



PROYECTO CINERD

Informe final programación

Presentado por:


Matías Barra
Consuelo Ardiles
Edinson Godoy



ÍNDICE

CONTENIDO DEL INFORME

Introduccion y propósito del proyecto.	1
Diagrama UML inicial	2
Diseño de interfaces	3
Características del sistema.	7
Módulo conexión	9
Módulo Login	10
Módulo Cartelera	12
Módulo Seleccion de asientos	13
Módulo Método de pago	14
Módulo Factura	15
Descripción fragmento código	16
Diagrama UML final	20
Comparativa Diagrama UML	21





INTRODUCCIÓN A POO

La programación orientada a objetos es fundamental porque permite la representación de entidades del mundo real como objetos, facilitando la modularidad, reutilización y mantenimiento del código. Esta metodología fomenta la abstracción, encapsulamiento, herencia y polimorfismo, permitiendo un diseño más organizado, escalable y comprensible de sistemas complejos, lo que resulta vital para el desarrollo eficiente y sostenible de software.

Bajo este punto de vista podemos tener la capacidad e iniciativa de querer resolver o mejorar problemáticas de la vida cotidiana de los diversos usuarios que utilizan herramientas a base de software. Lo que nos lleva a nuestro proyecto “Cinerd”.

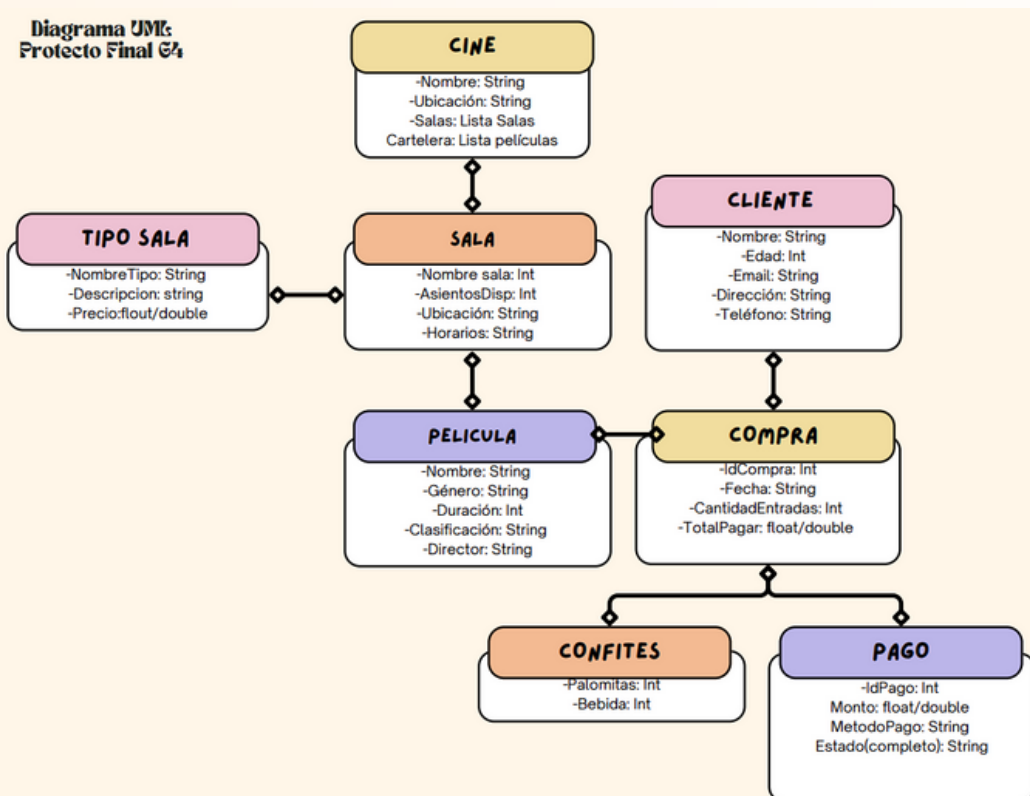
PROPÓSITO DEL PROYECTO

Nuestro proyecto “Cinerd” consiste en un software que permite interactuar con la interfaz de un Cine de forma online. La idea principal es brindar una solución eficiente y accesible para la gestión de información sobre películas, horarios y disponibilidad de asientos. Este tipo de sistema busca proporcionar a los usuarios una plataforma conveniente y fácil de usar para conocer la oferta cinematográfica, reservar asientos y realizar compras de entradas de manera rápida y efectiva, mejorando así la experiencia del cliente y fomentando la fidelidad a través de una interfaz intuitiva y servicios confiables. Cabe destacar que la implementación de los objetivos y visión final del proyecto cambiaron a lo largo del transcurso del curso, tales cambios se mencionarán a su debido momento.

POTENCIALES USUARIOS

Personas interesadas en ver películas y que utilizan la plataforma para explorar la cartelera, seleccionar películas, revisar horarios y comprar entradas de forma conveniente. Se espera que sean mayores de edad para utilizar la plataforma, ya que esta necesita datos bancarios de ser online su utilización.

DIAGRAMA UML INICIAL



Este diagrama UML inicial muestra las aspiraciones del equipo de trabajo con respecto a la creación del proyecto “Cinerd”. A manera de poder organizar o prever las ideas que se pueden agregar al proyecto. Los atributos añadidos son nombres estandar que están sujetos a cambios.

Se tuvo la idea inicial de crear un login de administrador y uno de cliente, pero eso conllevaba a ampliar demasiado el espectro del código y sus funciones, por lo que se optó por solo aplicar el modulo cliente y sus derivados.

DISEÑO DE INTERFACES

Aquí mostraremos los diseños iniciales de las interfaces que verá el usuario a lo largo de la plataforma “Cinerd”



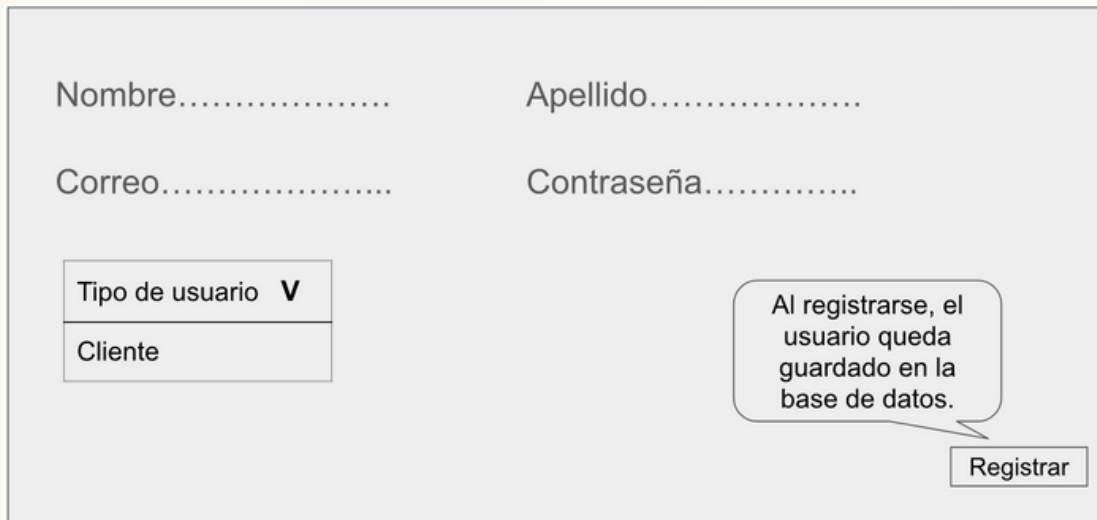
Logo Cine

Registro

Ingreso vía correo

Contraseña

Ingresar



Nombre..... Apellido.....

Correo..... Contraseña.....

Tipo de usuario **V**

Cliente

Al registrarse, el usuario queda guardado en la base de datos.

Registrar

DISEÑO DE INTERFACES

Aquí mostraremos los diseños iniciales de las interfaces que verá el usuario a lo largo de la plataforma “Cinerd”



Logo Cine

Registro

Ingreso vía correo

Contraseña

Ingresar



logo cine

Películas

Horarios

Tipo de sala

Pelicula 1

Horario 1

Horario 2

Horario 3

Sala V

2D

3D

3DXD

Pelicula 2

Horario 1

Horario 1

Horario 1

Sala V

2D

3D

3DXD

Seleccionar Asientos

DISEÑO DE INTERFACES

Aquí mostraremos los diseños iniciales de las interfaces que verá el usuario a lo largo de la plataforma “Cinerd”



Diagrama de la interfaz de usuario para la selección de películas. La pantalla muestra una grilla de 12 películas (4 filas por 3 columnas). Las películas están representadas por iconos de películas. Las películas seleccionadas están resaltadas en verde, y las reservadas están en gris. La leyenda indica: Reservados (gris) y Seleccionados (verde). En la parte inferior, hay un campo para el TOTAL (-----) y un botón "Ir a pagar".

Pantalla

Reservados Seleccionados

TOTAL: ----- Ir a pagar

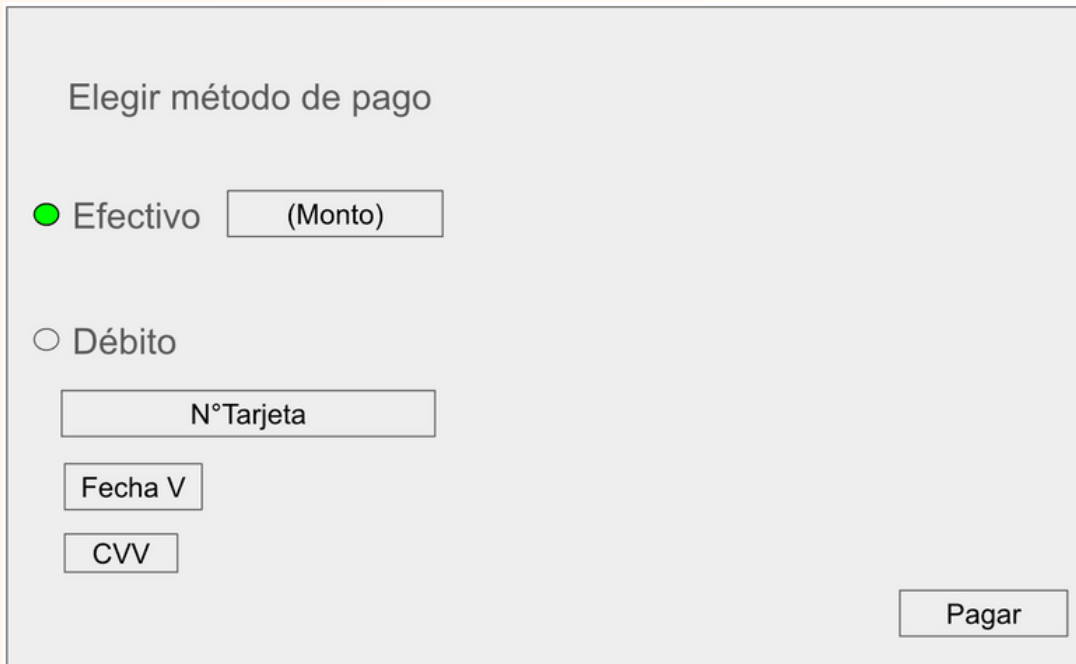


Diagrama de la interfaz de usuario para elegir el método de pago. La pantalla muestra el título "Elegir método de pago". Hay dos opciones: "Efectivo" (seleccionada) y "Débito". La opción "Efectivo" tiene un campo "Monto". La opción "Débito" tiene campos para "N°Tarjeta", "Fecha V" y "CVV". Un botón "Pagar" está ubicado en la parte inferior derecha.

Elegir método de pago

☒ Efectivo (Monto)

☐ Débito

N°Tarjeta

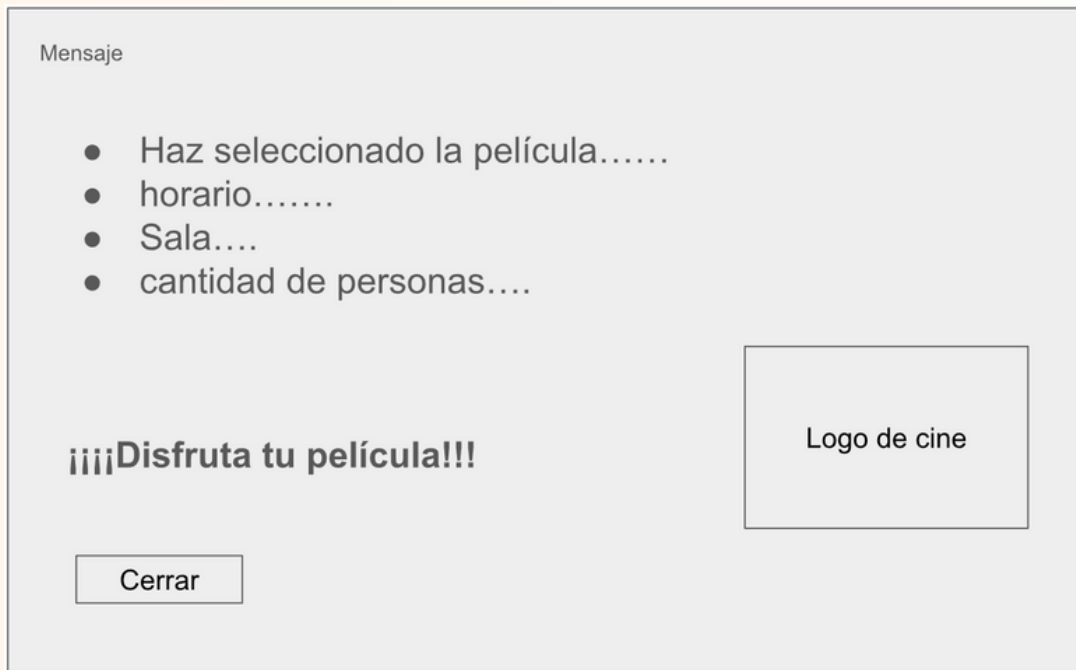
Fecha V

CVV

Pagar

DISEÑO DE INTERFACES

Aquí mostraremos los diseños iniciales de las interfaces que verá el usuario a lo largo de la plataforma “Cinerd”



Como se puede apreciar, las interfaces son bastante simples e intuitivas para el uso general del público. Y que posteriormente su modificación general se centró mas en el diseño estético mas que en su funcionamiento que parece ser correcto.

CARACTERÍSTICAS DEL SISTEMA

El entorno de desarrollo utilizado para este proyecto es Netbeans junto con la aplicación Xampp para poder inicializar una base de datos en Mysql de manera local. La cual es utilizada para almacenar los usuarios registrados (junto con sus datos derivados), los filmes disponibles en la plataforma y los productos adquiridos con tal de tener la información respectiva del consumo en la plataforma.

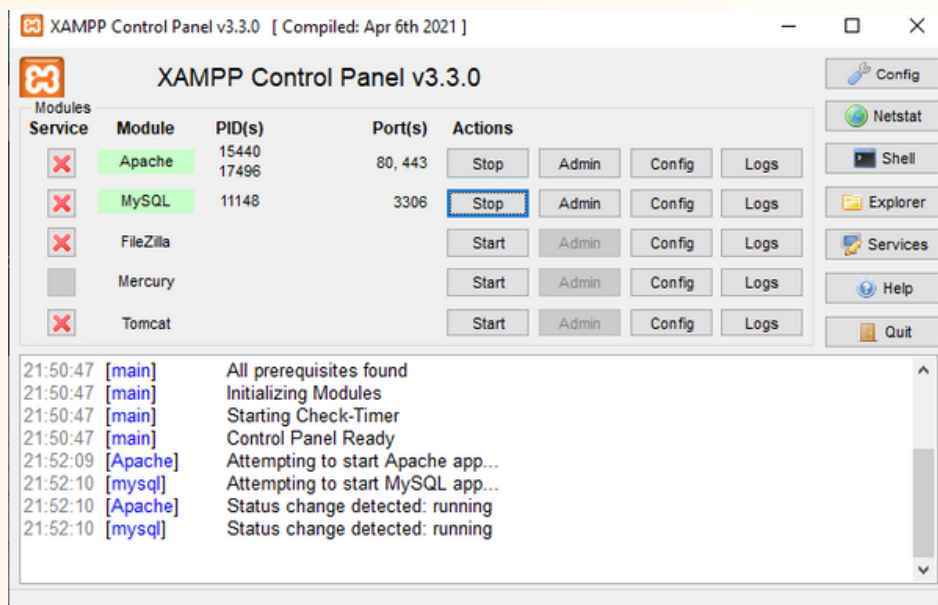


Netbeans



Xampp

El entorno de desarrollo de Xampp es así:



En donde es necesario inicializar los modulos de Mysql para la base de datos y Apache para poder vincularla con Netbeans.

CARACTERÍSTICAS DEL SISTEMA

Al hacer click en Admin en el modulo Mysql, se puede entrar al sitio de phpMyAdmin en donde podremos crear e inicializar una base de datos creando una tabla de datos llamada “cinerdproject”

ID	nombre	apellido	correoelectronico	pass	tipousuario
9	Matias	Barra	matias barra@userena.cl	246	Cliente
10	Matias	Barra	Matias@hotmail.com	12345	Cliente
11	matias	Barra	barra@hotmail.com	246	Cliente
12	Valentina	Barra	vale2005@gmail.com	123456	Cliente
13	Gonzalo	Honores	ghonores@userena.cl	qwerty	Cliente

Se pueden apreciar los usuarios registrados, así como la cartelera del sistema

pelicula	horario	sala
[value-1]	[value-2]	[value-3]
Película 3 (Aquaman)	14:00	2D
Película 1 (Napoleon)	15:00	3D
Película 1 (Napoleon)	21:00	2D
Película 6 (Whish)	17:20	3D
Película 4 (Noche de paz)	13:00	3D
Película 4 (Noche de paz)	16:00	2D
Película 2 (Wonka)	13:00	3D
Película 4 (Noche de paz)	16:00	3D
Película 6 (Whish)	12:40	3D
Película 5 (Teddy)	16:30	3D
Película 3 (Aquaman)	17:00	2D

Cabe destacar que la base de datos creada en Mysql debe tener el mismo nombre del proyecto que de lo contrario no podrá ser vinculada a este y al momento de registrar los datos e iniciar sesión en el sistema, arrojará un error de no conexión.

CARACTERÍSTICAS DEL SISTEMA

MÓDULO - CONEXIÓN

Para poder vincular esta base de datos creada de forma local se necesita tener un modulo de conexión en Netbeans el cual permita tener dicha interacción.

```
package Conexion;

import com.mysql.jdbc.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class Conex {
    Connection cn;

    public Connection ConectarBD() {
        try{
            Class.forName(className: "com.mysql.jdbc.Driver");
            cn=(Connection) DriverManager.getConnection(url: "jdbc:mysql://localhost/CinerdProyect", user: "root", password: "");
            System.out.println("conexion exitosa");
        }catch (ClassNotFoundException | SQLException e) {
            System.out.println("conexion fallida"+e);
        }
        return cn;
    }
}
```

Nuestro modulo se llama “conexión” y es necesario para poder trabajar con los demas modulos que se mencionarán a continuación

- Módulo Login
- Modulo Cartelera
- Módulo Seleccion de Asientos
- Módulo Registro
- Módulo Método de Pago
- Módulo Factura

CARACTERÍSTICAS DEL SISTEMA

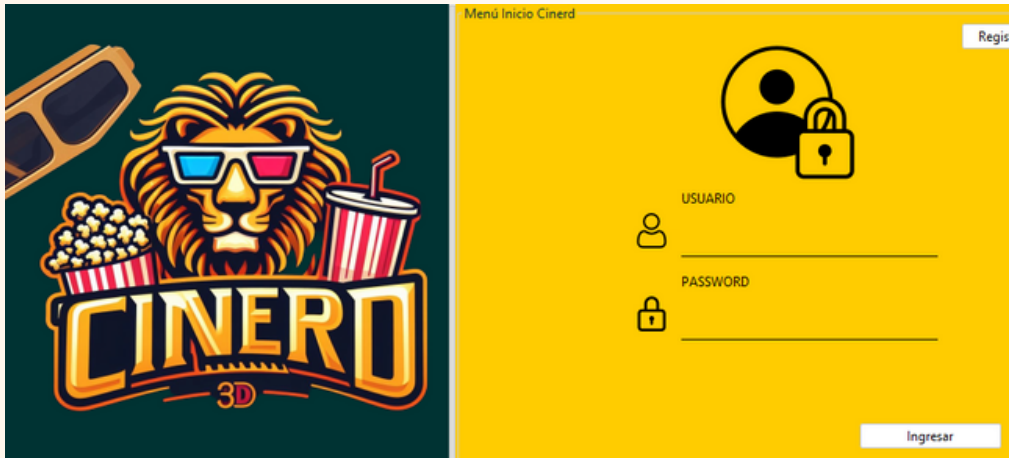
MÓDULO - LOGIN

Permite la interacción grafica con el usuario, en donde este deberá escribir sus datos para poder ingresar a la plataforma. Su interfaz debe ser llamativa, intuitiva y eficiente.

```
1  import Conexion.Conexion;
2  import com.mysql.jdbc.PreparedStatement;
3  import java.awt.HeadlessException;
4  import java.sql.Connection;
5  import java.sql.ResultSet;
6  import java.sql.SQLException;
7  import javax.swing.JOptionPane;
8  public class login extends javax.swing.JFrame {
9      Conexion.Conexion con=new Conexion();
10     Connection cn=con.ConectarBD();
11     public static Registro fr;
12     public login() {
13         initComponents();
14         setLocationRelativeTo(null); //para que aparezca centrado
15     }
16
17     /**
18      * This method is called from within the constructor to initialize the form.
19      * WARNING: Do NOT modify this code. The content of this method is always
20      * regenerated by the Form Editor.
21      */
22     @SuppressWarnings("unchecked")
23     Generated Code
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
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
261
```

CARACTERÍSTICAS DEL SISTEMA

MÓDULO - LOGIN



La interfaz graficamente es muy llamativa y de facil manipulaci3n. Clickeando el bot3n de “registro” podremos ingresar nuestros datos para poder vincularlos con la base de datos en Mysql.

Nombre	Apellido
Edinson	Godoy
Correo electronico	Password
edinson.godoyn@gmail.com	*****
Tipo de usuario	
Cliente	
Seleccionar	
Cliente	
Registrar Usuario	

Al clickear el bot3n produce un evento que abre una ventana en donde podremos llenar el formulario. Cabe destacar que la intenci3n inicial del proyecto era poder hacer 2 tipos de usuarios diferentes: “cliente” y “administrador” pero por motivos de eficiencia se opt3 por quitar ese modulo extra.

Esto llevaba a la creaci3n de modulos extras (edici3n de horarios, filmes, sinopsis, edici3n de usuarios plataforma, etc)

Obviamente el codigo quedaba mas complejo pero su funcionalidad quedaba perjudicada.

El principal controlador de evento para el módulo Login, está en la parte del código que se muestra a continuación:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    String usuario=txtusuario.getText();  
    String pass=txtpassword.getText();  
  
    if(!usuario.equals("")||!pass.equals("")){  
        try {  
            PreparedStatement ps;  
            ps = (PreparedStatement) cn.prepareStatement("SELECT tipousuario FROM usuarios WHERE correoelectronico='".  
            ResultSet rs=ps.executeQuery();  
            if(rs.next()){  
  
                String tipousuario=rs.getString(columnLabel: "tipousuario");  
                if(tipousuario.equalsIgnoreCase("Cliente")){  
                    dispose();  
                    Cartelera pa=new Cartelera();  
                    pa.setVisible(b: true);  
                }  
            }  
        }else{  
            JOptionPane.showMessageDialog(parentComponent: null, message: "USUARIO O CONTRASEÑA INCORRECTOS");  
        }  
    }catch (HeadlessException | SQLException e) {  
        JOptionPane.showMessageDialog(parentComponent: null, "ERROR AL INICIAR SESION"+e);  
    }  
    }else{  
        JOptionPane.showMessageDialog(parentComponent: null, message: "DEBE COMPLETAR LOS CAMPOS");  
    }  
}
```

En este evento se activa al pulsar el botón “Ingresar”, el cual verifica mediante a un try-Catch si el usuario y la contraseña están dentro de la base de datos, si esta consulta funciona correctamente, se inicia sesión y se ingresa a la cartelera para posteriormente reservar la película.

De lo contrario, aparecen mensajes de error dependiendo del error, Si la consulta no devuelve resultados, se muestra un mensaje indicando que el usuario o la contraseña son incorrectos.

Otro error puede ser problemas directos con SQL, si sucede eso, el mensaje será de error al iniciar sesión.

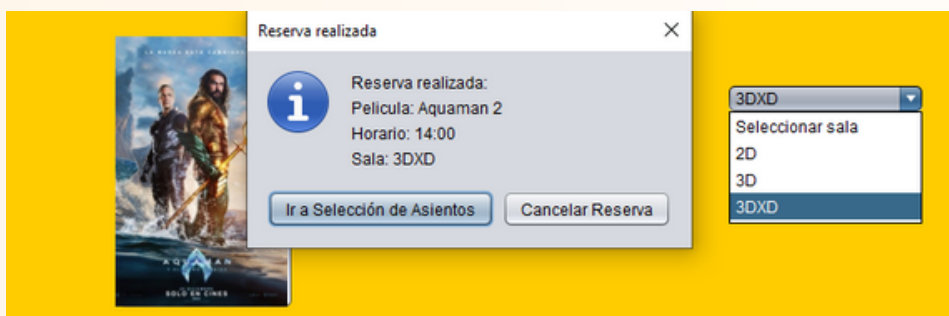
Por último se activó el evento con los campos vacío, este error recibirá un mensaje de “Debe completar campos”.

CARACTERÍSTICAS DEL SISTEMA

MÓDULO - CARTELERA

```
1 import Conexion.Conex;
2 import javax.swing.JButton;
3 import javax.swing.JOptionPane;
4 import java.sql.Connection;
5 import java.sql.PreparedStatement;
6 import java.sql.SQLException;
7
8 public class Cartelera extends javax.swing.JDialog {
9     private String peliculaSeleccionada;
10    private String horarioSeleccionado;
11    private String salaSeleccionada;
12    private JButton ultimoBotonHorarioSeleccionado;
13
14    public Cartelera(java.awt.Frame parent, boolean modal) {
15        super(owner: parent, modal);
16        initComponents();
17        bloquearControlesIniciales();
18    }
19
20    public Cartelera() {
21        super(new javax.swing.JFrame(), modal: true);
22        initComponents();
23        bloquearControlesIniciales();
24    }
25
26    private void bloquearControlesIniciales() {
27        JComboSeleccion1.setEnabled(b: false);
28        JComboSeleccion2.setEnabled(b: false);
29    }
30
31 }
```

El módulo cartelera fue implementado de manera similar al anterior, incluyendo las librerías de conexión a Mysql, aplicando arrastre para el diseño y luego en el código dando funcionalidad a los eventos y botones pertinentes.



En la cartelera podremos elegir el filme que se quiera reservar, junto con el tipo de sala y horario a elección. Cabe destacar que se incluyó un botón de sinopsis para cada filme que permite dar mas contexto sobre que filme puede gustar mas.

En este módulo existen algunos controladores para eventos como por ejemplo este controlador que maneja eventos relacionados con la selección de películas. Dependiendo de la película seleccionada, realiza acciones específicas asociadas a esa película, como relacionarla con ciertos elementos de la interfaz de usuario y bloquear horarios no relacionados.

```
private void JpeliculaActionPerformed(java.awt.event.ActionEvent evt) {  
    System.out.println("Evento generado por: " + evt.getSource());  
  
    desbloquearControlesPelículas();  
  
    peliculaSeleccionada = (String) ((JButton) evt.getSource()).getText();  
  
    switch (peliculaSeleccionada) {  
        case "Napoleon":  
            relacionarConPelicula1();  
            break;  
        case "Wonka":  
            relacionarConPelicula2();  
            break;  
        case "Aquaman 2":  
            relacionarConPelicula3();  
            break;  
        case "Noche de paz ":  
            relacionarConPelicula4();  
            break;  
        case "Teddy ":  
            relacionarConPelicula5();  
            break;  
        case "Whish ":  
            relacionarConPelicula6();  
            break;  
        default:  
            break;  
    }  
  
    bloquearHorariosNoRelacionados();  
}
```

Uno de los controladores de eventos en el módulo cartelera, es un Try-Catch al momento de guardar la reserva en la base de datos, se consulta la inserción de la película, el horario y la sala en una tabla SQL, si en algún momento existe una excepción, se activa el catch y se envía un mensaje de error de guardado de reserva.

```
private void guardarReservaEnBaseDeDatos() {  
    // Guardar reserva en la base de datos  
    try (Connection conex = new Conex().ConectarBD()) {  
        conex.setAutoCommit(true);  
        String sql;  
        sql = "INSERT INTO reserva (pelicula, horario, sala) VALUES (?, ?, ?)";  
        try (PreparedStatement declaracion = conex.prepareStatement(sql)) {  
            declaracion.setString(parameterIndex: 1, x: peliculaSeleccionada);  
            declaracion.setString(parameterIndex: 2, x: horarioSeleccionado);  
            declaracion.setString(parameterIndex: 3, x: salaSeleccionada);  
            declaracion.executeUpdate();  
        }  
    } catch (SQLException ex) {  
        // Imprimir el error  
        JOptionPane.showMessageDialog(parentComponent: this, message: "Error al guardar la reserva en la base de datos.", title: "Error")  
    }  
}
```


CARACTERÍSTICAS DEL SISTEMA

MÓDULO - SELECCIÓN DE ASIENTOS

```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5 import java.util.ArrayList;
6 import java.util.List;
7 import java.util.Random;
8
9 public class SeleccionDeAsientos extends JFrame {
10     private String peliculaSeleccionada;
11     private String horarioSeleccionado;
12     private String salaSeleccionada;
13
14     private static final int FILAS = 9;
15     private static final int COLUMNAS = 10;
16     private static final int PRECIO_POR_ASIENTO = 3000;
17     private static final String[] LETRAS_FILA = {"A", "B", "C", "D", "E", "F", "G", "H", "I"};
18
19     private JButton[][] asientos;
20     private JLabel montoTotalLabel;
21     private JLabel infoPeliculaLabel;
22
23     private JPanel panelAsientos;
24     private List<String> asientosReservados = new ArrayList<>();
25
26     public void setInfoPelicula(String nombre, String horario, String sala) {
27         this.peliculaSeleccionada = nombre;
28         this.horarioSeleccionado = horario;
29         this.salaSeleccionada = sala;
30         actualizarInfoPelicula();
31     }
```

El módulo fue creado en base a una matriz que de forma aleatoria tiene asientos ocupados, para dar mas dinamismo en la interacción con el usuario.

Al seleccionar los asientos se adjuntará su valor en la parte inferior de la ventana junto con los demas datos de los productos a adquirir en el cine

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
B1	B2	B3	B4	B5	B6	B7	B8	B9	B10
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
E1	E2	E3	E4	E5	E6	E7	E8	E9	E10
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
H1	H2	H3	H4	H5	H6	H7	H8	H9	H10
I1	I2	I3	I4	I5	I6	I7	I8	I9	I10

Monto Total: \$6000 Película: Aquaman 2 - Horario: 14:00 - Sala: 3DXD

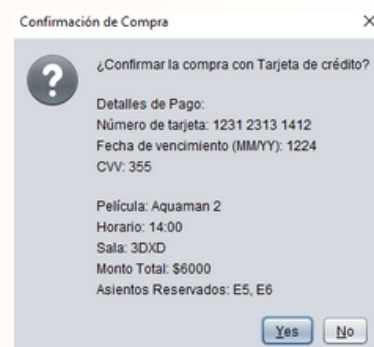
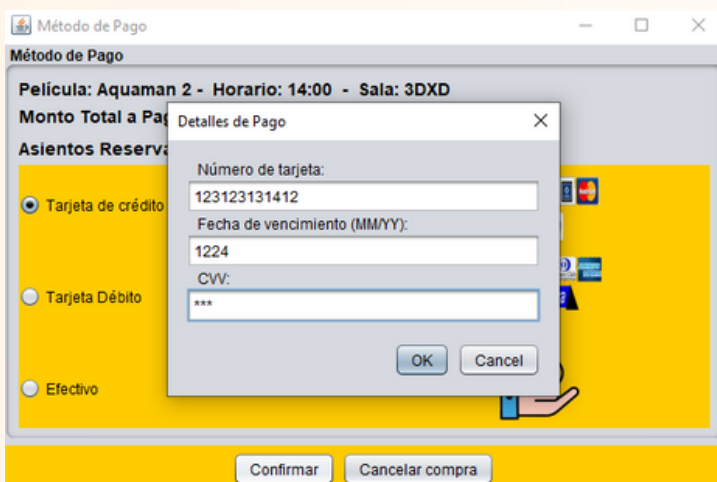
[Ir a Pagar](#)

CARACTERÍSTICAS DEL SISTEMA

MÓDULO - MÉTODO DE PAGO

```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5 import java.net.URL;
6 import java.util.List;
7
8 public class MetodoDePago extends JFrame {
9
10     private JRadioButton btnTarjetaCredito;
11     private JRadioButton btnTarjetaDebito;
12     private JRadioButton btnEfectivo;
13
14     private String pelicula;
15     private String horario;
16     private String sala;
17     private int montoTotal;
18     private List<String> asientosReservados;
19
20     public MetodoDePago(String pelicula, String horario, String sala, int montoTotal, List<String> asien
21         SwingUtilities.invokeLater(() -> {
22             this.pelicula = pelicula;
23             this.horario = horario;
24             this.sala = sala;
25             this.montoTotal = montoTotal;
26             this.asientosReservados = asientosReservados;
27
28             iniciarComponentes();
29             setVisible(true);
30             setLocationRelativeTo(null);
31         });
32     }
```

El código construye una ventana que muestra información. . Ofrece tres opciones de pago (tarjeta de crédito, tarjeta de débito o efectivo) representadas con botones de radio e imágenes respectivas.



La interfaz bastante intuitiva nos permite ahorrar tiempo y ser eficientes con los productos que querramos adquirir de la plataforma

Controladores en este modulo de son

```
int resultado = JOptionPane.showOptionDialog(  
    parentComponent: this,  
    message: panel,  
    title: "Detalles de Pago",  
    optionType: JOptionPane.OK_CANCEL_OPTION,  
    messageType: JOptionPane.PLAIN_MESSAGE,  
    icon: null,  
    options: null,  
    initialValue: null  
);  
  
if (resultado == JOptionPane.OK_OPTION) {  
    String numeroTarjeta = campoNumeroTarjeta.getText().replaceAll(regex: "\\s", replacement: ""); // Elimina  
    String fechaVencimiento = campoFechaVencimiento.getText();  
    String cvv = new String(value: campoCVV.getPassword());  
  
    // Validar campos  
    if (numeroTarjeta.isEmpty() || fechaVencimiento.isEmpty() || cvv.isEmpty() || !esNumero(texto: cvv)) {  
        return null;  
    }  
  
    // Formatear el número de tarjeta con espacios cada 4 dígitos  
    StringBuilder numeroFormateado = new StringBuilder();  
    for (int i = 0; i < numeroTarjeta.length(); i++) {  
        if (i > 0 && i % 4 == 0) {  
            numeroFormateado.append(str: " ");  
        }  
        numeroFormateado.append(c: numeroTarjeta.charAt(index: i));  
    }  
  
    return "Número de tarjeta: " + numeroFormateado.toString() +  
        "\nFecha de vencimiento (MM/YY): " + fechaVencimiento +  
        "\nCVV: " + cvv;  
}
```

En este caso, el controlador de eventos se activa cuando el usuario hace clic en el botón "OK" en el cuadro de diálogo de tarjeta de crédito.

Se obtiene el texto ingresado en los campos de texto (campoNumeroTarjeta, campoFechaVencimiento) y el contenido del campo de contraseña (campoCVV).

Se realiza una validación de campos para asegurarse de que ninguno de los campos esté vacío y que el CVV sea un número.

Si la validación es exitosa, la función devuelve un String que contiene la información de la tarjeta de crédito formateada.

el siguiente es el modulo de validacion el cual tiene un controlador Try-Catchel cual devuelve true si la cadena de texto puede convertirse exitosamente a un número entero y false si no puede

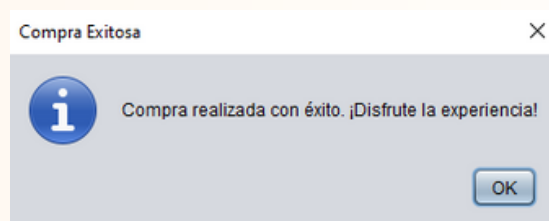
```
private boolean esNumero(String texto) {  
    try {  
        Integer.valueOf(s: texto);  
        return true;  
    } catch (NumberFormatException e) {  
        return false;  
    }  
}
```

CARACTERÍSTICAS DEL SISTEMA

MÓDULO - FACTURA

```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.text.NumberFormat;
4 import java.text.SimpleDateFormat;
5 import java.util.Date;
6 import java.util.List;
7
8 public class Factura extends JFrame {
9
10     private List<String> asientosReservados;
11
12     public Factura(String metodoPago, String pelicula, String horario, String sala, int montoT
13     SwingUtilities.invokeLater(() -> {
14         this.asientosReservados = asientosReservados;
15
16         // Configurar la ventana
17         setTitle(title: "Boleta de Compra");
18         setSize(width: 300, height: 400); // Tamaño ajustado
19         setDefaultCloseOperation(operation: JFrame.EXIT_ON_CLOSE);
20
21         // Obtener la resolución de la pantalla
22         Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
23
24         // Calcular la posición centrada en la pantalla
25         int x = (int) ((screenSize.getWidth() - getWidth()) / 2);
26         int y = (int) ((screenSize.getHeight() - getHeight()) / 2);
27
28         // Establecer la posición de la ventana
29         setLocation(x, y);
30
31         // Crear el panel principal de la boleta
32         JPanel boletaPanel = new JPanel();
```

El código construye una ventana que muestra información. Toma los datos de la reserva y se los muestra al usuario en forma de factura o boleta añadiendo un botón “imprimir que simula la descarga del archivo



La interfaz bastante intuitiva nos permite ahorrar tiempo y a la vez ser atractiva al uso general del público.

Y lo mas importante, el codigo funciona sin errores.

En el código el controlador de eventos principal se encuentra asociado al botón "Imprimir"

```
// Crear el botón imprimir
JButton imprimirButton = new JButton(text: "Imprimir");
imprimirButton.addActionListener(e -> imprimirBoleta());
imprimirButton.setAlignmentX(alignmentX: Component.CENTER_ALIGNMENT);

// Agregar el botón imprimir al panel principal
boletaPanel.add(comp: imprimirButton);
```

```
private void imprimirBoleta() {
    // Lógica para imprimir la boleta

    // Mostrar mensaje de compra exitosa después de imprimir
    JOptionPane.showMessageDialog(parentComponent: this, message: "Compra realizada con éxito. ¡Disfrute la experiencia!",
    dispose()); // Cerrar la ventana después de imprimir
}
```

El controlador de eventos está vinculado al botón "Imprimir" y, cuando se activa (cuando el botón es clicado), invoca el método `imprimirBoleta()`. Este método muestra un mensaje y cierra la ventana, indicando que la compra fue exitosa y la boleta fue impresa.

DESCRIPCIÓN FRAGMENTO CODIGO

MÓDULO SELECCIÓN DE ASIENTOS

Procederemos a describir algunas de las librerías implementadas en este módulo

```
1 import javax.swing.*;  
2 import java.awt.*;  
3 import java.awt.event.ActionEvent;  
4 import java.awt.event.ActionListener;  
5 import java.util.ArrayList;  
6 import java.util.List;  
7 import java.util.Random;
```

javax.swing.*: Proporciona componentes y clases para la creación de interfaces gráficas de usuario (GUI) en Java. Contiene elementos como botones, paneles, etiquetas, marcos (JFrame), entre otros, para construir aplicaciones con interfaz gráfica.

java.awt.*: Es un conjunto de clases para crear y manejar elementos de interfaz gráfica de usuario y manejo de eventos en Java. Proporciona componentes básicos y métodos para trabajar con ventanas, gráficos, eventos, entre otros.

java.awt.event.ActionEvent y java.awt.event.ActionListener: Estas clases son parte del paquete java.awt.event y se utilizan para manejar eventos de acción generados por diferentes componentes de la interfaz gráfica, como botones. ActionEvent es el evento en sí, y ActionListener es la interfaz utilizada para manejar dichos eventos.

java.util.ArrayList y java.util.List: Estas clases son parte del paquete java.util y se utilizan para trabajar con colecciones en Java. ArrayList es una implementación de la interfaz List, que proporciona una lista redimensionable que puede crecer o disminuir según sea necesario.

java.util.Random: Esta clase se encuentra en el paquete java.util y se usa para generar números pseudoaleatorios. Proporciona métodos para generar números aleatorios de diferentes tipos (enteros, doubles, etc.).

DESCRIPCIÓN FRAGMENTO CODIGO

MÓDULO SELECCIÓN DE ASIENTOS

```
9 public class SeleccionDeAsientos extends JFrame {
10     private String peliculaSeleccionada;
11     private String horarioSeleccionado;
12     private String salaSeleccionada;
13
14     private static final int FILAS = 9;
15     private static final int COLUMNAS = 10;
16     private static final int PRECIO_POR_ASIENTO = 3000;
17     private static final String[] LETRAS_FILA = {"A", "B", "C", "D", "E", "F", "G", "H", "I", "J"};
18
19     private JButton[][] asientos;
20     private JLabel montoTotalLabel;
21     private JLabel infoPeliculaLabel;
22
23     private JPanel panelAsientos;
24     private List<String> asientosReservados = new ArrayList<>();
25
26     public void setInfoPelicula(String nombre, String horario, String sala) {
27         this.peliculaSeleccionada = nombre;
28         this.horarioSeleccionado = horario;
29         this.salaSeleccionada = sala;
30         actualizarInfoPelicula();
31     }
```

Declaración de variables: Define las variables `peliculaSeleccionada`, `horarioSeleccionado` y `salaSeleccionada` para almacenar información sobre la película, horario y sala que el usuario selecciona.

Establece constantes como el número de filas, columnas, precio por asiento y letras para identificar las filas.

```
33 public SeleccionDeAsientos() {
34     super(title: "Sala de Cine");
35     setSize(width: 600, height: 400);
36     setDefaultCloseOperation(operation: JFrame.EXIT_ON_CLOSE);
37
38     inicializarAsientos();
39     crearGUI();
40     pack();
41     generarAsientosOcupadosAleatoriamente();
42     setLocationRelativeTo(c: null);
43     setVisible(b: true);
44 }
45
```

Crea una ventana de dimensiones 600x400 la cual contendrá la matriz con la selección de asientos que veremos a continuación

DESCRIPCIÓN FRAGMENTO CODIGO

MÓDULO SELECCIÓN DE ASIENTOS

```
46 private void crearGUI() {
47     JPanel panelPrincipal = new JPanel(new BorderLayout());
48     panelPrincipal.setLayout(new BoxLayout(target: panelPrincipal, axis: BoxLayout.Y_AXIS));
49
50     panelAsientos = new JPanel(new GridLayout(rows: FILAS, cols: COLUMNAS, hgap: 5, vgap: 5));
51     for (int i = 0; i < FILAS; i++) {
52         for (int j = 0; j < COLUMNAS; j++) {
53             JButton asiento = asientos[i][j];
54             asiento.setBackground(new Color(r: 255, g: 204, b: 0));
55             asiento.addActionListener(new AsientoListener(fila: i, columna: j));
56             panelAsientos.add(comp: asiento);
57         }
58     }
59
60     JPanel panelPantalla = new JPanel();
61     panelPantalla.setPreferredSize(new Dimension(width: 400, height: 200));
62     panelPantalla.setBackground(new Color(r: 0, g: 0, b: 0));
63     infoPelículaLabel = new JLabel();
64     infoPelículaLabel.setForeground(eg: Color.BLACK);
65     panelPantalla.add(comp: infoPelículaLabel);
66
67     JPanel panelInformacion = new JPanel(new FlowLayout());
68     montoTotalLabel = new JLabel(text: "Monto Total: $0");
69     panelInformacion.add(comp: montoTotalLabel);
70     montoTotalLabel.setForeground(eg: Color.BLACK);
71     panelPrincipal.add(comp: panelInformacion, constraints: BorderLayout.NORTH);
72     infoPelículaLabel = new JLabel(text: "Película: - Horario: - Sala: -");
73     panelInformacion.add(comp: infoPelículaLabel);
```

Utiliza Swing para crear la interfaz gráfica.

Crea un arreglo bidimensional de botones (asientos) que representa los asientos disponibles en la sala de cine.

Hay un panel de asientos donde se muestran los botones correspondientes a cada asiento.

Muestra información sobre la película seleccionada y el monto total a pagar.

```
116 private class AsientoListener implements ActionListener {
117     private int fila;
118     private int columna;
119
120     public AsientoListener(int fila, int columna) {
121         this.fila = fila;
122         this.columna = columna;
123     }
124
125     @Override
126     public void actionPerformed(ActionEvent e) {
127         JButton asiento = (JButton) e.getSource();
128         if (asiento.getBackground() == Color.GREEN) {
129             asiento.setBackground(new Color(r: 255, g: 204, b: 0));
130             asientosReservados.remove(LETRAS_FILA[fila] + (columna + 1));
131         } else {
132             asiento.setBackground(bg: Color.GREEN);
133             asientosReservados.add(LETRAS_FILA[fila] + (columna + 1));
134         }
135         actualizarMontoTotal();
136     }
137 }
```

Asigna un ActionListener a cada botón de asiento para marcar los asientos como reservados cuando se hace clic en ellos.

Actualiza dinámicamente el monto total y los asientos reservados cuando se hacen cambios en la selección de asientos.

DESCRIPCIÓN FRAGMENTO CODIGO

MÓDULO SELECCIÓN DE ASIENTOS

```
162 private class Pagolistener implements ActionListener {
163     @Override
164     public void actionPerformed(ActionEvent e) {
165         String pelicula = peliculaSeleccionada;
166         String horario = horarioSeleccionado;
167         String sala = salaSeleccionada;
168         int montoTotal = calcularMontoTotal();
169
170         MetodoDePago metodoDePago = new MetodoDePago(pelicula, horario, sala, montoTotal, asientosReservados);
171         metodoDePago.setVisible(b: true);
172         dispose(); // Cierra la ventana actual (SeleccionDeAsientos)
173     }
174 }
```

Cuando se hace clic en "Ir a Pagar", crea una instancia de MetodoDePago y cierra la ventana actual de selección de asientos.

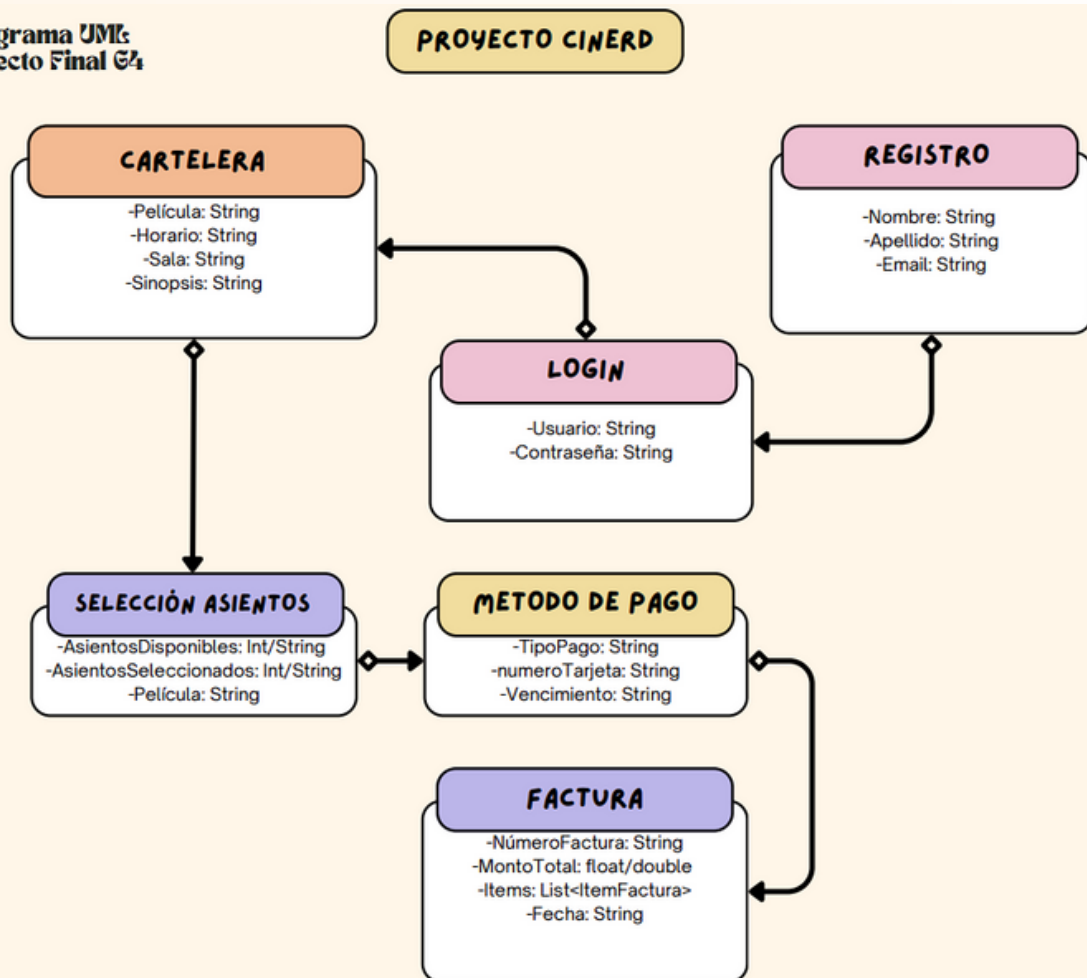
```
176 private void generarAsientosOcupadosAleatoriamente() {
177     Random random = new Random();
178     int numAsientosOcupados = random.nextInt(FILAS * COLUMNAS);
179
180     for (int i = 0; i < numAsientosOcupados; i++) {
181         int fila = random.nextInt(bound: FILAS);
182         int columna = random.nextInt(bound: COLUMNAS);
183
184         if (asientos[fila][columna].getBackground() != Color.GREEN) {
185             asientos[fila][columna].setBackground(bg: Color.GRAY);
186             asientos[fila][columna].setEnabled(b: false);
187         } else {
188             i--;
189         }
190     }
191 }
```

Rellena la matriz con asientos ocupados aleatoriamente para poder dar mas realismo a la hora de reservar asientos (mas clientes en la plataforma reservando asientos)

Estos son algunos Listener o características implementadas dentro del módulo de “Selección de asientos”. La idea es informar y contextualizar de su uso, ya que para detalles se adjuntará el código realizado en Netbeans junto a este informe

DIAGRAMA UML FINAL

Diagrama UML:
Proyecto Final G4



Representa el flujo del proyecto, los módulos y sus respectivas “flechas” simbolizan la dirección en que se van desarrollando o utilizando las clases con sus respectivos atributos

COMPARATIVA DIAGRAMA UML FINAL

Este diagrama sirvió como base para poder crear los módulos e interfaces iniciales, así como las aspiraciones organizacionales del grupo de trabajo

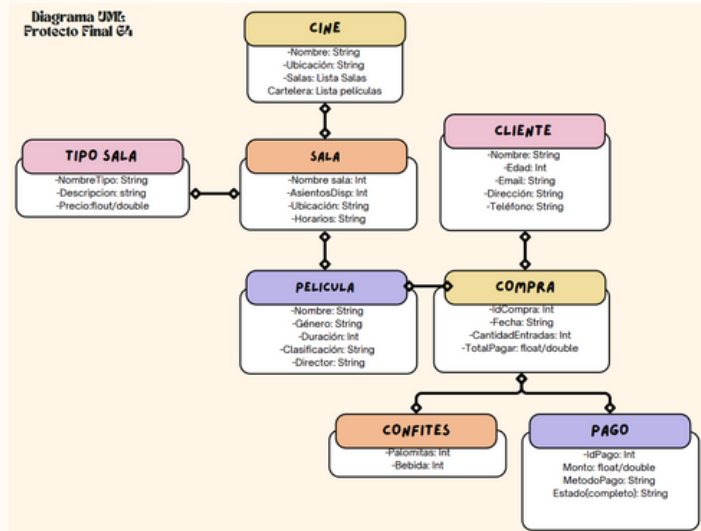
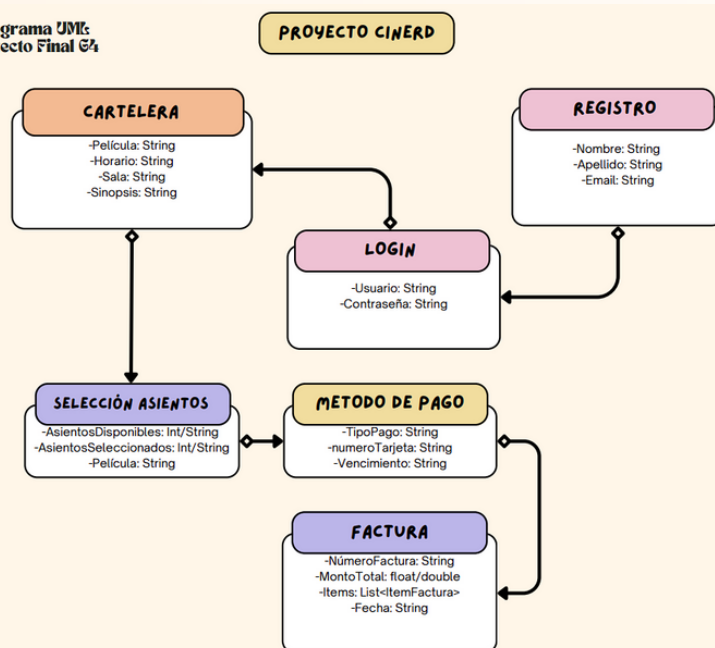


Diagrama UML: Proyecto Final 24



Este diagrama mas simple y facil de entender, nos da el flujo de tareas y dirección en que se van ejecutando los distintos módulos del proyecto. Todos explicados anteriormente con sus respectivas interfaces gráficas

Se actualizaron los nombres de los atributos que se utilizaron. El diseño permanece siendo el mismo con tal de que la diferencia con el diagrama UML inicial se pueda apreciar de forma mas notoria.

GRUPO 4

