

## PRUEBA TECNITCA QA AUTOMATION

### JUAN DAVID CORONADO DUSSÁN

- COMPONENTE PRÁCTICO (Funcional)

Creación de la tabla Students

CREATE TABLE Students (

```
student_id INT,  
name VARCHAR(100),  
birthdate DATE,  
gender CHAR(1),  
email VARCHAR(100),  
phone VARCHAR(20)
```

);

```
INSERT INTO Students (student_id, name, birthdate, gender, email, phone) VALUES  
(9000185, 'PEDRO GOMEZ', '1995-10-30', 'M', 'PEDRO.GOMEZ@GMAIL.COM', '1185432478'),  
(9000186, 'JUAN HERNANDEZ', '1995-01-25', 'M', 'Juan.Hernandez@gmail.com', '1185432479'),  
(9000187, 'MARIA GONZALEZ', '1998-08-01', 'F', 'Maria.Gonzalez@hotmail.com', '1185432480'),  
(9000188, 'PABLO SANCHEZ', '1997-03-09', 'M', 'pablo.sanchez@gmail.com', '1185432481'),  
(9000189, 'ELENA CASTRO', '2000-06-15', 'F', 'Elena.Castro@gmail.com', '1185432482'),  
(9000190, 'MATIAS PEREZ', '1998-09-06', 'M', 'Matias.Perez@gmail.com', '1185432483'),  
(9000191, 'SEBASTIAN VARGAS', '2002-08-25', 'M', 'Sebastian.Vargas@outlook.com',  
'1185432484'),  
(9000192, 'PEDRO HERNANDEZ', '1998-05-19', 'M', 'Pedro.Hernandez92@gmail.com',  
'1185432485'),  
(9000193, 'CAROLINA LUNA', '1997-07-21', 'F', 'Carolina.Luna@yahoo.com', '1185432486'),  
(9000194, 'MONICA FIGUEROA', '1980-06-18', 'F', 'Monica.Figueroa@hotmail.com', '1185432487');
```

SELECT \* FROM Students;

The screenshot shows the Paiza.io MySQL web interface. The top navigation bar includes links for 'Nuevo código', 'Códigos recientes', 'Web Dev(PaizaCloud)', 'Spanish', 'Registrarse', and 'Iniciar sesión'. The main area displays the executed SQL commands and their results. The first command creates the 'Students' table with columns: student\_id INT, name VARCHAR(100), birthdate DATE, gender CHAR(1), email VARCHAR(100), and phone VARCHAR(20). The second command inserts 14 rows of data into the table. The results section shows a table with 14 rows and 6 columns: student\_id, name, birthdate, gender, email, and phone. The data matches the values provided in the INSERT statement.

student_id	name	birthdate	gender	email	phone
9000185	PEDRO GOMEZ	1995-10-30	M	PEDRO.GOMEZ@GMAIL.COM	1185432478
9000186	JUAN HERNANDEZ	1995-01-25	M	Juan.Hernandez@gmail.com	1185432479
9000187	MARIA GONZALEZ	1998-08-01	F	Maria.Gonzalez@hotmail.com	1185432480
9000188	PABLO SANCHEZ	1997-03-09	M	pablo.sanchez@gmail.com	1185432481
9000189	ELENA CASTRO	2000-06-15	F	Elena.Castro@gmail.com	1185432482
9000190	MATIAS PEREZ	1998-09-06	M	Matias.Perez@gmail.com	1185432483
9000191	SEBASTIAN VARGAS	2002-08-25	M	Sebastian.Vargas@outlook.com	1185432484
9000192	PEDRO HERNANDEZ	1998-05-19	M	Pedro.Hernandez92@gmail.com	1185432485
9000193	CAROLINA LUNA	1997-07-21	F	Carolina.Luna@yahoo.com	1185432486
9000194	MONICA FIGUEROA	1980-06-18	F	Monica.Figueroa@hotmail.com	1185432487

creación de la tabla Career\_Student

CREATE TABLE Career\_Student (

```
career_id INT,  
student_id INT,  
start_date DATE,  
estimated_end_date DATE,
```

```
end_date DATE,  
extended VARCHAR(10),  
calification INT  
);
```

```
INSERT INTO Career_Student (career_id, student_id, start_date, estimated_end_date, end_date,
extended, calification) VALUES
(10025, 9000185, '2009-01-01', '2010-01-01', '2013-01-01', NULL, 85),
(10023, 9000186, '2018-01-01', '2021-01-01', NULL, NULL, 92),
(10026, 9000187, '2014-01-01', '2016-01-01', NULL, NULL, 58),
(10027, 9000188, '2017-01-01', '2020-01-01', NULL, NULL, 67),
(10028, 9000189, '2016-01-01', '2018-01-01', '2018-01-01', NULL, 91),
(10025, 9000190, '2019-01-01', '2022-01-01', NULL, NULL, 89),
(10026, 9000191, '2020-01-01', '2023-01-01', NULL, NULL, 75),
(10029, 9000192, '2021-01-01', '2024-01-01', NULL, NULL, 74),
(10025, 9000193, '2022-01-01', '2025-01-01', NULL, NULL, 91),
(10029, 9000185, '2008-01-01', '2013-01-01', '2013-01-01', NULL, 68);
```

```
SELECT * FROM Career_Student;
```

[illegible]

## Creación de la tabla Careers

```
CREATE TABLE Careers (
    career_id INT,
    name VARCHAR(100),
    description VARCHAR(100),
    subjects INT,
    cost DECIMAL(10,2)
);
```

```
INSERT INTO Careers (career_id, name, description, subjects, cost) VALUES
```

```
(10025, 'INGENIERIA', 'INGENIERIA', 90, 25000.00),
(10026, 'ADMINISTRACION', 'ADMINISTRACION', 72, 18000.00),
(10027, 'CONTABILIDAD', 'CONTABILIDAD', 60, 15000.00),
(10028, 'DERECHO', 'DERECHO', 85, 23000.00),
(10029, 'INFORMATICA', 'INFORMATICA', 78, 19500.00),
(10030, 'COMUNICACION', 'COMUNICACION', 65, 17800.00);
```

SELECT \* FROM Careers;

The screenshot shows a MySQL IDE interface. The top panel displays the SQL code for creating a table and inserting data. The bottom panel shows the results of the SELECT query.

```

104 CREATE TABLE Careers (
105   career_id INT,
106   name VARCHAR(100),
107   description VARCHAR(100),
108   subjects INT,
109   cost DECIMAL(10,2)
110 );
111
112 INSERT INTO Careers (career_id, name, description, subjects, cost) VALUES
113 (10025, 'INGENIERIA', 'INGENIERIA', 90, 25000.00),
114 (10026, 'ADMINISTRACION', 'ADMINISTRACION', 72, 18000.00),
115 (10027, 'CONTABILIDAD', 'CONTABILIDAD', 60, 15000.00),
116 (10028, 'DERECHO', 'DERECHO', 85, 23000.00),
117 (10029, 'INFORMATICA', 'INFORMATICA', 78, 19500.00),
118 (10030, 'COMUNICACION', 'COMUNICACION', 65, 17800.00);
119
120 SELECT * FROM Careers;

```

Salida	Entrada	Comments
career_id	name	description
10025	INGENIERIA	INGENIERIA
10026	ADMINISTRACION	ADMINISTRACION
10027	CONTABILIDAD	CONTABILIDAD
10028	DERECHO	DERECHO
10029	INFORMATICA	INFORMATICA
10030	COMUNICACION	COMUNICACION

1. Redactar las siguientes consultas en SQL

a) Seleccione el total de estudiantes Masculinos Registrados

SELECT COUNT(\*) AS total\_masculinos FROM Students WHERE gender = 'M';

The screenshot shows a MySQL IDE interface. The top panel displays the SQL query. The bottom panel shows the results of the query.

```

110
111
112
113
114 SELECT COUNT(*) AS total_masculinos FROM Students WHERE gender = 'M';
115
116
117
118
119

```

Ejecutar (Ctrl-Enter)

Salida	Entrada	Comments
total_masculinos		
6		

b) Elimine los estudiantes nacidos antes 2000

DELETE FROM Students WHERE birthdate < '2000-01-01';

```
111
112
113
114 DELETE FROM Students WHERE birthdate < '1990-01-01';
115
116
117
118
119
```

Ejecutar (Ctrl-Enter)

Salida Entrada Comentarios

(1.37 sec)

- c) Actualice la descripción de la carrera con id '10026' de "Administración" a "Administración de Empresas"

UPDATE Careers SET description = 'Administración de Empresas' WHERE career\_id = 10026;  
SELECT \* FROM Careers;

```
112
113
114 UPDATE Careers SET description = 'Administración de Empresas' WHERE career_id = 10026;
115
116
117 SELECT * FROM Careers;
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
```

Ejecutar (Ctrl-Enter)

Salida Entrada Comentarios

(1.37 sec)

career_id	name	description	subjects	cost
10025	INGENIERIA	INGENIERIA	90	25000.00
10026	ADMINISTRACION	Administración de Empresas	72	18000.00
10027	CONTABILIDAD	CONTABILIDAD	60	15000.00
10028	DERECHO	DERECHO	85	23000.00
10029	INFORMATICA	INFORMATICA	78	19500.00
10030	COMUNICACION	COMUNICACION	65	17000.00

- d) Muestre los nombres de los estudiantes que estén cursando dos o más materias

SELECT s.name FROM Students s JOIN Career\_Student cs ON s.student\_id = cs.student\_id  
GROUP BY s.student\_id, s.name HAVING COUNT(cs.career\_id) >= 2;

```
112
113
114
115 SELECT s.name FROM Students s JOIN Career_Student cs ON s.student_id = cs.student_id GROUP BY s.student_id, s.name HAVING COUNT(cs.career_id) >= 2;
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
```

Ejecutar (Ctrl-Enter)

Salida Entrada Comentarios

(1.37 sec)

name
PEDRO GOMEZ

- e) Calificación promedio de todos los estudiantes que estén cursando carreras

SELECT AVG(calification) AS promedio\_calificaciones FROM Career\_Student;

```
128
129
130
131
132
133
134
135 SELECT AVG(calification) AS promedio_calificaciones FROM Career_Student;
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
```

Ejecutar (Ctrl-Enter)

Salida Entrada Comentarios

(1.48 sec)

promedio_calificaciones
79.0000

- f) Muestra el nombre, carrera id, correos de los estudiantes que tienen una calificación mayor 70

SELECT s.name, cs.career\_id, s.email FROM Students s JOIN Career\_Student cs ON  
s.student\_id = cs.student\_id WHERE cs.calification > 70;

```
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
```

SELECT s.name, cs.career\_id, s.email FROM Students s JOIN Career\_Student cs ON s.student\_id = cs.student\_id WHERE cs.career\_id >= 2

Ejecutar (Ctrl+Enter)

Salida	Entrada	Comments
name	career_id	email
PEDRO GOMEZ	10025	Pedro.Gomez@gmail.com
JUAN HERNANDEZ	10023	Juan.Hernandez@gmail.com
ELENA CASTRO	10028	Elena.Castro@gmail.com
MATIAS PEREZ	10025	Matias.Perez@gmail.com
SEBASTIAN VARGAS	10026	Sebastian.Vargas@outlook.com
PEDRO HERNANDEZ	10029	Pedro.Hernandez92@gmail.com
CAROLINA LUNA	10025	Carolina.Luna@yahoo.com

g) Cuántos estudiantes que estén cursando 2 o más carreras  
SELECT COUNT(\*) AS estudiantes\_multicarrera FROM ( SELECT student\_id FROM Career\_Student GROUP BY student\_id HAVING COUNT(career\_id) >= 2 ) AS subquery;

```
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
```

SELECT COUNT(\*) AS estudiantes\_multicarrera FROM ( SELECT student\_id FROM Career\_Student GROUP BY student\_id HAVING COUNT(career\_id) >= 2 ) AS subquery;

Ejecutar (Ctrl+Enter)

Salida	Entrada	Comments
estudiantes_multicarrera		
1		

2. De la siguiente imagen, reporte al menos un bug que incluya todos los datos que considere relevantes para que un desarrollador pueda replicarlo y solucionarlo

ID del Bug: BUG-PAGO-001

Título: Duplicación del método de pago "RappiCuenta" en la pantalla de selección

Fecha: 2025-10-04

Reportado por: QA Tester

Descripción:

En la pantalla de selección de métodos de pago, el método "RappiCuenta" aparece listado dos veces: una como "RappiCuenta o débito" y otra como "RappiCuenta", ambos con la misma descripción: "¡Pago fácil e inmediato!". Esta duplicación puede generar confusión en el usuario respecto a cuál opción debe seleccionar y sugiere un problema de diseño o lógica de renderizado.

Pasos para reproducir:

1. Abrir la app de Rappi.
2. Iniciar un pago.
3. Proceder a la pantalla de selección de método de pago.
4. Observar las opciones listadas.

Resultado esperado:

Cada método de pago debería aparecer una sola vez, con su nombre claro y sin ambigüedad, evitando duplicados innecesarios o confusos.

Resultado actual:

Se presentan dos opciones visualmente similares:

- "RappiCuenta o débito"
- "RappiCuenta"

Ambas con descripciones idénticas y sin aclarar sus diferencias, si las hay.

Entorno:

- Dispositivo: Android / IOS
- Versión de la app: 1.7.3
- Pantalla: Método de pago

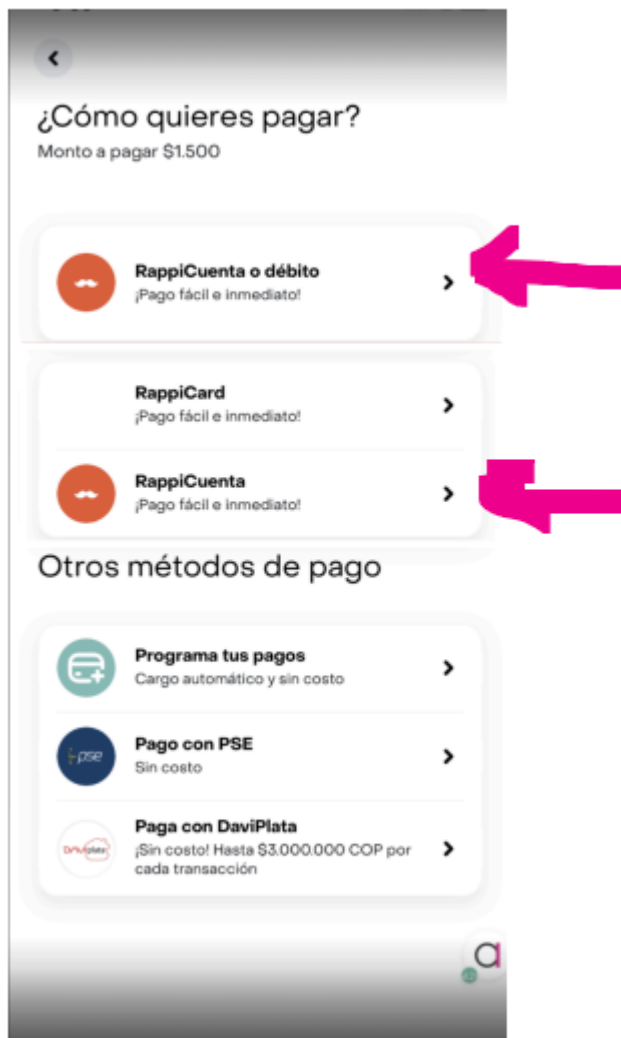
Severidad: Media

- Puede causar confusión en el usuario, pero no bloquea el flujo de pago.

Prioridad: Alta

- Es un tema de experiencia de usuario que puede impactar negativamente la percepción del sistema y generar errores en la selección del método de pago.

Imagen del bugs:



- COMPONENTE TEÓRICO (Automation)

Basado en su experiencia, responda las siguientes preguntas:

1. ¿Cuáles son las principales diferencias entre automatizar en Android vs iOS con Appium?

ASPECTO	IOS	ANDROID
Requisitos del Sistema Operativo	Solo funciona en macOS (por requerimiento de Xcode).	Puede ejecutarse en Windows, macOS o Linux.
Instalación del entorno	Necesitas Xcode, Xcode Command Line Tools, y un perfil de desarrollador.	Necesitas Android SDK, Java JDK, emuladores, y configurar variables de entorno.
Dispositivos y Emuladores	Emuladores más limitados; requieren simuladores Xcode y perfiles de confianza.	Fácil de ejecutar en emuladores sin restricciones.
Interacción con el sistema	Permite más acciones (ej: manipular notificaciones, cambiar red)	Más limitado por seguridad del sistema
Firma de Aplicaciones	Puedes usar APKs sin firmar o firmarlos tú mismo.	Las apps deben estar firmadas con certificados válidos y, para pruebas en dispositivos reales, deben estar en el perfil del equipo de desarrollo.

2. ¿Cómo diseñaría un framework de automatización para mobile multiplataforma que sea mantenible y escalable?

Lenguaje: JavaScript (con WebdriverIO), Java, o Python, según stack del equipo.

Serenity BDD: Ideal para automatización de pruebas con enfoque BDD, reporting robusto, integración con Cucumber.

Appium: Para automatización móvil multiplataforma (Android + iOS).

Cucumber: Para escribir pruebas en lenguaje natural (Gherkin) y facilitar colaboración con stakeholders.

Una posible estructura (Serenity)

```

├ README.md      # Documento principal del proyecto
├ azure.yml/gitaction.yml      # Archivo pipeline
├ .gitignore      # configurar el archivo que no se deben subir o información que no se debe subir
├ build.gradle/pom.xml #Dependencias
├ serenity.conf      # Configuración Serenity
├ src
├   main
├     java
├       ├── tasks      # Acciones complejas que realiza el actor
├       ├── questions  # Validaciones / consultas sobre el estado de la app
├       ├── ui          # PageObjects y localizadores multiplataforma
├       ├── utils       # Utilidades
├       └── models      # Clases modelo de datos para pruebas

```

```

└─ test
  └─ java
    ├── stepdefinitions      # Implementación de los pasos definidos en las features
    │   ├── Hooks.java
    │   ├── LoginStepDefinitions.java
    │   └─ OtrosStepDefinitions.java
    └─ runners              # Test runners para Serenity + Cucumber
        └─ TestRunner.java
└─ resources
    └─ features              # Archivos .feature con escenarios en Gherkin

```

Manejo de variable de entorno (environments)

Para la ejecución lo manejo de la siguiente manera

# Android

mvn clean test -Denvironment=android

gradle clean test -Denvironment=android

# iOS

mvn clean test -Denvironment=ios

gradle clean test -Denvironment=ios

Tendría muy en cuenta las buenas prácticas de codificación adicional manejarían principios SOLID

- Modulariza al máximo: cada task o interacción debe ser reutilizable.
- Usa naming claro (variables, funciones, clases, métodos, archivos, carpetas, etc)
- Dividir por features, no por pantallas.
- Evita lógica de negocio en los step definitions, todo debe ir en tasks/questions.
- Reutilización y abstracción en servicios y componentes.

3. ¿Qué criterios usa para seleccionar qué casos se deben automatizar y cuáles no?

#### **Criterios para automatizar:**

- Alta frecuencia de ejecución.
- Casos críticos para el negocio.
- Repetitivos y estables en el tiempo.
- Funcionalidades ya estabilizadas.
- Costos y beneficios.

#### **Criterios para NO automatizar:**

- Flujos que cambian con frecuencia.
- Procesos muy esporádicos o de bajo volumen.
- Casos que requieren interacción física (ej: huella dactilar).
- Pruebas exploratorias o de usabilidad.



4. ¿Qué estrategias aplicaría para integrar pruebas automáticas en un pipeline CI/CD?

- Cual herramienta usar para la creación del pipeline
- Diseña un pipeline con etapas claras (Build, ejecución de prueba, sonar, escaneo de vulnerabilidades)
- Ejecución por rama
- integración con granjas de dispositivos remotas
- Paralelización y optimización del tiempo de ejecución
- Resultados automáticos
- Notificaciones
- Integrar con los desarrollos

5. Mencione las ventajas y desventajas de trabajar con dispositivos físicos vs. dispositivos en remoto / granja de dispositivos cuando ejecuta una suite de automatización.

<b>Factor</b>	<b>Dispositivos Físicos</b>	<b>Granja de Dispositivos (Remoto)</b>
<b>Costo</b>	Alto (compra, mantenimiento)	Pago por uso, escalable
<b>Acceso y control</b>	Total (red, logs, debugging, gestos)	Limitado a lo que permite el proveedor
<b>Paralelismo</b>	Limitado por la cantidad de dispositivos físicos	Alta escalabilidad
<b>Configuración inicial</b>	Compleja pero controlada	Configuración ya proporcionada
<b>Integración CI/CD</b>	Requiere hardware conectado al servidor	Integración directa con servicios en la nube

6. ¿Qué patrones de diseño ha implementado en sus proyectos que considere eficaces para minimizar la duplicación de código y mejorar la legibilidad al desarrollar automatizaciones para dispositivos móviles?

- Page Object Model (POM)

- Serenity Screenplay + Cucumber

- COMPONENTE PRÁCTICO (Automation)

Contexto: Se requiere implementar un nuevo script para una prueba automatizada. El caso de prueba involucra tres (3) vistas: Home del App, Login y el Home de producto. El flujo que se desea automatizar valida que el usuario Ingrese al Home del App donde da un tap al widget del producto para posteriormente ser redireccionado al Login. Allí ingresará un código de 6 dígitos y finalmente navegará de forma automática al Home del producto donde se encuentra el detalle y acciones que el usuario puede realizar sobre el producto.

- a) En la vista de Login se implementa un método para validar que todos los labels definidos en el diseño se encuentren presentes. A continuación, verá el código implementado.

```

1  ✓ public boolean verifyLabelsOnScreen() {
2  ✓      try {
3          // Espera para que la pantalla cargue los elementos
4          Thread.sleep(15000);
5
6          boolean flagLabelIniciarSesion = false;
7          boolean flagLabelHola = false;
8          boolean flagLabelIngresarContrasena = false;
9          boolean flagLabelOlvideContrasena = false;
10
11         // Recorre todos los labels en la pantalla
12  ✓     for (MobileElement label : screenLabels) {
13         String text = label.getAttribute("text");
14
15  ✓         if (text.equals("Iniciar Sesión")) {
16             flagLabelIniciarSesion = true;
17         }
18  ✓         if (text.contains("¡Hola!")) {
19             flagLabelHola = true;
20         }
21  ✓         if (text.equals("Ingresa tu contraseña de 6 dígitos")) {
22             flagLabelIngresarContrasena = true;
23         }
24  ✓         if (text.equals("Olvidé mi contraseña")) {
25             flagLabelOlvideContrasena = true;
26         }
27     }
28
29     // Verifica que todos los labels estén presentes
30     if (flagLabelIniciarSesion && flagLabelHola
31  ✓     && flagLabelIngresarContrasena && flagLabelOlvideContrasena) {
32         return true;
33     }
34
35  ✓ } catch (Exception e) {
36     return false;
37 }
38     return false;
39 }
40

```

Instrucción: De acuerdo con su experiencia indique si está de acuerdo con la forma en la que fue implementado el método. En caso de no estar de acuerdo. Indíquenos qué modificaciones haría o reescriba el método para ver reflejados sus cambios y justifique su respuesta.

De acuerdo con mi experiencia, con la forma de implementación del código presentado tiene varias mejoras por realizar:

- `EL Thread.sleep(15000)`: es una mala practica que este siempre espere 15 segundos, ya que hace que la prueba se mas lenta, adicional a esto aun que los elementos ya estén cargados y se puedan interactuar con estos antes de 15 segundos este obliga a esperar los 15 segundos  
Para ello una buena practica seria usar los `wait` + condición ya solo espera a que el o los elementos estén cargados e interactúa inmediatamente con ellos
- El uso de las banderas booleanas repetitiva: código repetitivo y agrega complejidad, adicional en temas de escalabilidad y mantenibilidad en el tiempo se vuelve insostenible. Mejor usar una lista de textos esperados y se comparada con los textos encontrados.
- Uso de `equals`: el riesgo que genera esto es si un label tiene un espacio de más, mayúsculas/minúsculas distintas o un carácter especial, el test puede fallar. Se puede usar `contains` si el label varia o `equalsIgnoreCase` si no mitigando el riesgo.
- Código repetitivo y uso excesivo del `if` generando complejidad, mantenibilidad y escalabilidad.  
Para ello es mejor usar `Set` o `Map` con textos esperados y validarlos en un bucle.
- Uso de `return false` redundante el último `return false` (fuera del `catch`) es innecesario porque el flujo ya está cubierto.
- Falta mensajes de error si la validación falla, no sabes qué label faltó. Solo devuelve `false`, para ellos lo ideal es logs o lanzar excepción con el detalle.
- Falta de reutilización lo cual hace que no sea posible usar en otras partes del código si es requerido.
- Problema de escalabilidad ya que por ahora son 4 pero si son 20 o 50 hace que el método se vuelve enorme y difícil de leer.

- b) En la vista del Home del App, se necesita crear dos métodos: El primero estará encargado de verificar si el usuario se encuentra o no en el Home del App y el segundo realizará la acción de dar tap en el botón de Rappipay para ingresar al login. A continuación, verá la definición de los elementos que se tuvieron en cuenta.

```

1  @AndroidFindBy(id = "com.grability.rappi:id/menu_image")
2  private MobileElement buttonMenu;
3  @AndroidFindBy(id = "com.grability.rappi:id/support_image")
4  private MobileElement buttonSupport;
5  @AndroidFindBy(className = "android.widget.TextView")
6  private List<MobileElement> widgetElements;
7  @AndroidFindBy(id = "com.grability.rappi:id/imageView_action")
8  private MobileElement buttonRappiPay;
9  @AndroidFindBy(id = "layout_primary_action")
10 private MobileElement BTN_RAPPIPAY_CO;
11 @AndroidFindBy(xpath = "//*[contains(@text, 'Restaurante')]")
12 private MobileElement RESTAURANT;
13 @AndroidFindBy(xpath = "//*[contains(@text, 'Mercado/Ex')]")
14 private MobileElement MARKET;
15 @AndroidFindBy(xpath = "//*[contains(@text, 'Turbo')]")
16 private MobileElement TURBO;
17

```

Instrucción: De acuerdo con su experiencia indique si está de acuerdo con la forma en la que se definieron los elementos. En caso de no estar de acuerdo. Indíquenos qué modificaciones haría o reescriba el código para ver reflejados sus cambios y justifique su respuesta.

De acuerdo con mi experiencia, con la forma de implementación funciona, sin embargo, tiene varias mejor que se pueden realizar:

- Uso de className = "android.widget.TextView" para widgetElements: muy genérico. Capturará todos los TextView de la pantalla haciendo que guarde información no requerida.
- Xpath con //\* hace que la automatización se vuelva lenta ya que va a buscar en todo el DOM del código es mejor especificar ejemplo: //a o //h1
- Los xpath con texto son frágiles si cambia el idioma de la app (ES → EN)
- Nombre de las variables no tienen estructura, mezclas mayúsculas/abreviaturas para ello usar naming, ejemplo btnMenu.
- Los nombres de algunas variables no tienen contexto para que va a ser su uso ejemplo: private List<MobileElement> widgetElements;
- Localizadores muy específicos (com.grability.rappi:id/menu\_image) hacen que el Page Object sea poco portable si la app cambia la estructura.
- Si estamos usando localizadores como el Id no es una buena práctica com.grability.rappi:id/menu\_image es mejor colocar menu\_image