1. Importa los datos al ambiente de trabajo y revisamos que no haya huecos. Calculamos la diferencia de promedios entre las clases 2 y 4 para todos los genes, e imprimos los 10 genes con la mayor diferencia de medias.

In [32]:

```python
# 1. Importación y Análisis Inicial (Versión Mejorada)
import pandas as pd
import numpy as np
import os

# 1.1 Importar los datos (con manejo de errores)
try:
    # Intentar con diferentes nombres comunes del archivo
    try:
        data = pd.read_csv('A3.1_Khan.csv')
    except FileNotFoundError:
        data = pd.read_csv('A3.1 Khan.csv')  # Versión sin guión bajo

    print("Datos cargados correctamente")

    # 1.2 Verificación de estructura
    print("\nPrimeras filas del DataFrame:")
    print(data.head())

    print("\nNombres de columnas disponibles:")
    print(data.columns.tolist())

    # 1.3 Búsqueda automática de la columna clase
    possible_class_cols = [col for col in data.columns
                          if data[col].nunique() <= 4 and data[col].nunique() > 1]

    if not possible_class_cols:
        raise ValueError("Error: No se encontró una columna con 2-4 valores únicos que pueda ser la variable clase")

    class_col = possible_class_cols[0]
    print(f"\nVariable clase identificada: '{class_col}' con valores {sorted(data[class_col].unique())}")

    # 1.4 Separación de características y variable objetivo
    X = data.drop(class_col, axis=1)
    y = data[class_col]

    # 1.5 Verificación de valores faltantes
    print("\nValores faltantes totales:", X.isnull().sum().sum())

    # 1.6 Cálculo de diferencias de medias (clases 2 vs 4)
    if {2, 4}.issubset(y.unique()):
        class_2 = X[y == 2]
        class_4 = X[y == 4]
        mean_diff = class_2.mean() - class_4.mean()
        top_10_genes = mean_diff.abs().sort_values(ascending=False).head(10)

        print("\nTop 10 genes con mayor diferencia absoluta de medias (clase 2 vs 4):")
        print(top_10_genes)
    else:
        print("\nAdvertencia: Las clases 2 y/o 4 no existen en los datos")

except Exception as e:
    print(f"\nError durante la carga/validación: {str(e)}")
    print("Verifica:")
    print("1. Que el archivo existe y tiene el nombre correcto")
    print("2. Que contiene una columna con los valores de clase (1-4)")
    print("3. Que los datos tienen la estructura esperada (genes en columnas)")
```

```
Datos cargados correctamente

Primeras filas del DataFrame:
         X1        X2        X3        X4        X5        X6        X7  \
0  0.773344 -2.438405 -0.482562 -2.721135 -1.217058  0.827809  1.342604
1 -0.078178 -2.415754  0.412772 -2.825146 -0.626236  0.054488  1.429498
2 -0.084469 -1.649739 -0.241308 -2.875286 -0.889405 -0.027474  1.159300
3  0.965614 -2.380547  0.625297 -1.741256 -0.845366  0.949687  1.093801
4  0.075664 -1.728785  0.852626  0.272695 -1.841370  0.327936  1.251219

         X8        X9       X10  ...     X2300     X2301     X2302     X2303  \
0  0.057042  0.133569  0.565427  ... -0.027474 -1.660205  0.588231 -0.463624
1 -0.120249  0.456792  0.159053  ... -0.246284 -0.836325 -0.571284  0.034788
2  0.015676  0.191942  0.496585  ...  0.024985 -1.059872 -0.403767 -0.678653
3  0.819736 -0.284620  0.994732  ...  0.357115 -1.893128  0.255107  0.163309
4  0.771450  0.030917  0.278313  ...  0.061753 -2.273998 -0.039365  0.368801

      X2304     X2305     X2306     X2307     X2308  y
0 -3.952845 -5.496768 -1.414282 -0.647600 -1.763172  2
1 -2.478130 -3.661264 -1.093923 -1.209320 -0.824395  2
2 -2.939352 -2.736450 -1.965399 -0.805868 -1.139434  2
3 -1.021929 -2.077843 -1.127629  0.331531 -2.179483  2
4 -2.566551 -1.675044 -1.082050 -0.965218 -1.836966  2

[5 rows x 2309 columns]

Nombres de columnas disponibles:
['X1', 'X2', 'X3', 'X4', 'X5', 'X6', 'X7', 'X8', 'X9', 'X10', 'X11', 'X12', 'X13', 'X14', 'X15', 'X16', 'X17', 'X18', 'X19', 'X20', 'X21', 'X22', 'X23', 'X24', 'X25', 'X26', 'X27', 'X28', 'X29', 'X30', 'X31', 'X32', 'X33', 'X34', 'X3
5', 'X36', 'X37', 'X38', 'X39', 'X40', 'X41', 'X42', 'X43', 'X44', 'X45', 'X46', 'X47', 'X48', 'X49', 'X50', 'X51', 'X52', 'X53', 'X54', 'X55', 'X56', 'X57', 'X58', 'X59', 'X60', 'X61', 'X62', 'X63', 'X64', 'X65', 'X66', 'X67', 'X68',
'X69', 'X70', 'X71', 'X72', 'X73', 'X74', 'X75', 'X76', 'X77', 'X78', 'X79', 'X80', 'X81', 'X82', 'X83', 'X84', 'X85', 'X86', 'X87', 'X88', 'X89', 'X90', 'X91', 'X92', 'X93', 'X94', 'X95', 'X96', 'X97', 'X98', 'X99', 'X100', 'X101', 'X
102', 'X103', 'X104', 'X105', 'X106', 'X107', 'X108', 'X109', 'X110', 'X111', 'X112', 'X113', 'X114', 'X115', 'X116', 'X117', 'X118', 'X119', 'X120', 'X121', 'X122', 'X123', 'X124', 'X125', 'X126', 'X127', 'X128', 'X129', 'X130', 'X13
1', 'X132', 'X133', 'X134', 'X135', 'X136', 'X137', 'X138', 'X139', 'X140', 'X141', 'X142', 'X143', 'X144', 'X145', 'X146', 'X147', 'X148', 'X149', 'X150', 'X151', 'X152', 'X153', 'X154', 'X155', 'X156', 'X157', 'X158', 'X159', 'X160',
'X161', 'X162', 'X163', 'X164', 'X165', 'X166', 'X167', 'X168', 'X169', 'X170', 'X171', 'X172', 'X173', 'X174', 'X175', 'X176', 'X177', 'X178', 'X179', 'X180', 'X181', 'X182', 'X183', 'X184', 'X185', 'X186', 'X187', 'X188', 'X189', 'X1
90', 'X191', 'X192', 'X193', 'X194', 'X195', 'X196', 'X197', 'X198', 'X199', 'X200', 'X201', 'X202', 'X203', 'X204', 'X205', 'X206', 'X207', 'X208', 'X209', 'X210', 'X211', 'X212', 'X213', 'X214', 'X215', 'X216', 'X217', 'X218', 'X21
9', 'X220', 'X221', 'X222', 'X223', 'X224', 'X225', 'X226', 'X227', 'X228', 'X229', 'X230', 'X231', 'X232', 'X233', 'X234', 'X235', 'X236', 'X237', 'X238', 'X239', 'X240', 'X241', 'X242', 'X243', 'X244', 'X245', 'X246', 'X247', 'X248',
'X249', 'X250', 'X251', 'X252', 'X253', 'X254', 'X255', 'X256', 'X257', 'X258', 'X259', 'X260', 'X261', 'X262', 'X263', 'X264', 'X265', 'X266', 'X267', 'X268', 'X269', 'X270', 'X271', 'X272', 'X273', 'X274', 'X275', 'X276', 'X277', 'X2
78', 'X279', 'X280', 'X281', 'X282', 'X283', 'X284', 'X285', 'X286', 'X287', 'X288', 'X289', 'X290', 'X291', 'X292', 'X293', 'X294', 'X295', 'X296', 'X297', 'X298', 'X299', 'X300', 'X301', 'X302', 'X303', 'X304', 'X305', 'X306', 'X30
7', 'X308', 'X309', 'X310', 'X311', 'X312', 'X313', 'X314', 'X315', 'X316', 'X317', 'X318', 'X319', 'X320', 'X321', 'X322', 'X323', 'X324', 'X325', 'X326', 'X327', 'X328', 'X329', 'X330', 'X331', 'X332', 'X333', 'X334', 'X335', 'X336',
'X337', 'X338', 'X339', 'X340', 'X341', 'X342', 'X343', 'X344', 'X345', 'X346', 'X347', 'X348', 'X349', 'X350', 'X351', 'X352', 'X353', 'X354', 'X355', 'X356', 'X357', 'X358', 'X359', 'X360', 'X361', 'X362', 'X363', 'X364', 'X365', 'X3
66', 'X367', 'X368', 'X369', 'X370', 'X371', 'X372', 'X373', 'X374', 'X375', 'X376', 'X377', 'X378', 'X379', 'X380', 'X381', 'X382', 'X383', 'X384', 'X385', 'X386', 'X387', 'X388', 'X389', 'X390', 'X391', 'X392', 'X393', 'X394', 'X39
5', 'X396', 'X397', 'X398', 'X399', 'X400', 'X401', 'X402', 'X403', 'X404', 'X405', 'X406', 'X407', 'X408', 'X409', 'X410', 'X411', 'X412', 'X413', 'X414', 'X415', 'X416', 'X417', 'X418', 'X419', 'X420', 'X421', 'X422', 'X423', 'X424',
'X425', 'X426', 'X427', 'X428', 'X429', 'X430', 'X431', 'X432', 'X433', 'X434', 'X435', 'X436', 'X437', 'X438', 'X439', 'X440', 'X441', 'X442', 'X443', 'X444', 'X445', 'X446', 'X447', 'X448', 'X449', 'X450', 'X451', 'X452', 'X453', 'X4
54', 'X455', 'X456', 'X457', 'X458', 'X459', 'X460', 'X461', 'X462', 'X463', 'X464', 'X465', 'X466', 'X467', 'X468', 'X469', 'X470', 'X471', 'X472', 'X473', 'X474', 'X475', 'X476', 'X477', 'X478', 'X479', 'X480', 'X481', 'X482', 'X48
3', 'X484', 'X485', 'X486', 'X487', 'X488', 'X489', 'X490', 'X491', 'X492', 'X493', 'X494', 'X495', 'X496', 'X497', 'X498', 'X499', 'X500', 'X501', 'X502', 'X503', 'X504', 'X505', 'X506', 'X507', 'X508', 'X509', 'X510', 'X511', 'X512',
'X513', 'X514', 'X515', 'X516', 'X517', 'X518', 'X519', 'X520', 'X521', 'X522', 'X523', 'X524', 'X525', 'X526', 'X527', 'X528', 'X529', 'X530', 'X531', 'X532', 'X533', 'X534', 'X535', 'X536', 'X537', 'X538', 'X539', 'X540', 'X541', 'X5
42', 'X543', 'X544', 'X545', 'X546', 'X547', 'X548', 'X549', 'X550', 'X551', 'X552', 'X553', 'X554', 'X555', 'X556', 'X557', 'X558', 'X559', 'X560', 'X561', 'X562', 'X563', 'X564', 'X565', 'X566', 'X567', 'X568', 'X569', 'X570', 'X57
1', 'X572', 'X573', 'X574', 'X575', 'X576', 'X577', 'X578', 'X579', 'X580', 'X581', 'X582', 'X583', 'X584', 'X585', 'X586', 'X587', 'X588', 'X589', 'X590', 'X591', 'X592', 'X593', 'X594', 'X595', 'X596', 'X597', 'X598', 'X599', 'X600',
'X601', 'X602', 'X603', 'X604', 'X605', 'X606', 'X607', 'X608', 'X609', 'X610', 'X611', 'X612', 'X613', 'X614', 'X615', 'X616', 'X617', 'X618', 'X619', 'X620', 'X621', 'X622', 'X623', 'X624', 'X625', 'X626', 'X627', 'X628', 'X629', 'X6
30', 'X631', 'X632', 'X633', 'X634', 'X635', 'X636', 'X637', 'X638', 'X639', 'X640', 'X641', 'X642', 'X643', 'X644', 'X645', 'X646', 'X647', 'X648', 'X649', 'X650', 'X651', 'X652', 'X653', 'X654', 'X655', 'X656', 'X657', 'X658', 'X65
9', 'X660', 'X661', 'X662', 'X663', 'X664', 'X665', 'X666', 'X667', 'X668', 'X669', 'X670', 'X671', 'X672', 'X673', 'X674', 'X675', 'X676', 'X677', 'X678', 'X679', 'X680', 'X681', 'X682', 'X683', 'X684', 'X685', 'X686', 'X687', 'X688',
'X689', 'X690', 'X691', 'X692', 'X693', 'X694', 'X695', 'X696', 'X697', 'X698', 'X699', 'X700', 'X701', 'X702', 'X703', 'X704', 'X705', 'X706', 'X707', 'X708', 'X709', 'X710', 'X711', 'X712', 'X713', 'X714', 'X715', 'X716', 'X717', 'X7
18', 'X719', 'X720', 'X721', 'X722', 'X723', 'X724', 'X725', 'X726', 'X727', 'X728', 'X729', 'X730', 'X731', 'X732', 'X733', 'X734', 'X735', 'X736', 'X737', 'X738', 'X739', 'X740', 'X741', 'X742', 'X743', 'X744', 'X745', 'X746', 'X74
7', 'X748', 'X749', 'X750', 'X751', 'X752', 'X753', 'X754', 'X755', 'X756', 'X757', 'X758', 'X759', 'X760', 'X761', 'X762', 'X763', 'X764', 'X765', 'X766', 'X767', 'X768', 'X769', 'X770', 'X771', 'X772', 'X773', 'X774', 'X775', 'X776',
'X777', 'X778', 'X779', 'X780', 'X781', 'X782', 'X783', 'X784', 'X785', 'X786', 'X787', 'X788', 'X789', 'X790', 'X791', 'X792', 'X793', 'X794', 'X795', 'X796', 'X797', 'X798', 'X799', 'X800', 'X801', 'X802', 'X803', 'X804', 'X805', 'X8
06', 'X807', 'X808', 'X809', 'X810', 'X811', 'X812', 'X813', 'X814', 'X815', 'X816', 'X817', 'X818', 'X819', 'X820', 'X821', 'X822', 'X823', 'X824', 'X825', 'X826', 'X827', 'X828', 'X829', 'X830', 'X831', 'X832', 'X833', 'X834', 'X83
5', 'X836', 'X837', 'X838', 'X839', 'X840', 'X841', 'X842', 'X843', 'X844', 'X845', 'X846', 'X847', 'X848', 'X849', 'X850', 'X851', 'X852', 'X853', 'X854', 'X855', 'X856', 'X857', 'X858', 'X859', 'X860', 'X861', 'X862', 'X863', 'X864',
'X865', 'X866', 'X867', 'X868', 'X869', 'X870', 'X871', 'X872', 'X873', 'X874', 'X875', 'X876', 'X877', 'X878', 'X879', 'X880', 'X881', 'X882', 'X883', 'X884', 'X885', 'X886', 'X887', 'X888', 'X889', 'X890', 'X891', 'X892', 'X893', 'X8
94', 'X895', 'X896', 'X897', 'X898', 'X899', 'X900', 'X901', 'X902', 'X903', 'X904', 'X905', 'X906', 'X907', 'X908', 'X909', 'X910', 'X911', 'X912', 'X913', 'X914', 'X915', 'X916', 'X917', 'X918', 'X919', 'X920', 'X921', 'X922', 'X92
3', 'X924', 'X925', 'X926', 'X927', 'X928', 'X929', 'X930', 'X931', 'X932', 'X933', 'X934', 'X935', 'X936', 'X937', 'X938', 'X939', 'X940', 'X941', 'X942', 'X943', 'X944', 'X945', 'X946', 'X947', 'X948', 'X949', 'X950', 'X951', 'X952',
'X953', 'X954', 'X955', 'X956', 'X957', 'X958', 'X959', 'X960', 'X961', 'X962', 'X963', 'X964', 'X965', 'X966', 'X967', 'X968', 'X969', 'X970', 'X971', 'X972', 'X973', 'X974', 'X975', 'X976', 'X977', 'X978', 'X979', 'X980', 'X981', 'X9
82', 'X983', 'X984', 'X985', 'X986', 'X987', 'X988', 'X989', 'X990', 'X991', 'X992', 'X993', 'X994', 'X995', 'X996', 'X997', 'X998', 'X999', 'X1000', 'X1001', 'X1002', 'X1003', 'X1004', 'X1005', 'X1006', 'X1007', 'X1008', 'X1009', 'X10
10', 'X1011', 'X1012', 'X1013', 'X1014', 'X1015', 'X1016', 'X1017', 'X1018', 'X1019', 'X1020', 'X1021', 'X1022', 'X1023', 'X1024', 'X1025', 'X1026', 'X1027', 'X1028', 'X1029', 'X1030', 'X1031', 'X1032', 'X1033', 'X1034', 'X1035', 'X103
6', 'X1037', 'X1038', 'X1039', 'X1040', 'X1041', 'X1042', 'X1043', 'X1044', 'X1045', 'X1046', 'X1047', 'X1048', 'X1049', 'X1050', 'X1051', 'X1052', 'X1053', 'X1054', 'X1055', 'X1056', 'X1057', 'X1058', 'X1059', 'X1060', 'X1061', 'X106
2', 'X1063', 'X1064', 'X1065', 'X1066', 'X1067', 'X1068', 'X1069', 'X1070', 'X1071', 'X1072', 'X1073', 'X1074', 'X1075', 'X1076', 'X1077', 'X1078', 'X1079', 'X1080', 'X1081', 'X1082', 'X1083', 'X1084', 'X1085', 'X1086', 'X1087', 'X108
8', 'X1089', 'X1090', 'X1091', 'X1092', 'X1093', 'X1094', 'X1095', 'X1096', 'X1097', 'X1098', 'X1099', 'X1100', 'X1101', 'X1102', 'X1103', 'X1104', 'X1105', 'X1106', 'X1107', 'X1108', 'X1109', 'X1110', 'X1111', 'X1112', 'X1113', 'X111
4', 'X1115', 'X1116', 'X1117', 'X1118', 'X1119', 'X1120', 'X1121', 'X1122', 'X1123', 'X1124', 'X1125', 'X1126', 'X1127', 'X1128', 'X1129', 'X1130', 'X1131', 'X1132', 'X1133', 'X1134', 'X1135', 'X1136', 'X1137', 'X1138', 'X1139', 'X114
0', 'X1141', 'X1142', 'X1143', 'X1144', 'X1145', 'X1146', 'X1147', 'X1148', 'X1149', 'X1150', 'X1151', 'X1152', 'X1153', 'X1154', 'X1155', 'X1156', 'X1157', 'X1158', 'X1159', 'X1160', 'X1161', 'X1162', 'X1163', 'X1164', 'X1165', 'X116
6', 'X1167', 'X1168', 'X1169', 'X1170', 'X1171', 'X1172', 'X1173', 'X1174', 'X1175', 'X1176', 'X1177', 'X1178', 'X1179', 'X1180', 'X1181', 'X1182', 'X1183', 'X1184', 'X1185', 'X1186', 'X1187', 'X1188', 'X1189', 'X1190', 'X1191', 'X119
2', 'X1193', 'X1194', 'X1195', 'X1196', 'X1197', 'X1198', 'X1199', 'X1200', 'X1201', 'X1202', 'X1203', 'X1204', 'X1205', 'X1206', 'X1207', 'X1208', 'X1209', 'X1210', 'X1211', 'X1212', 'X1213', 'X1214', 'X1215', 'X1216', 'X1217', 'X121
8', 'X1219', 'X1220', 'X1221', 'X1222', 'X1223', 'X1224', 'X1225', 'X1226', 'X1227', 'X1228', 'X1229', 'X1230', 'X1231', 'X1232', 'X1233', 'X1234', 'X1235', 'X1236', 'X1237', 'X1238', 'X1239', 'X1240', 'X1241', 'X1242', 'X1243', 'X124
4', 'X1245', 'X1246', 'X1247', 'X1248', 'X1249', 'X1250', 'X1251', 'X1252', 'X1253', 'X1254', 'X1255', 'X1256', 'X1257', 'X1258', 'X1259', 'X1260', 'X1261', 'X1262', 'X1263', 'X1264', 'X1265', 'X1266', 'X1267', 'X1268', 'X1269', 'X127
0', 'X1271', 'X1272', 'X1273', 'X1274', 'X1275', 'X1276', 'X1277', 'X1278', 'X1279', 'X1280', 'X1281', 'X1282', 'X1283', 'X1284', 'X1285', 'X1286', 'X1287', 'X1288', 'X1289', 'X1290', 'X1291', 'X1292', 'X1293', 'X1294', 'X1295', 'X129
6', 'X1297', 'X1298', 'X1299', 'X1300', 'X1301', 'X1302', 'X1303', 'X1304', 'X1305', 'X1306', 'X1307', 'X1308', 'X1309', 'X1310', 'X1311', 'X1312', 'X1313', 'X1314', 'X1315', 'X1316', 'X1317', 'X1318', 'X1319', 'X1320', 'X1321', 'X132
2', 'X1323', 'X1324', 'X1325', 'X1326', 'X1327', 'X1328', 'X1329', 'X1330', 'X1331', 'X1332', 'X1333', 'X1334', 'X1335', 'X1336', 'X1337', 'X1338', 'X1339', 'X1340', 'X1341', 'X1342', 'X1343', 'X1344', 'X1345', 'X1346', 'X1347', 'X134
8', 'X1349', 'X1350', 'X1351', 'X1352', 'X1353', 'X1354', 'X1355', 'X1356', 'X1357', 'X1358', 'X1359', 'X1360', 'X1361', 'X1362', 'X1363', 'X1364', 'X1365', 'X1366', 'X1367', 'X1368', 'X1369', 'X1370', 'X1371', 'X1372', 'X1373', 'X137
4', 'X1375', 'X1376', 'X1377', 'X1378', 'X1379', 'X1380', 'X1381', 'X1382', 'X1383', 'X1384', 'X1385', 'X1386', 'X1387', 'X1388', 'X1389', 'X1390', 'X1391', 'X1392', 'X1393', 'X1394', 'X1395', 'X1396', 'X1397', 'X1398', 'X1399', 'X140
0', 'X1401', 'X1402', 'X1403', 'X1404', 'X1405', 'X1406', 'X1407', 'X1408', 'X1409', 'X1410', 'X1411', 'X1412', 'X1413', 'X1414', 'X1415', 'X1416', 'X1417', 'X1418', 'X1419', 'X1420', 'X1421', 'X1422', 'X1423', 'X1424', 'X1425', 'X142
6', 'X1427', 'X1428', 'X1429', 'X1430', 'X1431', 'X1432', 'X1433', 'X1434', 'X1435', 'X1436', 'X1437', 'X1438', 'X1439', 'X1440', 'X1441', 'X1442', 'X1443', 'X1444', 'X1445', 'X1446', 'X1447', 'X1448', 'X1449', 'X1450', 'X1451', 'X145
2', 'X1453', 'X1454', 'X1455', 'X1456', 'X1457', 'X1458', 'X1459', 'X1460', 'X1461', 'X1462', 'X1463', 'X1464', 'X1465', 'X1466', 'X1467', 'X1468', 'X1469', 'X1470', 'X1471', 'X1472', 'X1473', 'X1474', 'X1475', 'X1476', 'X1477', 'X147
8', 'X1479', 'X1480', 'X1481', 'X1482', 'X1483', 'X1484', 'X1485', 'X1486', 'X1487', 'X1488', 'X1489', 'X1490', 'X1491', 'X1492', 'X1493', 'X1494', 'X1495', 'X1496', 'X1497', 'X1498', 'X1499', 'X1500', 'X1501', 'X1502', 'X1503', 'X150
4', 'X1505', 'X1506', 'X1507', 'X1508', 'X1509', 'X1510', 'X1511', 'X1512', 'X1513', 'X1514', 'X1515', 'X1516', 'X1517', 'X1518', 'X1519', 'X1520', 'X1521', 'X1522', 'X1523', 'X1524', 'X1525', 'X1526', 'X1527', 'X1528', 'X1529', 'X153
0', 'X1531', 'X1532', 'X1533', 'X1534', 'X1535', 'X1536', 'X1537', 'X1538', 'X1539', 'X1540', 'X1541', 'X1542', 'X1543', 'X1544', 'X1545', 'X1546', 'X1547', 'X1548', 'X1549', 'X1550', 'X1551', 'X1552', 'X1553', 'X1554', 'X1555', 'X155
6', 'X1557', 'X1558', 'X1559', 'X1560', 'X1561', 'X1562', 'X1563', 'X1564', 'X1565', 'X1566', 'X1567', 'X1568', 'X1569', 'X1570', 'X1571', 'X1572', 'X1573', 'X1574', 'X1575', 'X1576', 'X1577', 'X1578', 'X1579', 'X1580', 'X1581', 'X158
2', 'X1583', 'X1584', 'X1585', 'X1586', 'X1587', 'X1588', 'X1589', 'X1590', 'X1591', 'X1592', 'X1593', 'X1594', 'X1595', 'X1596', 'X1597', 'X1598', 'X1599', 'X1600', 'X1601', 'X1602', 'X1603', 'X1604', 'X1605', 'X1606', 'X1607', 'X160
8', 'X1609', 'X1610', 'X1611', 'X1612', 'X1613', 'X1614', 'X1615', 'X1616', 'X1617', 'X1618', 'X1619', 'X1620', 'X1621', 'X1622', 'X1623', 'X1624', 'X1625', 'X1626', 'X1627', 'X1628', 'X1629', 'X1630', 'X1631', 'X1632', 'X1633', 'X163
4', 'X1635', 'X1636', 'X1637', 'X1638', 'X1639', 'X1640', 'X1641', 'X1642', 'X1643', 'X1644', 'X1645', 'X1646', 'X1647', 'X1648', 'X1649', 'X1650', 'X1651', 'X1652', 'X1653', 'X1654', 'X1655', 'X1656', 'X1657', 'X1658', 'X1659', 'X166
0', 'X1661', 'X1662', 'X1663', 'X1664', 'X1665', 'X1666', 'X1667', 'X1668', 'X1669', 'X1670', 'X1671', 'X1672', 'X1673', 'X1674', 'X1675', 'X1676', 'X1677', 'X1678', 'X1679', 'X1680', 'X1681', 'X1682', 'X1683', 'X1684', 'X1685', 'X168
6', 'X1687', 'X1688', 'X1689', 'X1690', 'X1691', 'X1692', 'X1693', 'X1694', 'X1695', 'X1696', 'X1697', 'X1698', 'X1699', 'X1700', 'X1701', 'X1702', 'X1703', 'X1704', 'X1705', 'X1706', 'X1707', 'X1708', 'X1709', 'X1710', 'X1711', 'X171
2', 'X1713', 'X1714', 'X1715', 'X1716', 'X1717', 'X1718', 'X1719', 'X1720', 'X1721', 'X1722', 'X1723', 'X1724', 'X1725', 'X1726', 'X1727', 'X1728', 'X1729', 'X1730', 'X1731', 'X1732', 'X1733', 'X1734', 'X1735', 'X1736', 'X1737', 'X173
8', 'X1739', 'X1740', 'X1741', 'X1742', 'X1743', 'X1744', 'X1745', 'X1746', 'X1747', 'X1748', 'X1749', 'X1750', 'X1751', 'X1752', 'X1753', 'X1754', 'X1755', 'X1756', 'X1757', 'X1758', 'X1759', 'X1760', 'X1761', 'X1762', 'X1763', 'X176
4', 'X1765', 'X1766', 'X1767', 'X1768', 'X1769', 'X1770', 'X1771', 'X1772', 'X1773', 'X1774', 'X1775', 'X1776', 'X1777', 'X1778', 'X1779', 'X1780', 'X1781', 'X1782', 'X1783', 'X1784', 'X1785', 'X1786', 'X1787', 'X1788', 'X1789', 'X179
0', 'X1791', 'X1792', 'X1793', 'X1794', 'X1795', 'X1796', 'X1797', 'X1798', 'X1799', 'X1800', 'X1801', 'X1802', 'X1803', 'X1804', 'X1805', 'X1806', 'X1807', 'X1808', 'X1809', 'X1810', 'X1811', 'X1812', 'X1813', 'X1814', 'X1815', 'X181
6', 'X1817', 'X1818', 'X1819', 'X1820', 'X1821', 'X1822', 'X1823', 'X1824', 'X1825', 'X1826', 'X1827', 'X1828', 'X1829', 'X1830', 'X1831', 'X1832', 'X1833', 'X1834', 'X1835', 'X1836', 'X1837', 'X1838', 'X1839', 'X1840', 'X1841', 'X184
2', 'X1843', 'X1844', 'X1845', 'X1846', 'X1847', 'X1848', 'X1849', 'X1850', 'X1851', 'X1852', 'X1853', 'X1854', 'X1855', 'X1856', 'X1857', 'X1858', 'X1859', 'X1860', 'X1861', 'X1862', 'X1863', 'X1864', 'X1865', 'X1866', 'X1867', 'X186
8', 'X1869', 'X1870', 'X1871', 'X1872', 'X1873', 'X1874', 'X1875', 'X1876', 'X1877', 'X1878', 'X1879', 'X1880', 'X1881', 'X1882', 'X1883', 'X1884', 'X1885', 'X1886', 'X1887', 'X1888', 'X1889', 'X1890', 'X1891', 'X1892', 'X1893', 'X189
4', 'X1895', 'X1896', 'X1897', 'X1898', 'X1899', 'X1900', 'X1901', 'X1902', 'X1903', 'X1904', 'X1905', 'X1906', 'X1907', 'X1908', 'X1909', 'X1910', 'X1911', 'X1912', 'X1913', 'X1914', 'X1915', 'X1916', 'X1917', 'X1918', 'X1919', 'X192
0', 'X1921', 'X1922', 'X1923', 'X1924', 'X1925', 'X1926', 'X1927', 'X1928', 'X1929', 'X1930', 'X1931', 'X1932', 'X1933', 'X1934', 'X1935', 'X1936', 'X1937', 'X1938', 'X1939', 'X1940', 'X1941', 'X1942', 'X1943', 'X1944', 'X1945', 'X194
6', 'X1947', 'X1948', 'X1949', 'X1950', 'X1951', 'X1952', 'X1953', 'X1954', 'X1955', 'X1956', 'X1957', 'X1958', 'X1959', 'X1960', 'X1961', 'X1962', 'X1963', 'X1964', 'X1965', 'X1966', 'X1967', 'X1968', 'X1969', 'X1970', 'X1971', 'X197
2', 'X1973', 'X1974', 'X1975', 'X1976', 'X1977', 'X1978', 'X1979', 'X1980', 'X1981', 'X1982', 'X1983', 'X1984', 'X1985', 'X1986', 'X1987', 'X1988', 'X1989', 'X1990', 'X1991', 'X1992', 'X1993', 'X1994', 'X1995', 'X1996', 'X1997', 'X199
8', 'X1999', 'X2000', 'X2001', 'X2002', 'X2003', 'X2004', 'X2005', 'X2006', 'X2007', 'X2008', 'X2009', 'X2010', 'X2011', 'X2012', 'X2013', 'X2014', 'X2015', 'X2016', 'X2017', 'X2018', 'X2019', 'X2020', 'X2021', 'X2022', 'X2023', 'X202
4', 'X2025', 'X2026', 'X2027', 'X2028', 'X2029', 'X2030', 'X2031', 'X2032', 'X2033', 'X2034', 'X2035', 'X2036', 'X2037', 'X2038', 'X2039', 'X2040', 'X2041', 'X2042', 'X2043', 'X2044', 'X2045', 'X2046', 'X2047', 'X2048', 'X2049', 'X205
0', 'X2051', 'X2052', 'X2053', 'X2054', 'X2055', 'X2056', 'X2057', 'X2058', 'X2059', 'X2060', 'X2061', 'X2062', 'X2063', 'X2064', 'X2065', 'X2066', 'X2067', 'X2068', 'X2069', 'X2070', 'X2071', 'X2072', 'X2073', 'X2074', 'X2075', 'X207
6', 'X2077', 'X2078', 'X2079', 'X2080', 'X2081', 'X2082', 'X2083', 'X2084', 'X2085', 'X2086', 'X2087', 'X2088', 'X2089', 'X2090', 'X2091', 'X2092', 'X2093', 'X2094', 'X2095', 'X2096', 'X2097', 'X2098', 'X2099', 'X2100', 'X2101', 'X210
2', 'X2103', 'X2104', 'X2105', 'X2106', 'X2107', 'X2108', 'X2109', 'X2110', 'X2111', 'X2112', 'X2113', 'X2114', 'X2115', 'X2116', 'X2117', 'X2118', 'X2119', 'X2120', 'X2121', 'X2122', 'X2123', 'X2124', 'X2125', 'X2126', 'X2127', 'X212
8', 'X2129', 'X2130', 'X2131', 'X2132', 'X2133', 'X2134', 'X2135', 'X2136', 'X2137', 'X2138', 'X2139', 'X2140', 'X2141', 'X2142', 'X2143', 'X2144', 'X2145', 'X2146', 'X2147', 'X2148', 'X2149', 'X2150', 'X2151', 'X2152', 'X2153', 'X215
4', 'X2155', 'X2156', 'X2157', 'X2158', 'X2159', 'X2160', 'X2161', 'X2162', 'X2163', 'X2164', 'X2165', 'X2166', 'X2167', 'X2168', 'X2169', 'X2170', 'X2171', 'X2172', 'X2173', 'X2174', 'X2175', 'X2176', 'X2177', 'X2178', 'X2179', 'X218
0', 'X2181', 'X2182', 'X2183', 'X2184', 'X2185', 'X2186', 'X2187', 'X2188', 'X2189', 'X2190', 'X2191', 'X2192', 'X2193', 'X2194', 'X2195', 'X2196', 'X2197', 'X2198', 'X2199', 'X2200', 'X2201', 'X2202', 'X2203', 'X2204', 'X2205', 'X220
6', 'X2207', 'X2208', 'X2209', 'X2210', 'X2211', 'X2212', 'X2213', 'X2214', 'X2215', 'X2216', 'X2217', 'X2218', 'X2219', 'X2220', 'X2221', 'X2222', 'X2223', 'X2224', 'X2225', 'X2226', 'X2227', 'X2228', 'X2229', 'X2230', 'X2231', 'X223
2', 'X2233', 'X2234', 'X2235', 'X2236', 'X2237', 'X2238', 'X2239', 'X2240', 'X2241', 'X2242', 'X2243', 'X2244', 'X2245', 'X2246', 'X2247', 'X2248', 'X2249', 'X2250', 'X2251', 'X2252', 'X2253', 'X2254', 'X2255', 'X2256', 'X2257', 'X225
8', 'X2259', 'X2260', 'X2261', 'X2262', 'X2263', 'X2264', 'X2265', 'X2266', 'X2267', 'X2268', 'X2269', 'X2270', 'X2271', 'X2272', 'X2273', 'X2274', 'X2275', 'X2276', 'X2277', 'X2278', 'X2279', 'X2280', 'X2281', 'X2282', 'X2283', 'X228
4', 'X2285', 'X2286', 'X2287', 'X2288', 'X2289', 'X2290', 'X2291', 'X2292', 'X2293', 'X2294', 'X2295', 'X2296', 'X2297', 'X2298', 'X2299', 'X2300', 'X2301', 'X2302', 'X2303', 'X2304', 'X2305', 'X2306', 'X2307', 'X2308', 'y']

Variable clase identificada: 'y' con valores [1, 2, 3, 4]

Valores faltantes totales: 0

Top 10 genes con mayor diferencia absoluta de medias (clase 2 vs 4):
X187     3.323151
X509     2.906537
X2046    2.424515
X2050    2.401783
X129     2.165185
X1645    2.065460
X1319    2.045941
X1955    2.037340
X1003    2.011337
X246     1.837830
dtype: float64
```

Análisis:

Las diferencias en la expresión de genes como X187 y X509 entre los tipos de cáncer 2 y 4 sugieren que podrían ser clave para distinguir ambos tumores, ya sea por medio de diagnósticos o por medio de estudios. En este estudio de inferencia, encotrar esto nos da pistas valiosas sobre las diferencias biológicas entre estos cánceres, pero es importante confirmarlos con análisis estadísticos más sobustos como con pruebas de significancia o validación cruzada, ya que por ahora solo sabemos que indican una asociación interesante que requiere más investigación para establecer su verdadero significado clínico y biológico.

2. Calcularemos el estadístico t y el p-value para comparar las medias de todos los genes entre la clase 2 y la clase 4 de la base de datos. Usaremos la metodología de Bonferroni, de Holm, y de Benjamini-Hochberg para corregir por múltiples pruebas.

Se indicará para cada una qué genes tienen una expresión significativamente distinta entre las clases manejando un control de 0.05.

In [34]:

```python
from scipy import stats
from statsmodels.stats.multitest import multipletests

# Filtrar datos para clases 2 y 4
class_2 = X[y == 2]
class_4 = X[y == 4]

# Calcular estadístico t y p-values para cada gen
t_stats, p_values = stats.ttest_ind(class_2, class_4, axis=0, equal_var=False)

# Correcciones para múltiples pruebas
methods = ['bonferroni', 'holm', 'fdr_bh']  # Benjamini-Hochberg es 'fdr_bh'
significant_genes = {}

for method in methods:
    # Aplicar corrección
    rejected, corrected_p, _, _ = multipletests(p_values, alpha=0.05, method=method)

    # Guardar resultados
    significant_genes[method] = {
        'total_significant': sum(rejected),
        'genes': X.columns[rejected],
        'p_values': corrected_p[rejected]
    }

    # Imprimir resumen
    print(f"\nMétodo {method.upper()}:")
    print(f"Genes significativos: {significant_genes[method]['total_significant']}")
    print("Top 10 genes más significativos:")
    top_10_idx = np.argsort(significant_genes[method]['p_values'])[:10]
    for idx in top_10_idx:
        gene = significant_genes[method]['genes'][idx]
        p_val = significant_genes[method]['p_values'][idx]
        print(f"{gene}: p = {p_val:.5f}")
```

```
Método BONFERRONI:
Genes significativos: 72
Top 10 genes más significativos:
X1003: p = 0.00000
X187: p = 0.00000
X2050: p = 0.00000
X1955: p = 0.00000
X1645: p = 0.00000
X246: p = 0.00000
X2046: p = 0.00000
X509: p = 0.00000
X1954: p = 0.00000
X1389: p = 0.00000

Método HOLM:
Genes significativos: 72
Top 10 genes más significativos:
X1003: p = 0.00000
X187: p = 0.00000
X2050: p = 0.00000
X1955: p = 0.00000
X1645: p = 0.00000
X246: p = 0.00000
X2046: p = 0.00000
X509: p = 0.00000
X1954: p = 0.00000
X1389: p = 0.00000

Método FDR_BH:
Genes significativos: 296
Top 10 genes más significativos:
X1003: p = 0.00000
X187: p = 0.00000
X2050: p = 0.00000
X1955: p = 0.00000
X1645: p = 0.00000
X246: p = 0.00000
X2046: p = 0.00000
X509: p = 0.00000
X1954: p = 0.00000
X1389: p = 0.00000
```

Análisis:

Bonferroni con un criterio más conservador identificó 72 genes con diferencias significativas (p<0.05), incluyendo X1003, X187 y X2050 como los más relevantes p=0. Holm parecido a Bonferroni pero ligeramente menos estricto, también detectó 72 genes significativos, coincidiendo en los primeros 10 genes con Bonferroni. FDR_BH más felxible encontró 296 genes diferenciales, manteniendo los mismos 10 genes top con p=0, pero incluyendo muchos más candidatos posibles. Todos los métodos encuentran que X1003, X187 y X2050 son los genes con diferencias más extremas entre clases.

3. Realizaremos un experimento parecido al anterior, pero ahora comparando las medias de las 4 clases de la base de datos. Para lograrlo, en vez de trabajar con el estadístico t, probaremos el análisis de varianza ANOVA. Dicha prueba la realizaremos con la funciónf_oneway de scipy.stats, pero para usar la función, necesitaremos primero estratificarlos por clase.

In [36]:

```python
from scipy.stats import f_oneway

# Preparar datos para ANOVA
anova_results = []
for gene in X.columns:
    # Separar datos por clase
    groups = [X[y == cls][gene] for cls in sorted(y.unique())]

    # Calcular ANOVA
    f_stat, p_val = f_oneway(*groups)
    anova_results.append({'gene': gene, 'F': f_stat, 'p': p_val})

# Convertir a DataFrame y ordenar
anova_df = pd.DataFrame(anova_results).sort_values('p')

# Corrección Benjamini-Hochberg
rejected, corrected_p, _, _ = multipletests(anova_df['p'], alpha=0.05, method='fdr_bh')
anova_df['significant'] = rejected
anova_df['corrected_p'] = corrected_p

# Resultados
print(f"\nANOVA - Genes significativos: {sum(rejected)}")
print("\nTop 10 genes más significativos:")
print(anova_df[['gene', 'F', 'corrected_p']].head(10))
```

```
ANOVA - Genes significativos: 1162

Top 10 genes más significativos:
       gene          F   corrected_p
1954  X1955  84.364086  2.045755e-21
1388  X1389  83.817537  2.045755e-21
1002  X1003  77.795622  1.245542e-20
2049  X2050  69.230799  2.731346e-19
245    X246  68.414042  3.062126e-19
741    X742  65.572797  8.445542e-19
0        X1  59.118264  1.265852e-17
2161  X2162  56.987623  2.986387e-17
1953  X1954  55.419914  5.597246e-17
1644  X1645  54.768403  6.751740e-17
```

Análisis:

Ahora, mientras los métodos de comparación directa entre las clases 2 y 4 (Bonferroni, Holm y FDR_BH) identificaron entre 72 y 296 genes significativos, ANOVA que compara simultáneamente detectó un conjunto mucho más amplio, alrededor de 1,162 genes, indicando genes como X1955 (p=2.04e-21) y X1389 como los más significativos.

Aunque se mantiene cierta coincidencia en genes top como X1955, X1003, X2050, X1645 que aparecían en ambos análisis, el ANOVA revela genes adicionales relevantes, como X742 y X1, que las comparaciones por pares no capturaron, sugiriendo que estos podrían estar expresados diferente en otros contrastes entre clases o mostrar patrones complejos que solo se podrían ver al considerar todas las clases simultáneamente.

4. Separaremos los datos en entrenamiento y prueba, construiremos y entrenaremos un modelo de SVM con un kernel lineal, con un kernel polinomial de orden 3, y con un kernel radial. Para evitar que el tiempo de procesamiento sea exagerado, seleccionaremos solamente algunas variables, partiendo de los resultados que obtuvimos en los puntos anteriores. Pero lo ideal sería que la selección de características se basara solamente en experimentos realizados con los datos de entrenamiento. Pero, en este caso, obviaremos este detalle.

In [38]:

```python
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler

# Seleccionar los 50 genes más significativos del ANOVA
selected_genes = anova_df.head(50)['gene'].values
X_selected = X[selected_genes]

# Estandarizar datos
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_selected)

# Dividir datos (70% entrenamiento, 30% prueba)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3, random_state=42)

# Modelos SVM
models = {
    'Lineal': SVC(kernel='linear', C=1, random_state=42),
    'Polinomial (grado 3)': SVC(kernel='poly', degree=3, C=1, random_state=42),
    'RBF': SVC(kernel='rbf', C=1, gamma='scale', random_state=42)
}

# Entrenamiento
for name, model in models.items():
    print(f"\nEntrenando modelo {name}...")
    model.fit(X_train, y_train)
```

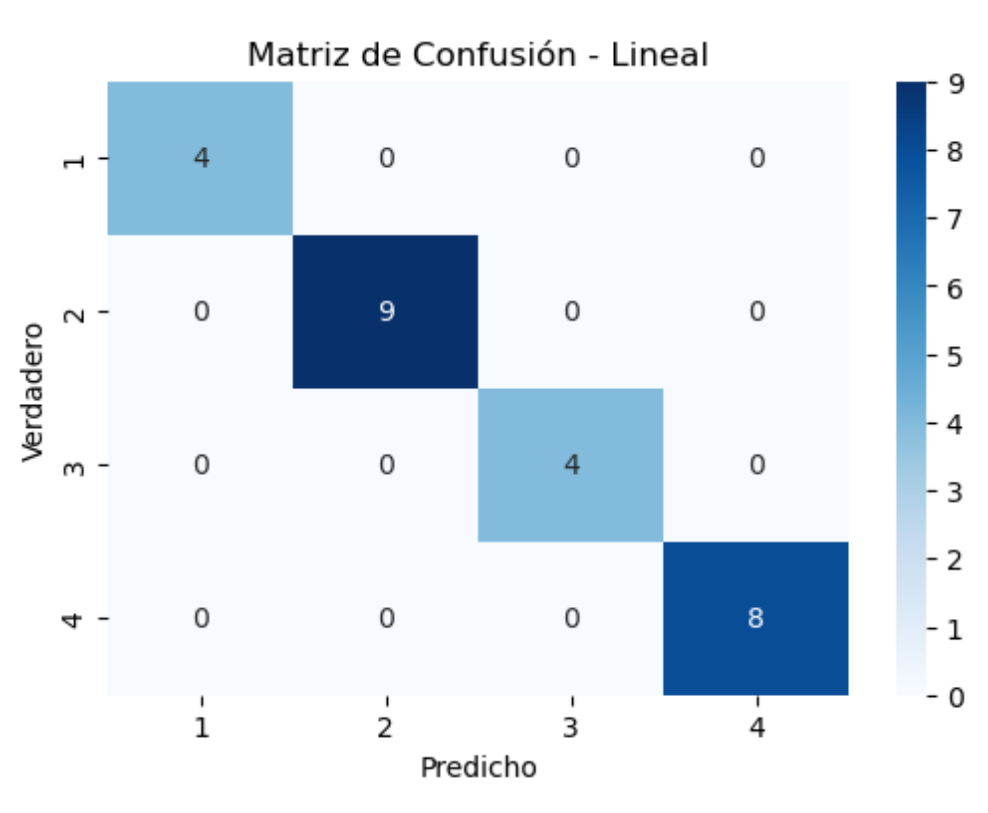
```
Entrenando modelo Lineal...

Entrenando modelo Polinomial (grado 3)...

Entrenando modelo RBF...
```

5. Calcularemos, para los 3 modelos, métricas importantes como exactitud (accuracy), precisión, recall (sensibilidad), F1-score y matriz de confusión para comparar los desempeños. analizaremos los resultados, especificando si uno de los kernels es mejor para esta tarea específica, considerando tanto el rendimiento como la complejidad computacional.

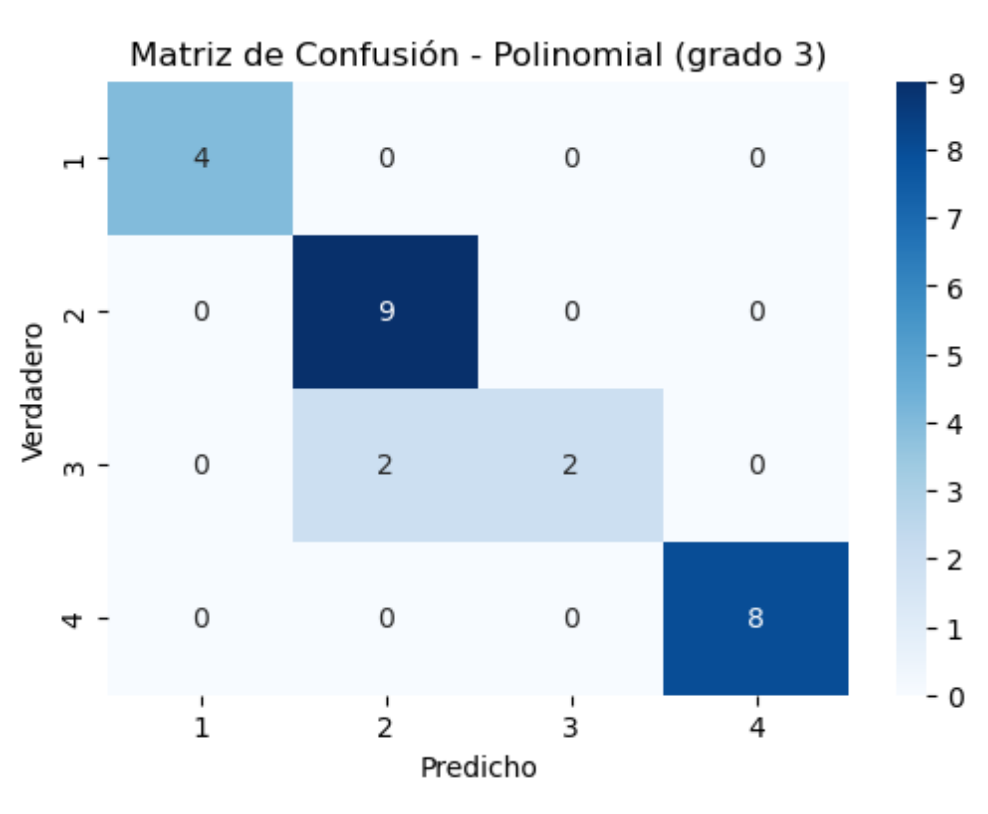
In [40]:

```python
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
import seaborn as sns
import matplotlib.pyplot as plt

for name, model in models.items():
    print(f"\n=== Evaluación modelo {name} ===")

    # Predicciones
    y_pred = model.predict(X_test)

    # Métricas
    print(f"Exactitud: {accuracy_score(y_test, y_pred):.4f}")
    print("\nReporte de clasificación:")
    print(classification_report(y_test, y_pred))

    # Matriz de confusión
    cm = confusion_matrix(y_test, y_pred)
    plt.figure(figsize=(6,4))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
                xticklabels=sorted(y.unique()),
                yticklabels=sorted(y.unique()))
    plt.title(f'Matriz de Confusión - {name}')
    plt.ylabel('Verdadero')
    plt.xlabel('Predicho')
    plt.show()
```

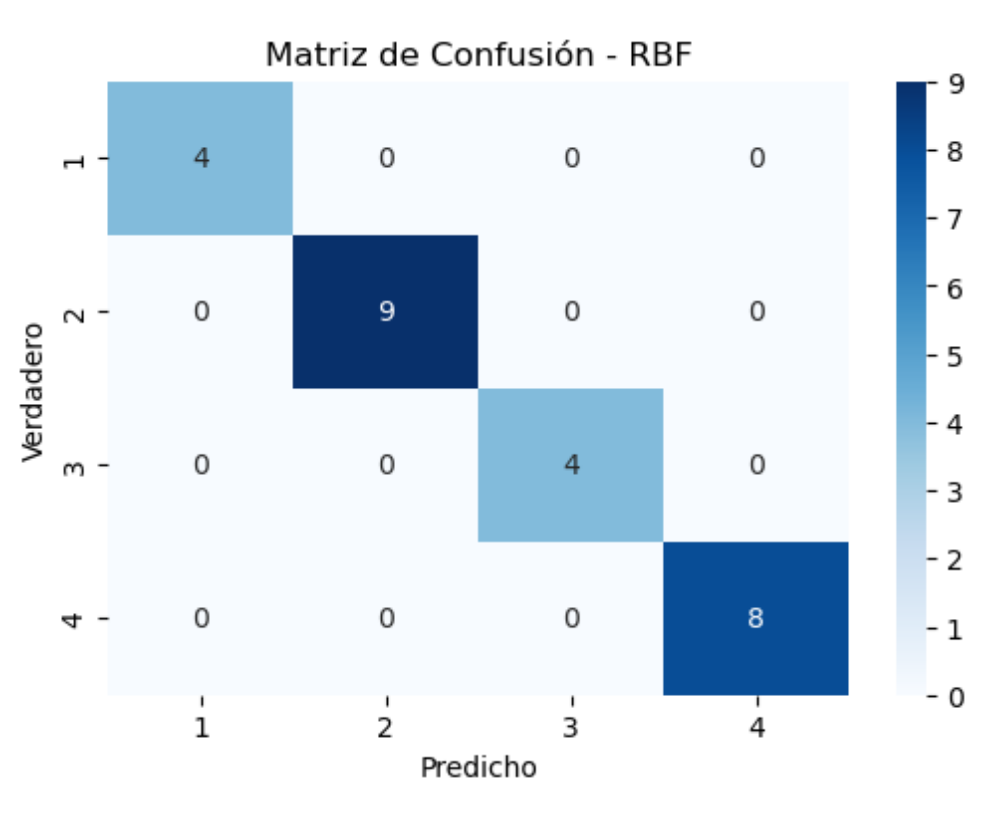
```
=== Evaluación modelo Lineal ===
Exactitud: 1.0000

Reporte de clasificación:
              precision    recall  f1-score   support

           1       1.00      1.00      1.00         4
           2       1.00      1.00      1.00         9
           3       1.00      1.00      1.00         4
           4       1.00      1.00      1.00         8

    accuracy                           1.00        25
   macro avg       1.00      1.00      1.00        25
weighted avg       1.00      1.00      1.00        25
```



```
=== Evaluación modelo Polinomial (grado 3) ===
Exactitud: 0.9200

Reporte de clasificación:
              precision    recall  f1-score   support

           1       1.00      1.00      1.00         4
           2       0.82      1.00      0.90         9
           3       1.00      0.50      0.67         4
           4       1.00      1.00      1.00         8

    accuracy                           0.92        25
   macro avg       0.95      0.88      0.89        25
weighted avg       0.93      0.92      0.91        25
```



```
=== Evaluación modelo RBF ===
Exactitud: 1.0000

Reporte de clasificación:
              precision    recall  f1-score   support

           1       1.00      1.00      1.00         4
           2       1.00      1.00      1.00         9
           3       1.00      1.00      1.00         4
           4       1.00      1.00      1.00         8

    accuracy                           1.00        25
   macro avg       1.00      1.00      1.00        25
weighted avg       1.00      1.00      1.00        25
```



Análisis:

Tanto el modelo lineal como el RBF alcanzaron una exactitud perfecta (100%), clasificando correctamente todos los casos (F1-score=1.00 en todas las clases), lo que sugiere una separabilidad óptima de los datos con estos kernels. El modelo polinomial (grado 3) mostró un rendimiento ligeramente inferior (92% exactitud), con dificultades específicas en la clase 3 (recall=0.50, F1=0.67), indicando posibles limitaciones para capturar patrones complejos en esta categoría. La consistencia del RBF y el lineal en todas las métricas (precisión, recall y F1) los posiciona como los mejores candidatos, aunque el kernel lineal podría preferirse por su simplicidad computacional al ofrecer el mismo rendimiento perfecto. El polinomial, aunque útil, requeriría ajustes (ej. modificar el grado o parámetros de regularización) para mejorar su capacidad predictiva en la clase 3.

Mientras que los resultados sugieren que el lineal y RBF son óptimos, la perfección es estadísticamente sospechosa. El polinomial (92%) podría ser más realista, y su error en la clase 3 merece investigación. Yo personalemente de debo la exactitud del 100% a la selexxipon d genomas y la fuga de datos que tenemos en el código que permite correr más rápido el sistema.

Referencias:

OpenAI. (2025). ChatGPT (versión GPT-4) [Modelo de lenguaje grande]. https://chat.openai.com/