
```

$$c) (b + d) / (c + 4)$$

$$Bd/c4$$

$$b = h$$

$$b = 2$$

$$d = 1$$

$c = 3$

$(b + d) / (c + 4)$

$3 / (c + 4)$

$3 / 7$

0.5

```
ejercicio 1 | ejercicio 2 | ejercicio 4 | ejercicio
1 int h = 2; ///cambie la b por la h///
2 int d = 1;
3 int c = 3;
4 resultado = (float) (b + d) / (c + 4);
5 println("resultado c): " + resultado);
6
7 /// me tira error, por eso lo hice aparte ///
8
9
10
11
12
13
14
15
16
17
18
19
```

El valor de la variable local «h» no se utiliza

```
1.06
resultado a): -12.0
resultado b): -33.0
resultado c): 0.5
resultado d): 0.0
```

Consola Errores

$d) (x^2 + y^2)^{(1/2)}$

$X = 0$

$Y = 0$

$(x^2 + y^2)^{(1/2)}$

$(0 + y^2)^{(1/2)}$

$(0 + 0)^{(1/2)}$

$0^{(1/2)}$

0^2

0



```
ejercicio 1 | ejercicio 2 | ejercicio 4 | <
1 int x = 0;
2 int y = 0;
3 float resultado;
4 resultado = sqrt(pow(x, 2) + pow(y, 2));
5 println("resultado d): " + resultado);
6
7
8
9
0
1
2
3
4
5
6
7
8
9
0
1.06
resultado a): -12.0
resultado b): -33.0
resultado c): 0.5
resultado d): 0.0
Consola Errores
```



Para aclarar que indicamos con "Luego escribirlas como expresiones algebraicas" lo aplicamos con el punto

a) $b^2 - 4.a.c$

b) $3 . X^4 - 5 . X^3 + X^{12} - 17$

c) $(b + d) / (c + 4)$

d) $(x^2 + y^2)^{(1 / 2)}$

 <p>TUDI Videojuegos Fundamentos de Programación Orientada a Objetos</p>	<p>FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS FACULTAD DE INGENIERÍA Universidad Nacional de Jujuy Trabajo Practico N° / Actividad</p>	
---	---	---

Punto 5:

Ejercicio 5: Si el valor de A es 4, el valor de B es 5 y el valor de C es 1, evaluar las siguientes expresiones:

Desarrollo del punto



a) $B * A - B^2 / 4 * C$

$(B * A) - (B^2 / 4 * C)$

$20 - (B^2 / 4 * C)$

$20 - (25 / 4 * C)$

$20 - 6,25$

$13,75$

```
sketch 240412a
1 println("resultado ejercicio 5");
2 float A = 4;
3 float B = 5;
4 float C = 1;
5 float resolver;
6
7 resolver = (B * A) - (float) Math.pow(B, 2) / (4 * C);
8 System.out.println(resolver);
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2
```

```
sketch 240412a ▼
1 println("resultado ejercicio 5");
2 float A = 4;
3 float B = 5;
4 float C = 1;
5 float resolver;
6
7 resolver = (B * A) - (float) Math.pow(B, 2) / (4 * C);
8 resolver = (A * B) / (float) Math.pow(3, 2);
9 System.out.println(resolver);
10
11
12
13
14
15
16
17
18
19
20
```

```
resultado ejercicio 5
2.2222223
```

Consola Errores

c) $((B + C) / 2 * A + 10) * 3 * B - 6$

$((6 / 2 * A + 10) * 3 * B) - 6$

$((3 * A + 10) * 3 * B) - 6$

$((12 + 10) * 3 * B) - 6$

$(22 * 3 * B) - 6$

$(66 * B) - 6$

$330 - 6$

324



```
sketch 240412a
1 println("resultado ejercicio 5");
2 float A = 4;
3 float B = 5;
4 float C = 1;
5 float resolver;
6
7 resolver = (B * A) - (float) Math.pow(B, 2) / (4 * C);
8 resolver = (A * B) / (float) Math.pow(3, 2);
9 resolver = (((B + C) / 2 * A + 10) * 3 * B) - 6;
10 System.out.println(resolver);
```

```
resultado ejercicio 5
324.0
```

Punto 6:

Ejercicio 6: Para $x=3$, $y=4$; $z=1$, evaluar el resultado de

$$R1 = y+z$$

$$R2 = x \geq R1$$

Desarrollo del punto

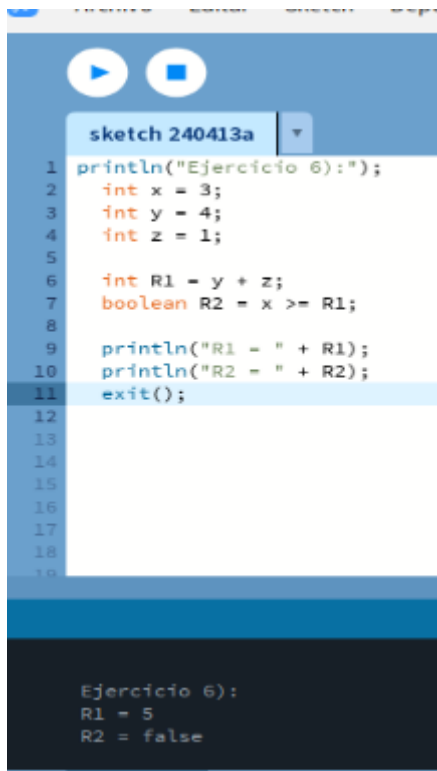
$$R1 = y+z$$

5

$$R2 = x \geq R1$$

$$x \geq 5$$

Falso



```
1 println("Ejercicio 6:");
2 int x = 3;
3 int y = 4;
4 int z = 1;
5
6 int R1 = y + z;
7 boolean R2 = x >= R1;
8
9 println("R1 = " + R1);
10 println("R2 = " + R2);
11 exit();
12
13
14
15
16
17
18
19
```

Ejercicio 6):
R1 = 5
R2 = false

Punto 7:

Ejercicio 7: Para contador1=3, contador3=4, evaluar el resultado de

R1 = ++contador1

R2 = contador1 < contador2

Desarrollo del punto

```
ejercicio 7
1 println("resultado ejercicio 7:");
2 int contador1 = 3;
3 int contador2 = 4;
4 ///se me bugg el void setup()///
5 int R1 = ++contador1;
6 boolean R2 = contador1 < contador2;
7 println("R1: " + R1);
8 println("R2: " + R2);
9 noLoop();
10
11
12
13
14
15
16
17
18
19

resultado ejercicio 7):
R1: 4
R2: false

Console Errores
```

Punto 8:

Ejercicio 8: Para $a=31$, $b=-1$; $x=3$, $y=2$, evaluar el resultado de

$$a+b-1 < x*y$$

Desarrollo del punto

```
sketch 240413b
1 println("ejercicio 8:");
2   int a = 31;
3   int b = -1;
4   int x = 3;
5   int y = 2;
6   boolean resultado = (a + b - 1) < (x * y);
7   println(resultado);
8   noLoop();
9
10
11
12
13
14
15
16
17
18
19
ejercicio 8):
false
```


Punto 9:

Ejercicio 9: Para $x=6$, $y=8$, evaluar el resultado de

$!(x < 5) \text{ CC } !(y \geq 7)$

Desarrollo del punto

```
1 void setup() {  
2   println("resultado ejercicio 9");  
3   int x = 6;  
4   int y = 8;  
5   boolean resultado = !(x < 5) && !(y >= 7);  
6   println(resultado);  
7   noLoop();  
8 }  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100
```

```
resultado ejercicio 9  
false
```

Punto 10:

Ejercicio 10: Para $i=22, j=3$, evaluar el resultado de

$!((i > 4) \parallel !(j \leq 6))$

Desarrollo del punto

```
sketch 240413d
1 void setup() {
2   println("ejercicio 10:");
3   int i = 22;
4   int j = 3;
5   boolean resultado = !((i > 4) || !(j <= 6));
6   println(resultado);
7   noLoop();
8 }
9
10
11
12
13
14
15
16
17
18
19
20
ejercicio 10):
false
```

Punto 11:

Ejercicio 11: Para $a=34$, $b=12$, $c=8$, evaluar el resultado de $!(a+b==c) \parallel (c!=0)CC(b-c \geq 19)$

Desarrollo del punto

```
sketch 240413e ▼
1 void setup() {
2   println("ejercicio 11:");
3   int a = 34;
4   int b = 12;
5   int c = 8;
6   boolean resultado = !(a + b == c) || (c != 0) && (b - c >= 19);
7   println(resultado);
8   noLoop();
9 }
10
11
12
13
14
15
16
17
18
19
ejercicio 11):
true
```

Punto 12:

Sección Análisis – Diseño y Codificación de algoritmos – Aplicación de estructuras de control Para cada ejercicio, en el archivo Word agregar las secciones de análisis y diseño, mientras que, para la codificación, crear el archivo de Processing.

Ejercicio 12: Un problema sencillo. Deberá pedir por teclado al usuario un nombre y posteriormente realizará la presentación en pantalla de un saludo con el nombre indicado.

Desarrollo del punto

Análisis

Datos de entrada:

Nombre del usuario: string

Datos de salida:

Saludo personalizado que incluya el nombre ingresado: string

Proceso:

Solicitar al usuario que ingrese su nombre.

Leer y almacenar el nombre proporcionado por el usuario.

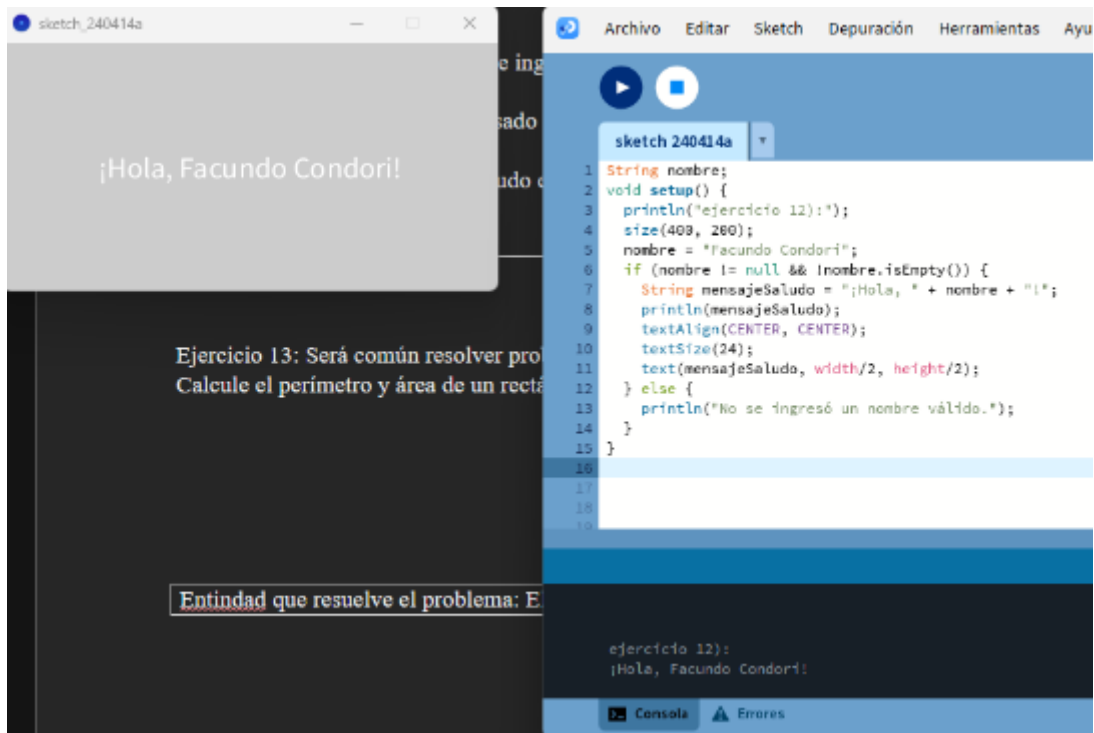
Construir un mensaje de saludo que incluya el nombre ingresado.

Mostrar el mensaje de saludo en pantalla.

Diseño:

Entidad que resuelve el problema: El programador
Variables iniciar el programa mostrarSaludo: String
Nombre del algoritmo nombre_saludo Proceso del algoritmo 1. Solicitar entrada 2. <i>Mostrar</i> un mensaje solicitando al usuario que ingrese su nombre

3. *Leer* Entrada
4. *Leer* y almacenar el nombre ingresado por el usuario
5. Construir Saludo
6. Combinar el nombre ingresado en un mensaje de saludo
7. *Mostrar* Saludo
8. Imprimir el mensaje de saludo en la pantalla
9. Fin



Punto 13:

Ejercicio 13: Será común resolver problemas utilizando variables.
Calcule el perímetro y área de un rectángulo dada su base y su altura.

Desarrollo del punto

Análisis:

Datos de entrada:

Base del rectángulo: real.

Altura del rectángulo: real.

Datos de salida:

Perímetro del rectángulo: real.

Área del rectángulo: real.

Fórmulas para el Cálculo:

El perímetro de un rectángulo se calcula como: $2 * (base + altura)$.

El área de un rectángulo se calcula como: $base * altura$.

Entindad que resuelve el problema: El programador
Entindad que resuelve el problema: El programador
Variables base: Real // almacena la base del rectángulo altura: Real // almacena la altura del rectángulo perimetro: Real // almacena el valor del perímetro del rectángulo area: Real // almacena el valor del área del rectángulo Variables base: Real // almacena la base del rectángulo altura: Real // almacena la altura del rectángulo perimetro: Real // almacena el valor del perímetro del rectángulo area: Real // almacena el valor del área del rectángulo
Nombre del algoritmo: calcular_perimetro_area_rectangulo
Proceso del algoritmo: 1. Leer base desde la entrada de usuario. 2. Leer altura desde la entrada de usuario.

3. Calcular el perímetro del rectángulo usando la fórmula:
$$\text{perimetro} = 2 * (\text{base} + \text{altura}).$$
4. Calcular el área del rectángulo usando la fórmula: $\text{area} = \text{base} * \text{altura}.$
5. *Mostrar* el valor del perímetro y el área del rectángulo en la pantalla.

```
sketch 240414b ▼
1 void setup() {
2   println("ejercicio 13:");
3   float base = 20.0;
4   float altura = 10.0;
5   float perimetro = 2 * (base + altura);
6   float area = base * altura;
7   println("Perímetro del rectángulo: " + perimetro);
8   println("Área del rectángulo: " + area);
9   noLoop();
10 }
11
12
13
14
15
16
17
18
19
20
```

```
ejercicio 13):
Perímetro del rectángulo: 60.0
Área del rectángulo: 200.0
```

Consola Errores

Punto 14:

Ejercicio 14: Una ayuda importante al momento de resolver problemas con algoritmos es asumir que su gran amigo son las matemáticas. Obtenga la hipotenusa de un triángulo rectángulo conociendo sus catetos.

Desarrollo del punto

Datos de Entrada:

catetoA: real

catetoB: real

Datos de Salida:

hipotenusa: real

Fórmula Utilizada:

El cálculo de la hipotenusa (c) de un triángulo rectángulo se basa en el teorema de Pitágoras, que establece que en un triángulo rectángulo con catetos a y b, la hipotenusa c se calcula como:

$$C = (\text{raiz})a^2 + b^2(\text{fin de raiz})$$

a y b son las longitudes de los catetos del triángulo rectángulo.

Entidad que Resuelve el Problema: Calculadora de Hipotenusa
Variables: cateto1: Real cateto2: Real hipotenusa: Real
Nombre del Algoritmo: calcular_hipotenusa Proceso del Algoritmo: <ol style="list-style-type: none"> 1. Leer cateto1 2. Leer cateto2 3. Calcular la hipotenusa del triángulo rectángulo utilizando el teorema de Pitágoras: hipotenusa = (raiz)cateto1^2+cateto2^2(fin de raiz) 4. Mostrar el valor de hipotenusa en la consola.



sketch 240414c

```
1 void setup() {  
2   println("ejercicio 14:");  
3   float catetoA = 10;  
4   float catetoB = 5;  
5   float hipotenusa = sqrt(catetoA * catetoA + catetoB * catetoB);  
6   println("La longitud de la hipotenusa es: " + hipotenusa);  
7   noLoop();  
8 }
```

```
ejercicio 14):  
La longitud de la hipotenusa es: 11.18034
```

Punto 15:

Ejercicio 15: Si viste algo de los apuntes y vídeos, esto debería ser muy fácil de resolver. Dados dos números permita calcular la suma, resta, multiplicación y división de estos. Considere que cada una de estas operaciones es un algoritmo cuando realice el diseño. Obviamente muestre los resultados.

Análisis

Datos de Entrada:

numeroA: Real

numeroB: Real

Datos de Salida:

suma: Real

resta: Real

multiplicacion: Real

division: Real

Quién realiza el proceso: Una calculadora

Descripción del Proceso:

La calculadora realiza las siguientes operaciones:

Suma: $\text{suma} = \text{numeroA} + \text{numeroB}$

Resta: $\text{resta} = \text{numeroA} - \text{numeroB}$

Multiplicación: $\text{multiplicacion} = \text{numeroA} * \text{numeroB}$

División:

Si numeroB es distinto de cero: $\text{division} = \text{numeroA} / \text{numeroB}$

Si numeroB es igual a cero: No se puede realizar la división (mensaje de error)

ENTIDAD QUE RESUELVE EL PROBLEMA: Calculadora
Variables: numeroA: Real numeroB: Real resultadoSuma: Real resultadoResta: Real resultadoMultiplicacion: Real resultadoDivision: Real
Nombre del Algoritmo: calcular_operaciones Proceso del Algoritmo: Leer numeroA // Solicitar al usuario que ingrese el primer número (numeroA).



Leer numeroB //

Solicitar al usuario que ingrese el segundo número (numeroB).

Calcular y Mostrar la Suma //

Calcular la suma de numeroA y numeroB y almacenar el resultado en resultadoSuma.

Mostrar el resultado de la suma (resultadoSuma).

Calcular y Mostrar la Resta //

Calcular la resta de numeroA menos numeroB y almacenar el resultado en resultadoResta.

Mostrar el resultado de la resta (resultadoResta).

Calcular y Mostrar la Multiplicación //

Calcular el producto de numeroA por numeroB y almacenar el resultado en resultadoMultiplicacion.

Mostrar el resultado de la multiplicación (resultadoMultiplicacion).

Calcular y Mostrar la División //

Verificar si numeroB es distinto de cero:

Si es así, calcular la división de numeroA entre numeroB y almacenar el resultado en resultadoDivision.

Mostrar el resultado de la división (resultadoDivision).

Si numeroB es cero, mostrar un mensaje indicando que la división por cero no es posible.

```
ejercicio 15 ▼
1 void setup() {
2   println("Ejercicio 15:");
3   float numero1 = 10;
4   float numero2 = 20;
5   float suma = sumarNumeros(numero1, numero2);
6   println("Suma: " + suma);
7   float resta = restarNumeros(numero1, numero2);
8   println("Resta: " + resta);
9   float multiplicacion = multiplicarNumeros(numero1, numero2);
10  println("Multiplicación: " + multiplicacion);
11  if (numero2 != 0) {
12    float division = dividirNumeros(numero1, numero2);
13    println("División: " + division);
14  } else {
15    println("No se puede dividir entre cero.");
16  }
17  noLoop();
18 }
19 float sumarNumeros(float a, float b) {
20   return a + b;
21 }
22 float restarNumeros(float a, float b) {
23   return a - b;
24 }
25 float multiplicarNumeros(float a, float b) {
26   return a * b;
27 }
28 float dividirNumeros(float a, float b) {
29   return a / b;
30 }
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
```

```
Ejercicio 15:
Suma: 30.0
Resta: -10.0
Multiplicación: 200.0
División: 0.5
```

Punto 16:

Ejercicio 16: Necesitamos convertir una temperatura Fahrenheit en grados Celsius. Si no conoce la forma en la que se realiza esta conversión, debería investigarlo; para eso sirve la etapa de análisis. Pero como somos buenos, daremos una ayuda.

Desarrollo del punto

Convertir una temperatura dada en grados Fahrenheit a grados Celsius.

Análisis

Datos de Entrada:

temperaturaFahrenheit: Real

Datos de Salida:

temperaturaCelsius: Real

Proceso:

Quién realiza el proceso: Algoritmo de conversión

Descripción del Proceso:

La conversión de temperatura de Fahrenheit a Celsius se realiza mediante la siguiente fórmula:

$$\text{temperaturaCelsius} = (\text{temperaturaFahrenheit} - 32) \times \frac{5}{9}$$

Se solicita al usuario ingresar la temperatura en grados Fahrenheit (temperaturaFahrenheit).

Se aplica la fórmula de conversión para calcular la temperatura equivalente en grados Celsius (temperaturaCelsius).

Se muestra el resultado de la conversión (temperaturaCelsius).

ENTIDAD QUE RESUELVE EL PROBLEMA: Conversor de temperatura

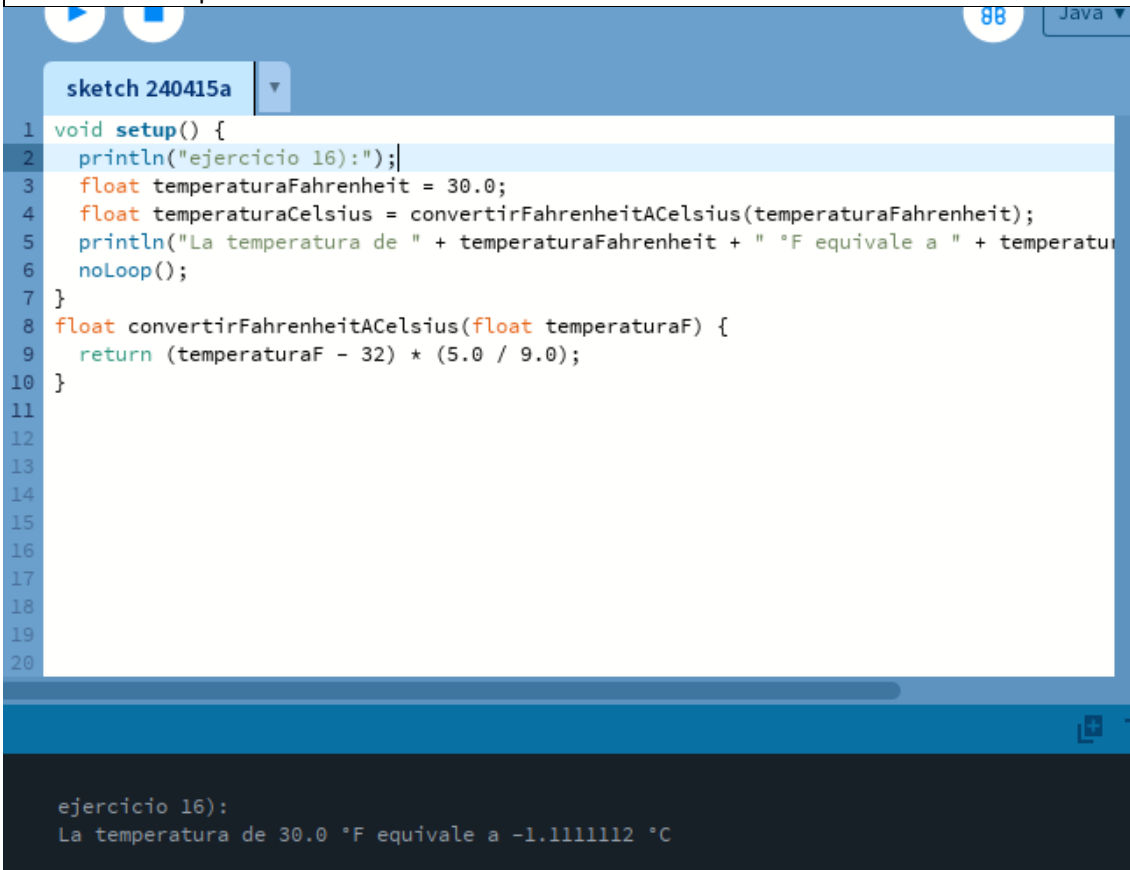
VARIABLES

tempFahrenheit, tempCelsius: Real // almacenan la temperatura en Fahrenheit y Celsius respectivamente

NOMBRE ALGORITMO: convertir_temperatura

PROCESO DEL ALGORITMO

1. Leer tempFahrenheit
2. $\text{tempCelsius} \leftarrow (\text{tempFahrenheit} - 32) \cdot (5/9)$ // fórmula de conversión
3. Mostrar tempCelsius



```
1 void setup() {
2   println("ejercicio 16):");
3   float temperaturaFahrenheit = 30.0;
4   float temperaturaCelsius = convertirFahrenheitACelsius(temperaturaFahrenheit);
5   println("La temperatura de " + temperaturaFahrenheit + " °F equivale a " + temperaturaCelsius);
6   noLoop();
7 }
8 float convertirFahrenheitACelsius(float temperaturaF) {
9   return (temperaturaF - 32) * (5.0 / 9.0);
10 }
11
12
13
14
15
16
17
18
19
20
```

ejercicio 16):
La temperatura de 30.0 °F equivale a -1.1111112 °C

Punto 17:

Ejercicio 17: Si queremos representar personajes o power ups (premios) en la pantalla debemos primero ubicarlos en alguna posición dentro de la pantalla. Imagine que está en un juego donde un power up desaparece porque el personaje se acerca a una distancia de x unidades, sin importar por donde se acerque. Por tanto, para que desaparezca, en primer lugar, hay que determinar esa distancia. La forma de representar la posición de un objeto en la pantalla es a través de las coordenadas de un punto. Suponga que la posición de Link está representada por la coordenada (x_1, y_1) , mientras que la de la caja de tesoro se halla en la posición (x_2, y_2) . Si observa con detenimiento se observa la conformación de un triángulo rectángulo, por lo que es posible aplicar Pitágoras para obtener la distancia. Para esto debe calcular el tamaño de los catetos y luego aplicar el teorema. Halle la distancia entre ambos objetos. Cuando programe, represente a Link con un Circulo, y al tesoro con un cuadrado. Además, mueva a Link mediante el mouse.

Desarrollo del punto

Datos de Entrada:

Coordenadas de Link:

x_1 : Real

y_1 : Real

Coordenadas del Tesoro:

x_2 : Real

y_2 : Real

Datos de Salida:

distancia: Real

Proceso:

Quién realiza el proceso: Algoritmo de juego

Descripción del Proceso:

Solicitar al usuario que mueva el mouse para actualizar la posición de Link (x_1, y_1) .

Calcular la distancia entre Link y el tesoro utilizando el teorema de Pitágoras: $\text{distancia} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ (fin de la raíz)



Entidad que resuelve el problema: Calculadora

Variables:

x1, y1: Real

x2, y2: Real

distancia: Real

Nombre del algoritmo: calcular_distancia

Proceso del algoritmo:

1. Leer x1, y1
2. Leer x2, y2
3. Calcular el tamaño del cateto horizontal: $\Delta x = |x2 - x1|$.
4. Calcular el tamaño del cateto vertical: $\Delta y = |y2 - y1|$.
5. Calcular la distancia usando el teorema de Pitágoras: $distancia = \sqrt{(\Delta x^2 + \Delta y^2)}$.
6. Mostrar distancia.

sketch 240415c

```
1 float x1, y1;
2 float x2, y2;
3
4 void setup() {
5   println("ejercicio 17:");
6   size(400, 400);
7   x1 = width / 2;
8   y1 = height / 2;
9   x2 = random(width);
10  y2 = random(height);
11 }
12
13 void draw() {
14   background(255);
15
16   fill(0, 0, 255);
17   ellipse(x1, y1, 30, 30);
18   fill(255, 0, 0);
19   rect(x2, y2, 30, 30);
20 }
```

```
ejercicio 17):
Distancia entre Link y el tesoro: 157.34404
Distancia entre Link y el tesoro: 157.34404
Distancia entre Link y el tesoro: 157.34404
```

Consola Errores



FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS
TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS
FACULTAD DE INGENIERÍA
Universidad Nacional de Jujuy
Trabajo Practico N° / Actividad



```
1 float x1, y1;
2 float x2, y2;
3
4 void setup() {
5   println("ejercicio 17:");
6   size(400, 400);
7   x1 = width / 2;
8   y1 = height / 2;
9   x2 = random(width);
10  y2 = random(height);
11 }
12
13 void draw() {
14   background(255);
15
16   fill(0, 0, 255);
17   ellipse(x1, y1, 30, 30);
18   fill(255, 0, 0);
19   rect(x2, y2, 30, 30);
20 }
```

Distancia entre Link y el tesoro: 159.21936
Distancia entre Link y el tesoro: 159.21936
Distancia entre Link y el tesoro: 159.21936

Punto 18:

Ejercicio 18: Desarrolle el análisis y diseño de un algoritmo que permita obtener las raíces de una ecuación de segundo grado. Además, utilice la estructura según para el análisis de la discriminante de la ecuación cuadrática. Obviamente codifique en Processing.

Desarrollo del punto

Descripción del Problema:

Se desea desarrollar un algoritmo que permita calcular las raíces de una ecuación de segundo grado de la forma $ax^2+bx+c=0$, donde a, b y c son coeficientes reales y a (no igual) 0.

Datos de Entrada:

a: Real

b: Real

c: Real

Datos de Salida:

raiz1: Real

raiz2: Real

Proceso:

Calcular el discriminante de la ecuación cuadrática y luego determinar el tipo de raíces (reales o complejas) según el valor de Δ . Posteriormente, se utilizan fórmulas específicas para calcular las raíces en cada caso.

Este proceso te permitirá resolver cualquier ecuación cuadrática y determinar las raíces correspondientes dependiendo de las características del discriminante.

Entidad que resuelve el problema: Calculadora
Variables: a, b, c: Real discriminante, raiz1, raiz2: Real
Nombre del algoritmo: calcular_raices Proceso del algoritmo: 1. Leer a, b, c // Coeficientes de la ecuación cuadrática. 2. Calcular el discriminante: $\text{discriminante} = b^2 - 4ac$. 3. Según el valor del discriminante: - Si $\text{discriminante} > 0$: - Calcular las raíces reales: $\text{raiz1} = (-b + \sqrt{\text{discriminante}}) / (2a)$ y $\text{raiz2} = (-b - \sqrt{\text{discriminante}}) / (2a)$. - Mostrar raiz1 y raiz2 como las raíces reales de la ecuación. - Si $\text{discriminante} = 0$: - Calcular la raíz real doble: $\text{raiz1} = \text{raiz2} = -b / (2a)$. - Mostrar raiz1 y raiz2 como la raíz real doble de la ecuación.

- Si discriminante < 0 :
- Mostrar un mensaje indicando que las raíces son complejas y no se pueden calcular en este algoritmo.

```
void setup() {  
    println("ejercicio 18:");  
    float a = 1.0;  
    float b = -3.0;  
    float c = 2.0;  
    float discriminante = b*b - 4*a*c;  
  
    if (discriminante > 0) {  
        float raiz1 = (-b + sqrt(discriminante)) / (2*a);  
        float raiz2 = (-b - sqrt(discriminante)) / (2*a);  
        println("Raíces reales distintas:");  
        println("Raíz 1: " + raiz1);  
        println("Raíz 2: " + raiz2);  
    } else if (discriminante == 0) {  
        float raiz = -b / (2*a);  
        println("Raíz real doble:");  
        println("Raíz: " + raiz);  
    } else {  
        float parteReal = -b / (2*a);  
    }  
}
```

```
ejercicio 18):  
Raíces reales distintas:  
Raíz 1: 2.0  
Raíz 2: 1.0
```

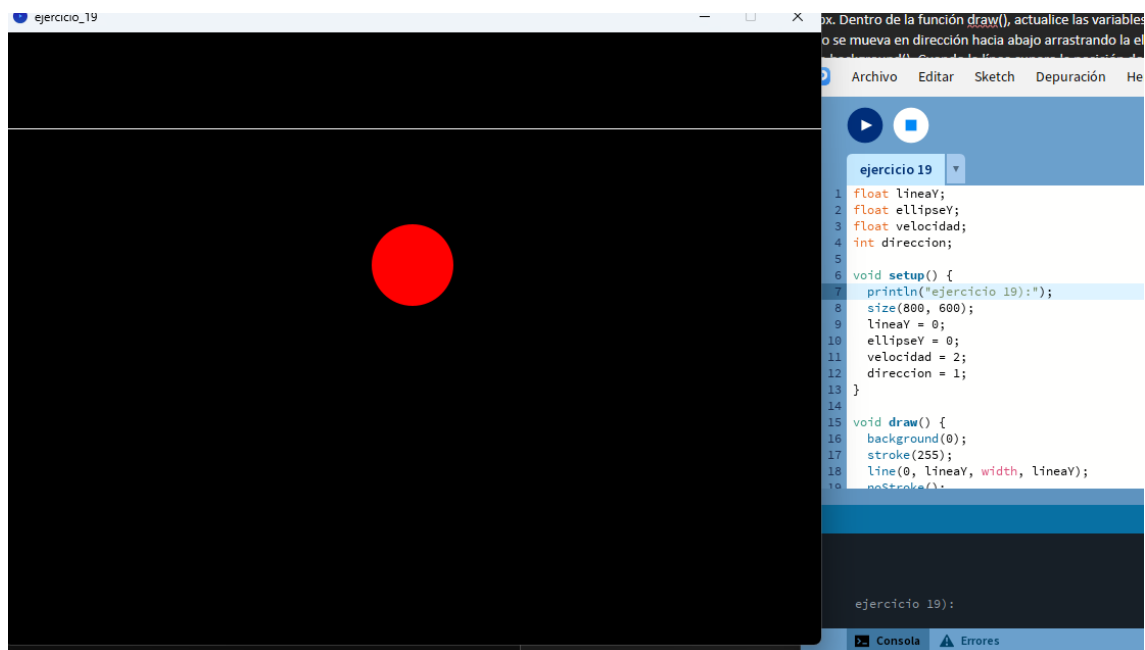
Consola Errores

Punto 19:

Ejercicio 19: Declare las variables necesarias para dibujar una línea que se dibuja desde las coordenadas iniciales del lienzo y se extiende por todo el ancho. Sobre el punto medio de la línea y a una distancia de 40px (en sentido vertical desde la línea) dibuje una elipse que tenga como ancho 80px y de alto 80px. Dentro de la función draw(), actualice las variables necesarias para que la línea desde su inicio se mueva en dirección hacia abajo arrastrando la elipse.

Mantenga en cero el valor para background(). Cuando la línea supere la posición de la altura del lienzo, debe invertir su sentido, es decir dirigirse hacia arriba arrastrando la elipse. Cuando la línea alcance nuevamente el valor 0 para su posición en y, el desplazamiento debe ser hacia abajo y así sucesivamente. El lienzo debería verse como en las siguientes figuras.

Desarrollo del punto



Entidad que resuelve el problema: Dibujador de línea con elipse

Variables:

xLinea: Real

yLinea: Real

yEllipse: Real

direccion: Booleano



Nombre del algoritmo: dibujarLineaConElipse

Proceso del algoritmo:

1. Inicializar xLinea
2. Inicializar yLinea
3. Inicializar yEllipse
4. Inicializar direccion

Dentro de la función draw():

1. Limpiar el lienzo con background(0).
2. Dibujar una línea horizontal que se extiende por todo el ancho del lienzo desde (xLinea, yLinea) hasta (width, yLinea).
3. Dibujar una elipse en el punto medio de la línea, en las coordenadas (xLinea + (width - xLinea) / 2, yEllipse), con ancho 80 y alto 80.
4. Actualizar la posición de yLinea y yEllipse para mover la línea y la elipse hacia abajo o hacia arriba, dependiendo de la dirección actual.
5. Verificar si la línea ha superado los límites del lienzo (height). Si es así, invertir la dirección de movimiento.
6. Repetir este proceso en cada iteración de draw() para crear la animación.

	<p>FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS FACULTAD DE INGENIERÍA Universidad Nacional de Jujuy Trabajo Practico N° / Actividad</p>	
---	--	---

Punto 20:

Ejercicio 20: Dibuje en toda la extensión del lienzo de (440, 420) rectángulos de idénticas medidas (40 ancho y 20 de alto) y que mantengan una distancia de 20 pixeles entre ellos tanto horizontal como verticalmente. Utilice la estructura de control repetitiva for. El lienzo debería verse así:

Desarrollo del punto

Análisis

Datos de Entrada:

Dimensiones del lienzo: Ancho de 440 pixeles y alto de 420 pixeles.

Medidas del rectángulo: Ancho de 40 pixeles y alto de 20 pixeles.

Espacio entre rectángulos: 20 pixeles tanto horizontal como verticalmente.

Datos de Salida:

Un lienzo que muestra rectángulos de tamaño 40x20 pixeles, distribuidos uniformemente con un espacio de 20 pixeles entre ellos tanto horizontal como verticalmente.

Proceso:

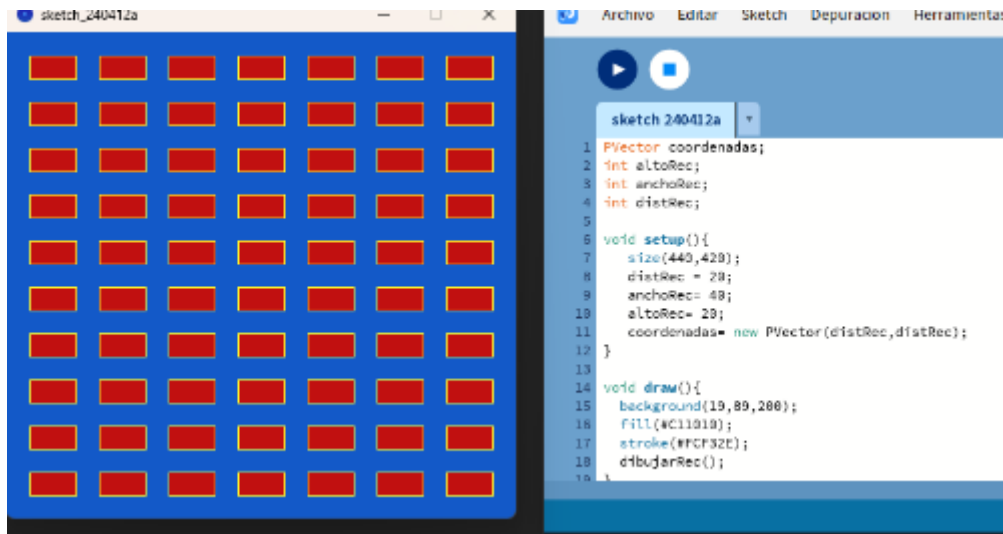
Utilizaremos una estructura de control repetitiva for para iterar sobre las filas y columnas del lienzo y calcular las posiciones de cada rectángulo.

En cada iteración:

Calcular la posición x y y del rectángulo utilizando las dimensiones del rectángulo y el espacio entre ellos.

Dibujar un rectángulo en la posición calculada.

ENTIDAD QUE RESUELVE EL PROBLEMA: Dibujador de Rectángulos
<p>VARIABLES</p> <p>anchoLienzo: Real // Ancho del lienzo en pixeles</p> <p>altoLienzo: Real // Alto del lienzo en pixeles</p> <p>anchoRect: Real // Ancho de los rectángulos en pixeles</p> <p>altoRect: Real // Alto de los rectángulos en pixeles</p> <p>espacioEntreRect: Real // Espacio entre los rectángulos en pixeles</p>
<p>NOMBRE ALGORITMO: Dibujar_Rectangulos</p> <p>PROCESO DEL ALGORITMO</p> <ol style="list-style-type: none"> 1. Leer anchoLienzo 2. Leer altoLienzo 3. Leer anchoRect 4. Leer altoRect 5. Leer espacioEntreRect 6. Configurar el lienzo con dimensiones (anchoLienzo, altoLienzo) 7. Establecer el color de fondo del lienzo en blanco 8. Iterar sobre las filas del lienzo utilizando la estructura de control repetitiva for: <p>Iniciar en la posición fila = 0</p> <p>Incrementar fila en altoRect + espacioEntreRect en cada iteración hasta que fila sea menor que altoLienzo</p> 9. Dentro del bucle de filas, iterar sobre las columnas utilizando otra estructura for: <p>Iniciar en la posición col = 0</p> <p>Incrementar col en anchoRect + espacioEntreRect en cada iteración hasta que col sea menor que anchoLienzo</p> 10. Dentro de cada iteración de filas y columnas: <p>Calcular la posición del rectángulo en (col, fila)</p> <p>Dibujar un rectángulo en la posición calculada utilizando las dimensiones anchoRect y altoRect</p> <p>Establecer el color del rectángulo en negro (fill(0)) y sin contorno (noStroke())</p>



Punto 21:

Ejercicio 21: Utilizando la estructura de control repetitiva while() dibuje la siguiente imagen utilizando líneas que forman escalones y sobre cada borde de escalón se dibuje un punto de color rojo El tamaño del lienzo es size(500,500). La estructura while() se ejecuta dentro de la función setup(). La condición es que solo se dibuje dentro del lienzo. Utilice variables que puedan ayudar a la construcción del dibujo, por ej: x, y, anchoEscalon, altoEscalon, etc.

Desarrollo del punto

Análisis

Datos de Entrada:

Tamaño del lienzo: Ancho y alto de 500x500 píxeles.

Dimensiones de los escalones:

anchoEscalon: 60 píxeles

altoEscalon: 30 píxeles

Datos de Salida:

Una imagen en forma de escalones dibujados dentro del lienzo, donde cada borde de escalón tiene un punto rojo.

Proceso:

Utilizaremos la estructura de control repetitiva `while()` para dibujar líneas que formen escalones y puntos rojos en cada borde de escalón.

El bucle `while()` se ejecutará dentro de la función `setup()` para asegurar que el dibujo se realice al iniciar el programa y solo dentro del área del lienzo.

Utilizaremos variables como `x` y `y` para rastrear la posición actual mientras se dibujan los escalones y los puntos.

Dibujaremos líneas horizontales para formar escalones de forma ascendente en el eje Y.

En cada borde de escalón, dibujaremos un punto rojo utilizando la función `point()`.

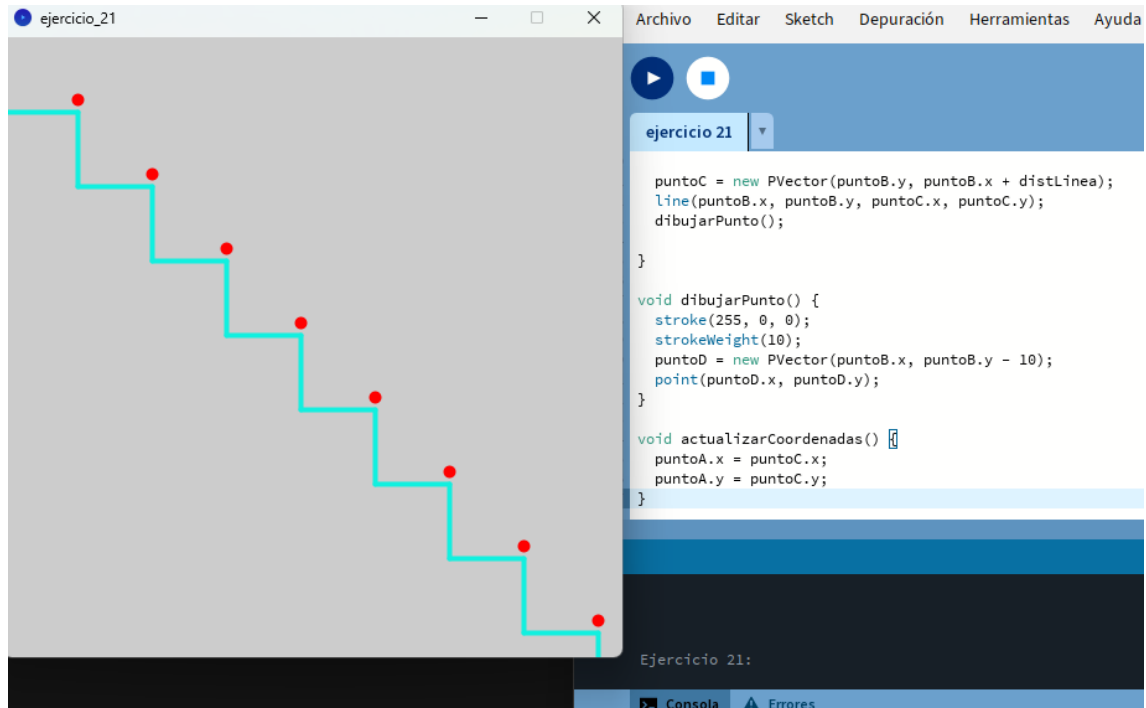
Entidad que Resuelve el Problema: Dibujador de Escalones
Variables x: Real y: Real anchoEscalon: Real altoEscalon: Real dibujarPunto: Booleano
NOMBRE ALGORITMO: Dibujar_Escalones Proceso del algoritmo: 1. Configuración del Lienzo: Se utiliza la función <code>size()</code> para crear un lienzo de 500x500 píxeles. Se inicializan las variables <code>x</code> , <code>y</code> , <code>anchoEscalon</code> y <code>altoEscalon</code> con valores adecuados para la construcción del dibujo. 2. Dibujo de Escalones con Puntos Rojos: Utilizando una estructura <code>while()</code> , se dibujan escalones de líneas horizontales en el lienzo. En cada iteración, se verifica si se debe dibujar un punto rojo en el borde del escalón (<code>dibujarPunto</code> es verdadero).

Se controla la posición y el tamaño de cada escalón con las variables x , y , anchoEscalon y altoEscalon.

Se actualizan las variables x e y para pasar al siguiente escalón.

3. Condición de Finalización:

La estructura `while()` se ejecuta mientras las coordenadas y no superen la altura del lienzo ($y \leq \text{height}$).



Punto 22:

Ejercicio 22: Utilizando la estructura de control repetitiva do-while. Replique la siguiente imagen La imagen debe ser construida desde la función setup(). Defina el tamaño del lienzo en size(600,600), verticalmente se divide el lienzo en franjas de igual medida, se deben dibujar los círculos sobre cada línea de por medio es decir en la línea 1 se dibujan círculos con distanciamiento, en la línea 2 no se dibuja y así sucesivamente. Las líneas tienen un color fijo, los círculos asumen colores aleatorios.

Desarrollo del punto

Análisis

Datos de Entrada:

anchoLienzo: Real

altoLienzo: Real

Datos de Salida:

Visualización de círculos dibujados en el lienzo según el patrón especificado.

Proceso del Algoritmo:

Establecer el tamaño del lienzo utilizando size(anchoLienzo, altoLienzo).

Dividir verticalmente el lienzo en franjas de igual medida.

Utilizar la estructura do-while para dibujar los círculos siguiendo un patrón de franjas alternas.

En cada iteración, dibujar círculos sobre las franjas impares y omitir las franjas pares.

Asignar colores aleatorios a los círculos.

ENTIDAD QUE RESUELVE EL PROBLEMA: Dibujador de Círculos en Franjas alternas
VARIABLES: anchoLienzo: Real. altoLienzo: Real. numFranjas: Entero. altoFranja: Real. y: Real. dibujarCirculos: Booleano
NOMBRE ALGORITMO: Dibujar_Circulos_Franjas_Alternas PROCESO DEL ALGORITMO: <ol style="list-style-type: none"> 1. Definir las dimensiones del lienzo utilizando size(anchoLienzo, altoLienzo). 2. Inicializar el fondo del lienzo con background(255) para un fondo blanco. 3. Calcular el altoFranja dividiendo altoLienzo por numFranjas.

4. Inicializar la posición vertical y en 0.
5. Utilizar la estructura do-while para iterar sobre las franjas del lienzo.
6. En cada iteración:

Verificar si se debe dibujar círculos en la franja actual (dibujarCirculos es verdadero).

Si se deben dibujar círculos:

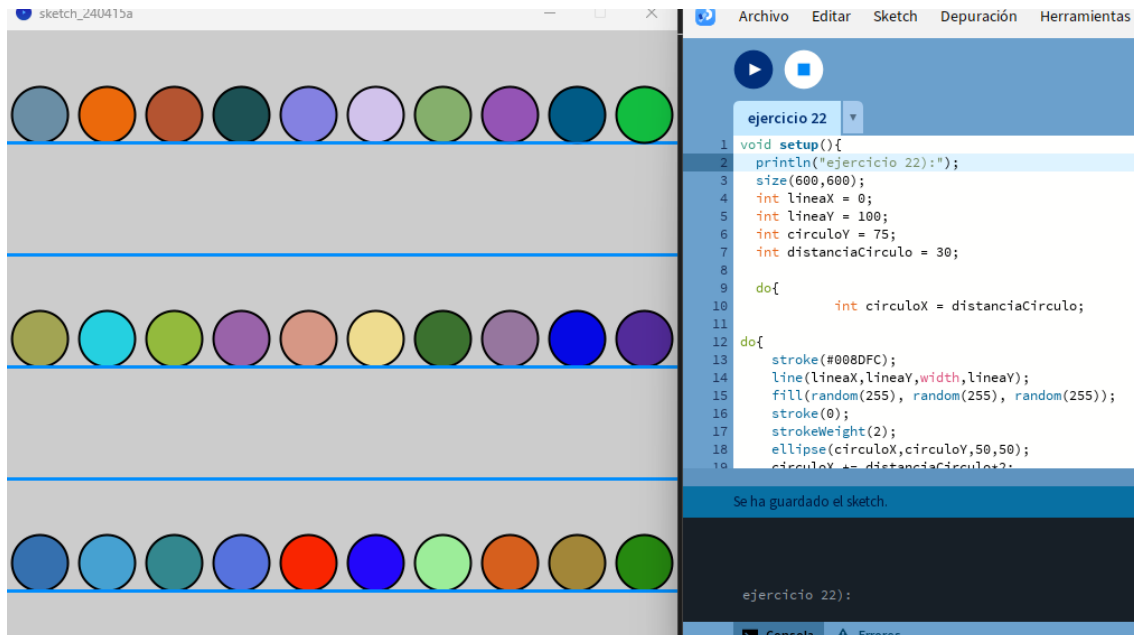
Iterar horizontalmente sobre la franja y dibujar círculos en posiciones específicas.

Asignar un color aleatorio a cada círculo utilizando fill(random(255), random(255), random(255)).

Actualizar la posición vertical y para la siguiente franja.

Alternar la variable dibujarCirculos para cambiar entre franjas impares y pares.

7. Continuar el bucle do-while hasta que se haya completado el dibujo de todas las franjas verticales del lienzo.



Conclusión

Párrafos de las conclusiones

Fuentes bibliográficas

Se deben enunciar las fuentes (apuntes de la materia, páginas web, videos de youtube, libro (nombre, autores, año), etc)

El unico que me lei fue de Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction" (Segunda Edición 2015) por Daniel Shiffman en español.





FUNDAMENTOS DE PROGRAMACIÓN ORIENTADA A OBJETOS
TECNICATURA UNIVERSITARIA EN DISEÑO INTEGRAL DE VIDEOJUEGOS

FACULTAD DE INGENIERÍA
Universidad Nacional de Jujuy
Trabajo Practico N° / Actividad

