

Evidencias de funcionamiento

Capturas de pantalla o scripts de prueba ejecutados

Realizando inserts en la tabla Paciente de MySQL y Replicando

Url de Video: <https://youtu.be/8uUC5KJhB1c>

-Sin Datos en MySQL

The screenshot shows the MySQL Workbench interface. In the top navigation bar, there are tabs for 'DB-Clinic' and several 'unconnected' tabs. The left sidebar shows the 'SCHEMAS' tree, which includes the 'clinica' schema containing tables like 'bitacora', 'cita', 'clínica', 'doctor', 'historial', 'medicamento', 'paciente', 'recepta', 'receta_medicamento', and 'replicacion_control'. Below the schemas is a 'SYS' section. The main area has a tab titled 'Paciente' with a query editor containing the SQL command: 'SELECT * FROM clinica.paciente;'. The results grid shows one row of data: ID_PACIENTE (1), NOMBRE ('Juan'), FECHA_NACIMIENTO ('1990-01-01'), TELFONO ('1234567890'), EMAIL ('juan@example.com'), and GENERO ('M'). The status bar at the bottom right indicates a duration of 0.109 sec / 0.000 sec.

Activar Windows

-Sin datos en Oracle

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes 'Archivo', 'Ver', 'Navegar', 'Ejecutar', 'Equipo', 'Herramientas', 'Ventana', and 'Ayuda'. The left sidebar shows the 'Conexiones' tree, which includes 'Oracle conexiones' and 'Clinica'. The main area displays the 'PACIENTE' table structure with columns: ID_PACIENTE, NOMBRE, FECHA_NACIMIENTO, TELFONO, EMAIL, and GENERO. A query result grid below shows one row of data: ID_PACIENTE (1), NOMBRE ('Juan'), FECHA_NACIMIENTO ('1990-01-01'), TELFONO ('1234567890'), EMAIL ('juan@example.com'), and GENERO ('M'). The status bar at the bottom right indicates a duration of 0.109 sec / 0.000 sec.

Activar Windows

-Realizando Inserts en MySQL

The screenshot shows the DB-Console interface with the following details:

- Schemas:** clínica
- Tables:** paciente
- Columns:** ID_PACIENTE (PK), NOMBRE, FECHA_NACIMIENTO, TELFONO, EMAIL, GENERO
- Query 1:** Multiple INSERT statements into the paciente table, each adding a new patient record.
- Action Output:** Shows the execution log with 14 rows affected.
- Message:** 0 rows) returned / 0 rows affected
- Duration / Fetch:** 0.109 sec / 0.000 sec

```

1 • INSERT INTO paciente (id_paciente, nombre, fecha_nacimiento, telefono, email, genero)
VALUES (2001, 'Javier Torres', '1992-03-19', '334567890123', 'javier.torres@example.com', 'M');
2
3
4 • INSERT INTO paciente (id_paciente, nombre, fecha_nacimiento, telefono, email, genero)
VALUES (2002, 'Sofía Herrera', '1988-09-27', '9456789011', 'sofia.herrera@example.com', 'F');
5
6
7 • INSERT INTO paciente (id_paciente, nombre, fecha_nacimiento, telefono, email, genero)
VALUES (2003, 'Andrés Vega', '1979-01-12', '9567890123', 'andres.vega@example.com', 'M');
8
9
10 • INSERT INTO paciente (id_paciente, nombre, fecha_nacimiento, telefono, email, genero)
VALUES (2004, 'Valentina Ruiz', '1995-08-05', '9678901234', 'valentina.ruiz@example.com', 'F');
11
12
13 • INSERT INTO paciente (id_paciente, nombre, fecha_nacimiento, telefono, email, genero)
VALUES (2005, 'Felipe Navarro', '1983-04-22', '9789012345', 'felipe.navarro@example.com', 'M');
14
15
16

```

Activar Windows

-Bitácora resultante después de inserts en MySQL:

The screenshot shows the DB-Console interface with the following details:

- Schemas:** clínica
- Tables:** bitacora
- Columns:** ID_BITACORA, TABLA, OPERACION, ID_REGISTRO, DATOS, ORIGEN, REPLICADO, INTENTOS, ULTIMO_ERROR
- Query 1:** SELECT * FROM clínica.bitacora;
- Action Output:** Shows the audit log with 9 rows affected.
- Message:** 0 rows) returned / 9 rows affected
- Duration / Fetch:** 0.109 sec / 0.000 sec

ID_BITACORA	TABLA	OPERACION	ID_REGISTRO	DATOS	ORIGEN	REPLICADO	INTENTOS	ULTIMO_ERROR
61	paciente	INSERT	2001	{'email': 'javier.torres@example.com', 'genero': 'M', 'fecha_nacimiento': '1992-03-19', 'nombre': 'Javier Torres', 'telefono': '334567890123'}	MySQL	0	0	
62	paciente	INSERT	2002	{'email': 'sofia.herrera@example.com', 'genero': 'F', 'fecha_nacimiento': '1988-09-27', 'nombre': 'Sofía Herrera', 'telefono': '9456789011'}	MySQL	0	0	
63	paciente	INSERT	2003	{'email': 'andres.vega@example.com', 'genero': 'M', 'fecha_nacimiento': '1979-01-12', 'nombre': 'Andrés Vega', 'telefono': '9567890123'}	MySQL	0	0	
64	paciente	INSERT	2004	{'email': 'valentina.ruiz@example.com', 'genero': 'F', 'fecha_nacimiento': '1995-08-05', 'nombre': 'Valentina Ruiz', 'telefono': '9678901234'}	MySQL	0	0	
65	paciente	INSERT	2005	{'email': 'felipe.navarro@example.com', 'genero': 'M', 'fecha_nacimiento': '1983-04-22', 'nombre': 'Felipe Navarro', 'telefono': '9789012345'}	MySQL	0	0	
66	bitacora	INSERT	2006		MySQL	0	0	
67	bitacora	INSERT	2007		MySQL	0	0	
68	bitacora	INSERT	2008		MySQL	0	0	
69	bitacora	INSERT	2009		MySQL	0	0	

bitacora 1 <--> paciente

Output

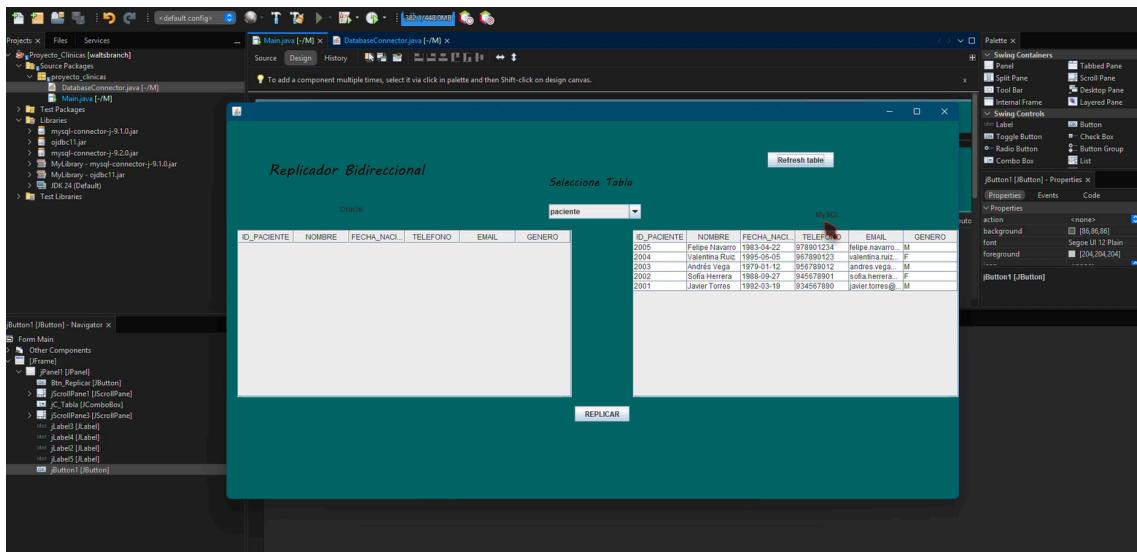
```

1 21:49:11 SELECT * FROM clínica.paciente LIMIT 0, 1000
2 21:49:15 SELECT * FROM clínica.bitacora LIMIT 0, 1000
3
4 21:49:34 INSERT INTO paciente (id_paciente, nombre, fecha_nacimiento, telefono, email, genero) VALUES (2001, 'Javier Torres', '1992-03-19', '334567890123', 'javier.torres@example.com', 'M')
5 21:49:34 INSERT INTO paciente (id_paciente, nombre, fecha_nacimiento, telefono, email, genero) VALUES (2002, 'Sofía Herrera', '1988-09-27', '9456789011', 'sofia.herrera@example.com', 'F')
6 21:49:34 INSERT INTO paciente (id_paciente, nombre, fecha_nacimiento, telefono, email, genero) VALUES (2003, 'Andrés Vega', '1979-01-12', '9567890123', 'andres.vega@example.com', 'M')
7 21:49:34 INSERT INTO paciente (id_paciente, nombre, fecha_nacimiento, telefono, email, genero) VALUES (2004, 'Valentina Ruiz', '1995-08-05', '9678901234', 'valentina.ruiz@example.com', 'F')
8 21:49:37 SELECT * FROM clínica.paciente LIMIT 0, 1000
9 21:49:41 SELECT * FROM clínica.bitacora LIMIT 0, 1000

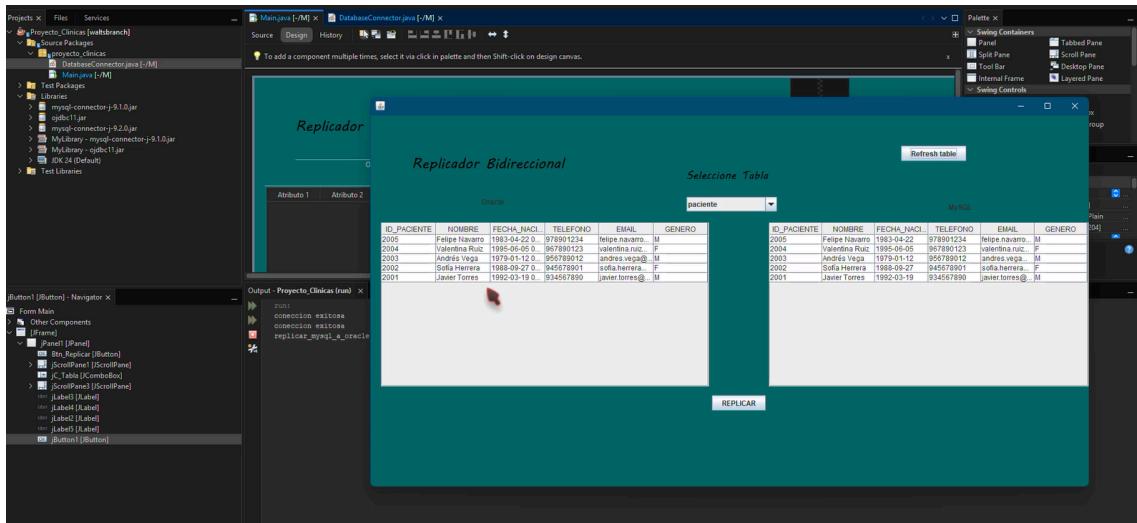
```

Activar Windows

- En el aplicativo, Antes de realizar la replicación:



- En el Aplicativo, Después de oprimir Replicación y Refresh Table



- Datos insertados en Oracle:

The screenshot shows the Oracle SQL Developer interface. On the left, the Object Navigator displays various database objects like tables, packages, and functions. The central area shows a table named 'PACIENTE' with the following data:

ID_PACIENTE	NOMBRE	FECHA_NACIMIENTO	TELEFONO	GENERO
1	2001 Javier Torres	19/03/92	934547890	javier.torres@example.com M
2	2002 Sofia Herreza	27/09/98	945478901	sofia.herreza@example.com F
3	2003 Andres Vega	12/01/79	956789012	andres.vega@example.com M
4	2004 Valentina Ruiz	05/06/95	947890123	valentina.ruiz@example.com F
5	2005 Felipe Navarro	22/04/93	978901234	felipe.navarro@example.com M

Realizando deletes en la tabla Paciente de MySQL y Replicando

Url del video: <https://youtu.be/twQ3hVw1JFw>

- Mostrando Bitácora previo al delete

The screenshot shows the DB-Clinic interface with the 'paciente' table selected in the schema browser. The audit log (bitacora) shows the following history:

ID_BITACORA	TABLA	OPRACION	ID_REGISTRO	DATOS	ORDEN	REPLICADO	INTENTOS	ULTIMO_ERROR
61	paciente	INSERT	2001	{...}	MySQL	1	1	
62	paciente	INSERT	2002	{...}	MySQL	1	1	
63	paciente	INSERT	2003	{...}	MySQL	1	1	
64	paciente	INSERT	2004	{...}	MySQL	1	1	
65	paciente	INSERT	2005	{...}	MySQL	1	1	
66	paciente	DELETE	2005	{...}	MySQL	1	1	
67	paciente	DELETE	2004	{...}	MySQL	1	1	

The audit log also shows the following actions:

#	Time	Action	Message	Duration / Fetch
1	21:49:41	SELECT * FROM clinica.bitacora LIMIT 0, 1000	5row(s) returned	0.110 sec / 0.000 sec
2	21:53:13	DELETE FROM paciente WHERE id_paciente = 2005	1row(s) affected	0.125 sec
3	21:53:19	SELECT * FROM clinica.paciente LIMIT 0, 1000	4row(s) returned	0.110 sec / 0.000 sec
4	21:55:19	SELECT * FROM clinica.bitacora LIMIT 0, 1000	6row(s) returned	0.172 sec / 0.000 sec
5	21:59:20	DELETE FROM paciente WHERE id_paciente = 2004	1row(s) affected	0.125 sec / 0.000 sec
6	21:59:24	SELECT * FROM clinica.paciente LIMIT 0, 1000	6row(s) returned	0.110 sec / 0.000 sec
7	21:59:32	DELETE FROM paciente WHERE id_paciente = 2004	1row(s) affected	0.172 sec
8	21:59:35	SELECT * FROM clinica.paciente LIMIT 0, 1000	7row(s) returned	0.110 sec / 0.000 sec
9	21:59:40	SELECT * FROM clinica.paciente LIMIT 0, 1000	3row(s) returned	0.094 sec / 0.000 sec
10	22:01:44	SELECT * FROM clinica.paciente LIMIT 0, 1000	3row(s) returned	0.125 sec / 0.000 sec

- Eliminando un paciente (existente) con el id = 2003

The screenshot shows the DB-Clinic interface with the 'paciente' table selected in the schema browser. The audit log (bitacora) shows the following history:

ID_BITACORA	TABLA	OPRACION	ID_REGISTRO	DATOS	ORDEN	REPLICADO	INTENTOS	ULTIMO_ERROR
61	paciente	INSERT	2001	{...}	MySQL	1	1	
62	paciente	INSERT	2002	{...}	MySQL	1	1	
63	paciente	INSERT	2003	{...}	MySQL	1	1	
64	paciente	INSERT	2004	{...}	MySQL	1	1	
65	paciente	INSERT	2005	{...}	MySQL	1	1	
66	paciente	DELETE	2003	{...}	MySQL	1	1	
67	paciente	DELETE	2004	{...}	MySQL	1	1	

The audit log also shows the following actions:

#	Time	Action	Message	Duration / Fetch
1	21:49:41	SELECT * FROM clinica.bitacora LIMIT 0, 1000	5row(s) returned	0.110 sec / 0.000 sec
2	21:53:13	DELETE FROM paciente WHERE id_paciente = 2005	1row(s) affected	0.125 sec
3	21:53:19	SELECT * FROM clinica.paciente LIMIT 0, 1000	4row(s) returned	0.110 sec / 0.000 sec
4	21:55:19	SELECT * FROM clinica.bitacora LIMIT 0, 1000	6row(s) returned	0.172 sec / 0.000 sec
5	21:59:20	DELETE FROM paciente WHERE id_paciente = 2004	1row(s) affected	0.125 sec / 0.000 sec
6	21:59:24	SELECT * FROM clinica.paciente LIMIT 0, 1000	6row(s) returned	0.110 sec / 0.000 sec
7	21:59:32	DELETE FROM paciente WHERE id_paciente = 2004	1row(s) affected	0.172 sec
8	21:59:35	SELECT * FROM clinica.paciente LIMIT 0, 1000	7row(s) returned	0.110 sec / 0.000 sec
9	21:59:40	SELECT * FROM clinica.paciente LIMIT 0, 1000	3row(s) returned	0.094 sec / 0.000 sec
10	22:01:44	SELECT * FROM clinica.paciente LIMIT 0, 1000	3row(s) returned	0.125 sec / 0.000 sec

- Mostrando Bitácora MySQL después de eliminar:

ID_JBTACORA	TABLA	OPERACION	ID_REGISTRO	DATOS	ORIGEN	REPLICADO	INTENTOS	ULTIMO_ERROR
61	paciente	INSERT	2001	{...}	MySQL	1	1	
62	paciente	INSERT	2002	{...}	MySQL	1	1	
63	paciente	INSERT	2003	{...}	MySQL	1	1	
64	paciente	INSERT	2004	{...}	MySQL	1	1	
65	paciente	INSERT	2005	{...}	MySQL	1	1	
66	paciente	INSERT	2006	{...}	MySQL	1	1	
67	paciente	DELETE	2004	{...}	MySQL	1	1	
68	paciente	DELETE	2003	{...}	MySQL	0	0	

Action Output:

#	Time	Action	Message	Duration / Fetch
12	21:53:10	SELECT * FROM clinica.bitacora LIMIT 0, 1000	6 rows(s) returned	0.110 sec / 0.000 sec
13	21:55:10	SELECT * FROM clinica.bitacora LIMIT 0, 1000	6 rows(s) returned	0.172 sec / 0.000 sec
14	21:59:02	SELECT * FROM clinica.paciente LIMIT 0, 1000	4 rows(s) returned	0.125 sec / 0.000 sec
15	21:59:22	SELECT * FROM clinica.paciente WHERE id_paciente = 2004	1 row(s) returned	0.110 sec / 0.000 sec
16	22:00:02	DELETE FROM paciente WHERE id_paciente = 2004	1 row(s) affected	0.172 sec / 0.000 sec
17	21:59:35	SELECT * FROM clinica.paciente LIMIT 0, 1000	3 rows(s) returned	0.119 sec / 0.000 sec
18	21:59:40	SELECT * FROM clinica.paciente LIMIT 0, 1000	3 rows(s) returned	0.094 sec / 0.000 sec
19	22:01:42	SELECT * FROM clinica.paciente LIMIT 0, 1000	3 rows(s) returned	0.125 sec / 0.000 sec
20	22:01:54	SELECT * FROM clinica.paciente LIMIT 0, 1000	7 rows(s) returned	0.105 sec / 0.000 sec
21	22:02:16	DELETE FROM paciente WHERE id_paciente = 2003	1 row(s) affected	0.109 sec / 0.000 sec
22	22:02:20	SELECT * FROM clinica.paciente LIMIT 0, 1000	2 rows(s) returned	0.094 sec / 0.000 sec

- Mostrando la existencia del paciente en oracle previo la replicación:

ID_PACIENTE	NOMBRE	FECHA_NACIMIENTO	TELÉFONO	EMAIL	GÉNERO
1	2001 Javier Torrealba	19/02/92	984567799	javier.torrealba@example.com	M
2	2003 Sofía Herrera	27/03/98	984578901	sofia.herrera@example.com	F
3	2003 Andrés Vega	12/01/79	984579012	andres.vega@example.com	M

- Mostrando Aplicativo Previo Replicación y Refrescando las tablas:

Replicador Bidireccional

Seleccione Tabla

Oracle

paciente

MySQL

Refresh table

ID_PACIENTE	NOMBRE	FECHA_NACI...	TELEFONO	EMAIL	GENERO
2003	Andrés Vega	1979-01-12 0...	956789012	andres.vega@... M	
2002	Sofía Herrera	1988-09-27 0...	945678901	sofia.herrera... F	
2001	Javier Torres	1992-03-19 0...	934567890	javier.torres@... M	

ID_PACIENTE	NOMBRE	FECHA_NACI...	TELEFONO	EMAIL	GENERO
2002	Sofía Herrera	1988-09-27	945678901	sofia.herrera... F	
2001	Javier Torres	1992-03-19	934567890	javier.torres@... M	

REPLICAR

- Aplicativo después de Replicación y Refrescar tablas:

Replicador Bidireccional

Seleccione Tabla

Oracle

paciente

MySQL

Refresh table

ID_PACIENTE	NOMBRE	FECHA_NACI...	TELEFONO	EMAIL	GENERO
2002	Sofía Herrera	1988-09-27 0...	945678901	sofia.herrera... F	
2001	Javier Torres	1992-03-19 0...	934567890	javier.torres@... M	

ID_PACIENTE	NOMBRE	FECHA_NACI...	TELEFONO	EMAIL	GENERO
2002	Sofía Herrera	1988-09-27	945678901	sofia.herrera... F	
2001	Javier Torres	1992-03-19	934567890	javier.torres@... M	

REPLICAR

- Mostrando la Eliminación del paciente en la base Oracle

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Conexiones' sidebar lists various database objects under the 'Clinica' connection, including tables like ATACORA, CITA, CLINICA, HISTORIAL_MEDICO, MEDICAMENTO, RECEITA, and RECEITA_MEDICAMENTO, along with views, packages, procedures, functions, synonyms, and sequences. The main workspace displays the 'PACIENTE' table with the following data:

ID_PACIENTE	NOMBRE	FECHA_NACIMIENTO	TELEFONO	EMAIL	GENERO
1	Javier Torres	19/03/92	934547890	javier.torres@example.com	M
2	Sofia Herrera	27/09/88	945478901	sofia.herrera@example.com	F

A message box at the bottom left says 'Mensajes - Log'. A small red cursor arrow is visible near the top center of the table area.

Realizando Updates en la tabla Paciente de MySQL y Replicando

Url Video: <https://youtu.be/oO3pxBu4nZE>

- Mostrando que registro será actualizado: un paciente con id: 2002

The screenshot shows the MySQL Workbench interface. In the left sidebar, under 'SCHEMAS', there is a tree view of tables, including 'clinica' (with 'bitacora', 'cte', 'clínica', 'doctor', 'historial_médico', 'medicamento', 'paciente', 'receta', 'receta_medicamento', 'tbl_receta_control'), 'Views', 'Stored Procedures', and 'Functions'. Below this, 'Table: paciente' is selected, showing its columns: ID_PACIENTE (int PK), NOMBRE (varchar(100)), FECHA_NACIMIENTO (date), TELÉFONO (varchar(20)), EMAIL (varchar(100)), and GÉNERO (varchar(1)).

In the main area, a query editor window titled 'Query 1' contains the following SQL code:

```

5 VALUES (2002, 'Sofía Herrera', '1988-09-27', '945678901', 'sofia.herrera@example.com', 'F')
6
7 • INSERT INTO paciente (id_paciente, nombre, fecha_nacimiento, telefono, email, genero)
8 VALUES (2003, 'Andrés Vega', '1979-01-12', '956789012', 'andres.vega@example.com', 'M')
9
10 • INSERT INTO paciente (id_paciente, nombre, fecha_nacimiento, telefono, email, genero)
11 VALUES (2004, 'Valentina Ruiz', '1995-06-05', '967890123', 'valentina.ruiz@example.com', 'F')
12
13 • INSERT INTO paciente (id_paciente, nombre, fecha_nacimiento, telefono, email, genero)
14 VALUES (2005, 'Felipe Navarro', '1983-04-22', '978901234', 'felipe.navarro@example.com', 'M')
15 • DELETE FROM paciente
16 WHERE id_paciente = 2005
17 • DELETE FROM paciente
18 WHERE id_paciente = 2003
19
20 • UPDATE paciente
21 SET nombre = 'Sofía Herrera de León'
22 telefono = '987654321'
23 email = 'sofia.helion@example.com'
24 WHERE id_paciente = 2002
25
26

```

The cursor is positioned at line 23, where the update statement is being edited.

Below the query editor, the 'Output' tab shows the execution results:

Query	Action	Message	Duration / Fetch
14 21:59:02	SELECT * FROM clinica.paciente LIMIT 0, 1000	4 rows(s) returned	0.125 sec / 0.000 sec
15 21:59:24	SELECT * FROM clinica.bitacora	6 rows(s) returned	0.110 sec / 0.000 sec
16 21:59:32	DELETE FROM paciente WHERE id_paciente = 2004	1 row(s) affected	0.172 sec
17 21:59:35	SELECT * FROM clinica.paciente LIMIT 0, 1000	7 rows(s) returned	0.110 sec / 0.000 sec
18 21:59:40	SELECT * FROM clinica.paciente LIMIT 0, 1000	3 rows(s) returned	0.094 sec / 0.000 sec
19 22:01:44	SELECT * FROM clinica.paciente LIMIT 0, 1000	3 rows(s) returned	0.125 sec / 0.000 sec
20 22:01:49	SELECT * FROM clinica.paciente LIMIT 0, 1000	7 rows(s) returned	0.109 sec / 0.000 sec
21 22:02:16	DELETE FROM paciente WHERE id_paciente = 2003	1 row(s) affected	0.109 sec
22 22:02:20	SELECT * FROM clinica.paciente LIMIT 0, 1000	2 rows(s) returned	0.094 sec / 0.000 sec
23 22:02:25	SELECT * FROM clinica.paciente LIMIT 0, 1000	2 rows(s) returned	0.140 sec / 0.000 sec
24 22:02:30	SELECT * FROM clinica.bitacora LIMIT 0, 1000	2 rows(s) returned	0.160 sec / 0.000 sec

- Mostrando la bitácora después del Update

The screenshot shows the MySQL Workbench interface. In the left sidebar, under 'SCHEMAS', there is a tree view of tables, including 'clinica' (with 'bitacora', 'cte', 'clínica', 'doctor', 'historial_médico', 'medicamento', 'paciente', 'receta', 'receta_medicamento', 'tbl_receta_control'), 'Views', 'Stored Procedures', and 'Functions'. Below this, 'Table: paciente' is selected, showing its columns: ID_PACIENTE (int PK), NOMBRE (varchar(100)), FECHA_NACIMIENTO (date), TELÉFONO (varchar(20)), EMAIL (varchar(100)), and GÉNERO (varchar(1)).

In the main area, a query editor window titled 'Query 1' contains the following SQL code:

```

1 • SELECT * FROM clinica.bitacora;

```

The results grid shows the following data:

ID_BITACORA	TABLA	OPERACION	ID_REGISTRO	DATOS	OKGRN	REPLICADO	INTENTOS	ULTIMO_ERROR
61	paciente	INSERT	2001	{'email': 'javier.torres@example.com', 'genero': 'M', 'fecha_nacimiento': '1983-09-15', 'telefono': '934567890', 'nombre': 'Javier Torres', 'email': 'javier.torres@example.com'}	MySQL	1	1	0000
62	paciente	INSERT	2002	{'email': 'sofia.herrera@example.com', 'genero': 'F', 'fecha_nacimiento': '1988-09-27', 'telefono': '945678901', 'nombre': 'Sofía Herrera', 'email': 'sofia.herrera@example.com'}	MySQL	1	1	0000
63	paciente	INSERT	2003	{'email': 'andres.vega@example.com', 'genero': 'M', 'fecha_nacimiento': '1979-01-12', 'telefono': '956789012', 'nombre': 'Andrés Vega', 'email': 'andres.vega@example.com'}	MySQL	1	1	0000
64	paciente	INSERT	2004	{'email': 'valentina.ruiz@example.com', 'genero': 'F', 'fecha_nacimiento': '1995-06-05', 'telefono': '967890123', 'nombre': 'Valentina Ruiz', 'email': 'valentina.ruiz@example.com'}	MySQL	1	1	0000
65	paciente	INSERT	2005	{'email': 'felipe.navarro@example.com', 'genero': 'M', 'fecha_nacimiento': '1983-04-22', 'telefono': '978901234', 'nombre': 'Felipe Navarro', 'email': 'felipe.navarro@example.com'}	MySQL	1	1	0000
66	paciente	DELETE	2005	{'email': 'felipe.navarro@example.com', 'genero': 'M', 'fecha_nacimiento': '1983-04-22', 'telefono': '978901234', 'nombre': 'Felipe Navarro', 'email': 'felipe.navarro@example.com'}	MySQL	1	1	0000
67	paciente	DELETE	2004	{'email': 'andres.vega@example.com', 'genero': 'M', 'fecha_nacimiento': '1979-01-12', 'telefono': '956789012', 'nombre': 'Andrés Vega', 'email': 'andres.vega@example.com'}	MySQL	1	1	0000
68	paciente	UPDATE	2002	{'email': 'sofia.herrera@example.com', 'genero': 'F', 'fecha_nacimiento': '1988-09-27', 'telefono': '945678901', 'nombre': 'Sofía Herrera', 'email': 'sofia.herrera@example.com'}	MySQL	0	0	0000
69	bitacora	INSERT	2006	{'accion': 'UPDATE', 'tabla': 'paciente', 'dato': 'Sofía Herrera', 'id_registro': 2002, 'id_bitacora': 69}	MySQL	1	1	0000

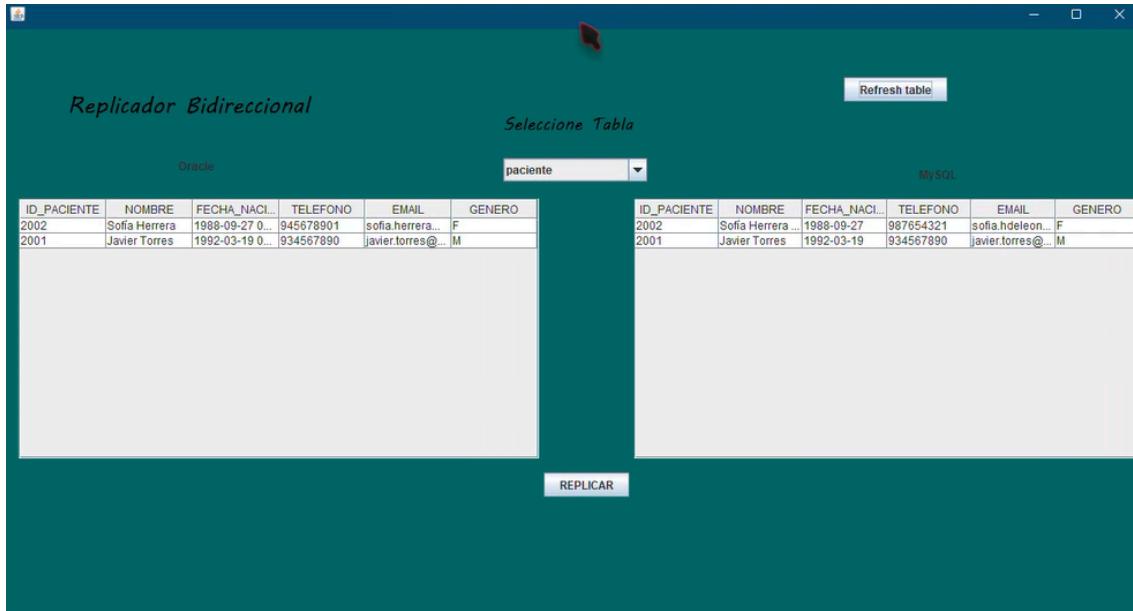
- Mostrando el registro en oracle que aún no ha sido actualizado:

The screenshot shows the Oracle SQL Developer interface. In the left sidebar, under 'Conexiones', there is a tree view of tables, including 'clinica' (with 'bitacora', 'cte', 'clínica', 'doctor', 'historial_médico', 'medicamento', 'paciente', 'receta', 'receta_medicamento', 'tbl_receta_control'), 'Views', 'Indices', 'Procedimientos', 'Operadores', 'Tablas', and 'Tablas de Colas'. Below this, 'Table: paciente' is selected, showing its columns: ID_PACIENTE (int PK), NOMBRE (varchar(100)), FECHA_NACIMIENTO (date), TELÉFONO (varchar(20)), EMAIL (varchar(100)), and GÉNERO (varchar(1)).

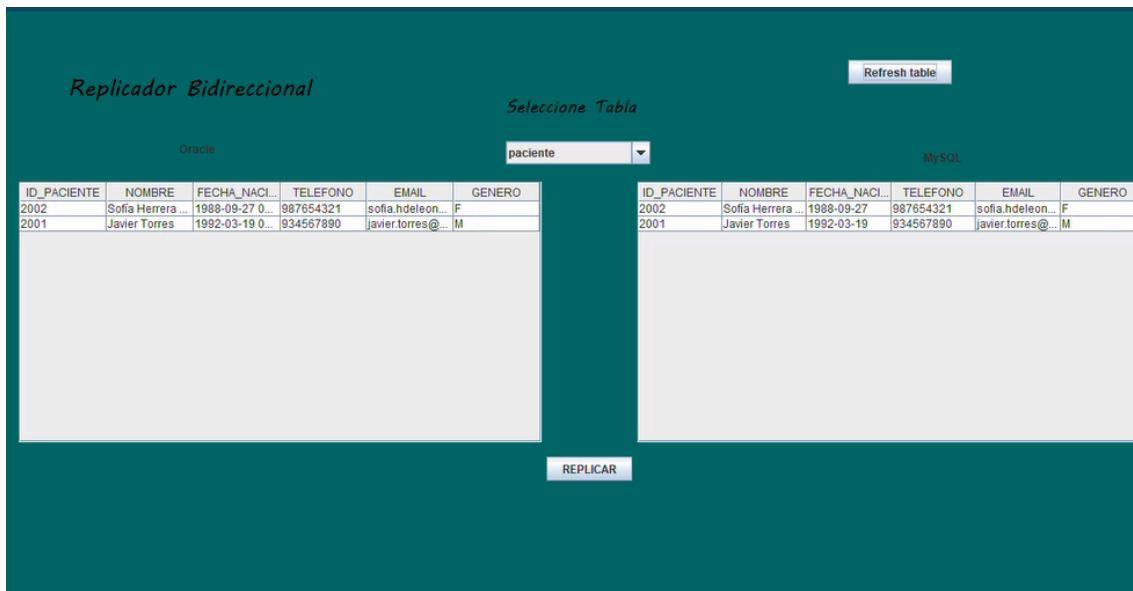
In the main area, a table viewer window titled 'Página de bienvenida' shows the following data:

ID_PACIENTE	NOMBRE	FECHA_NACIMIENTO	TELÉFONO	EMAIL	GÉNERO
2001	Javier Torres	19/03/92	934567890	javier.torres@example.com	M
2002	Sofía Herrera	27/09/88	945678901	sofia.herrera@example.com	F

- Aplicativo previo a la replicación y refrescar



- Aplicativo con el paciente actualizado en la base de oracle después de la replicación y refrescar:



Validación de reglas de negocio e integridad referencial

1. Clinica → Doctor

- Un doctor solo puede existir si está asociado a una clínica existente.
- Validado por:

CONSTRAINT fk_doctor_clinica FOREIGN KEY (id_clinica) REFERENCES clinica(id_clinica)

2. Doctor → Cita

- Una cita médica no puede existir sin un doctor asignado.
- Validado por:

CONSTRAINT fk_cita_doctor FOREIGN KEY (id_doctor) REFERENCES doctor(id_doctor)

3. Paciente → Cita y Historial Médico

- Un paciente puede tener muchas citas y registros de historial médico, pero no pueden existir esos registros sin que exista el paciente.
- Validado por:

CONSTRAINT fk_cita_paciente FOREIGN KEY (id_paciente) REFERENCES paciente(id_paciente)

CONSTRAINT fk_historial_paciente FOREIGN KEY (id_paciente)
REFERENCES paciente(id_paciente)

4. Cita → Receta

- Una receta médica solo tiene sentido si proviene de una cita médica registrada.
- Validado por:

CONSTRAINT fk_receta_cita FOREIGN KEY (id_cita) REFERENCES cita(id_cita)

5. Receta → Receta_Medicamento

Medicamento → Receta_Medicamento

- Una receta puede tener múltiples medicamentos asociados, pero no puede haber una relación en receta_medicamento si no existe la receta ni el medicamento.
- Validado por:

CONSTRAINT fk_rm_receta FOREIGN KEY (id_receta) REFERENCES

receta(id_receta)

CONSTRAINT fk_rm_medicamento FOREIGN KEY (id_medicamento)

REFERENCES medicamento(id_medicamento)