UNIVERSIDAD PERUANA LOS ANDES FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN



ASIGNATURA:

BASE DE DATOS II

DOCENTE:

FERNANDEZ BEJARANO RAUL

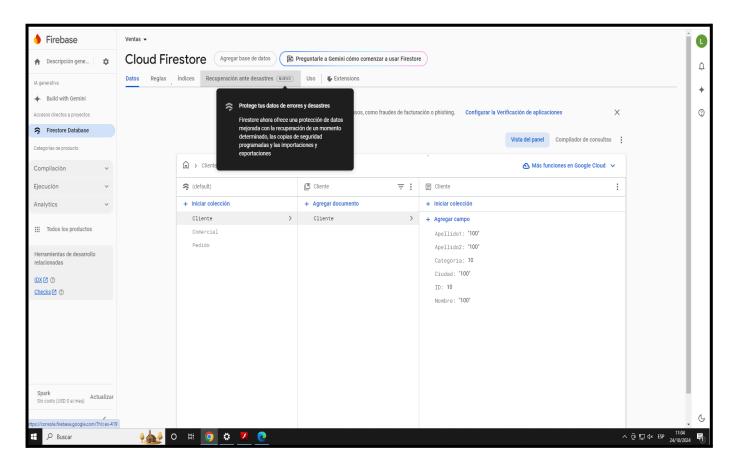
ESTUDIANTE:

GONZALO AGUILERA LUIS

HUANCAYO – PERÚ

2024

1, Crea la Base de Datos ventas



2, Crear tablas

TABLA COMERCIAL:

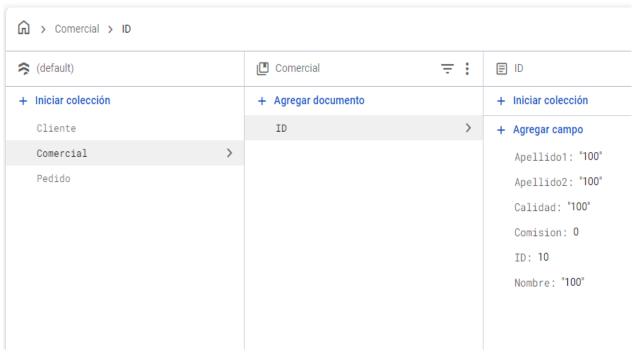


TABLA PEDIDO:

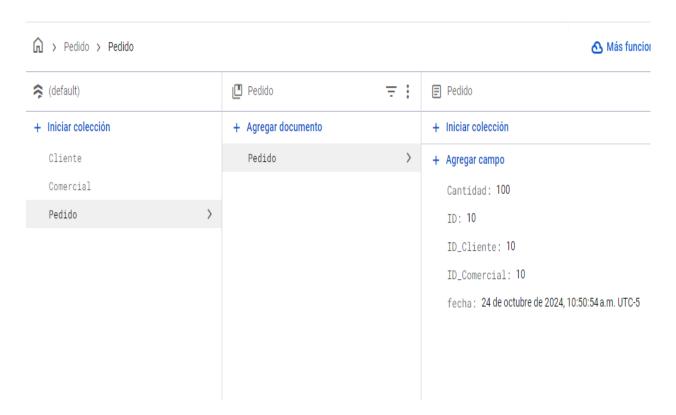
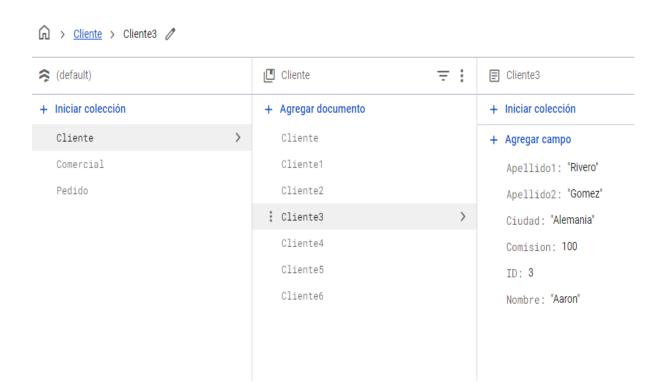


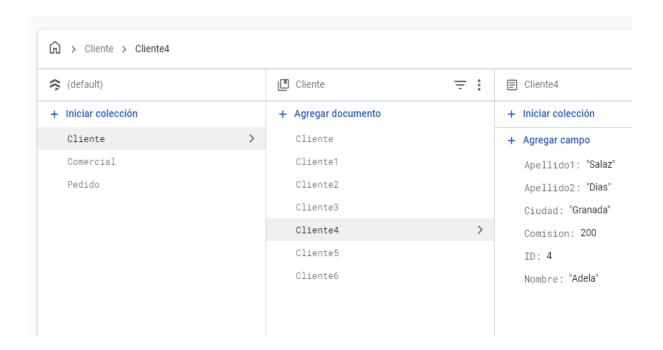
TABLA CLIENTE:

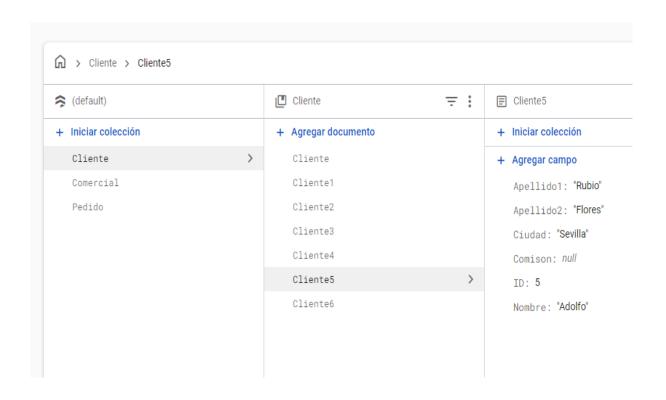
(default)		Cliente	₹ ;	Cliente
Iniciar colección		+ Agregar documento		+ Iniciar colección
Cliente	>	Cliente	>	+ Agregar campo
Comercial				Apellido1: "100"
Pedido				Apellido2: "100"
				Categoria: 10
				Ciudad: "100"
				ID: 10
				Nombre: "100"

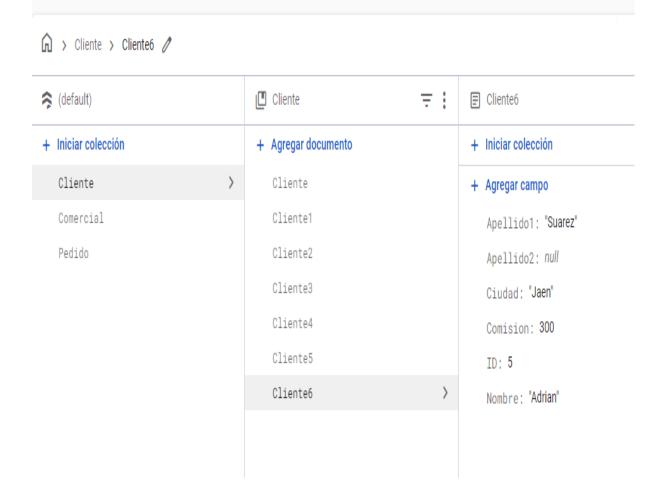
3.Insertar elementos a las tablas creadas:

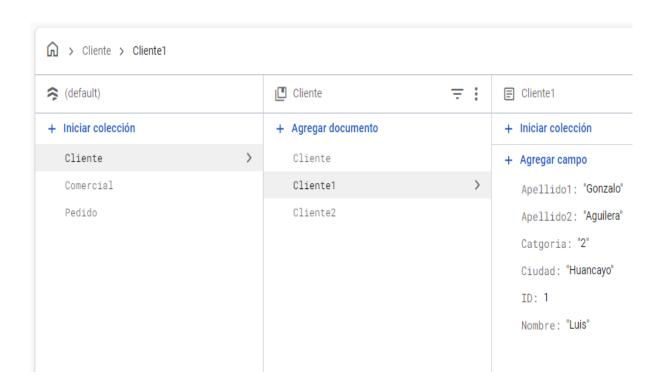
CLIENTE:











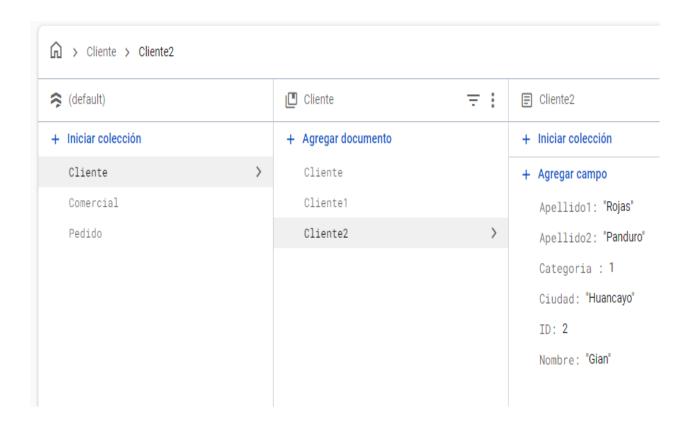
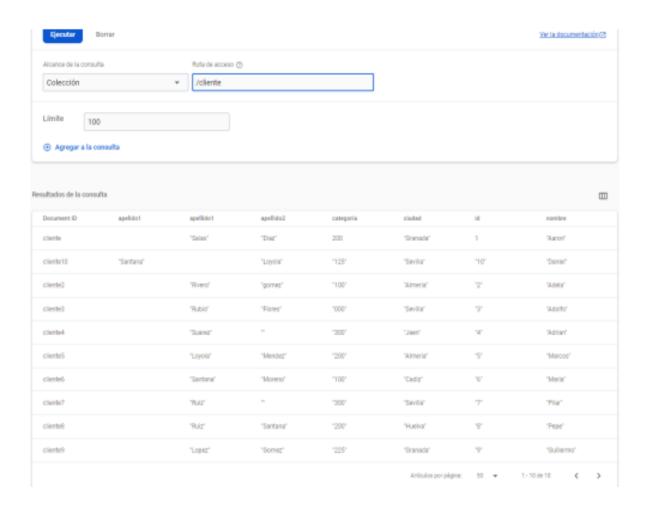


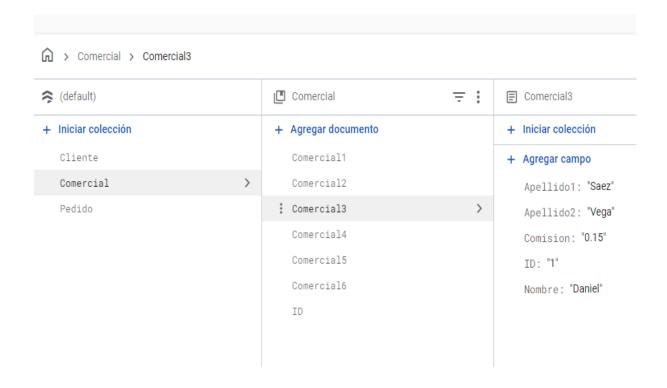
Tabla final CLIENTE con todos los datos:

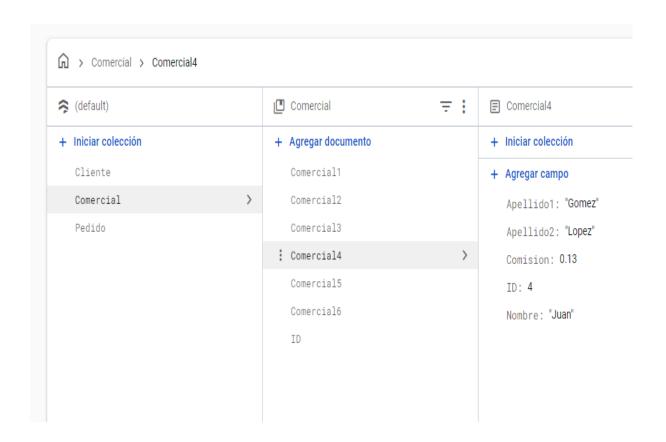


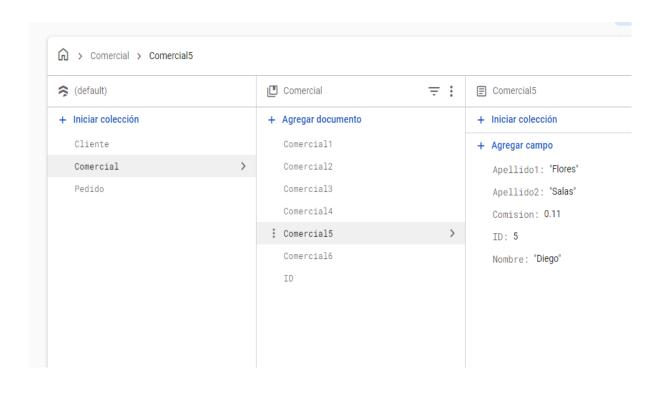
CÓDIGO CLIENTE:

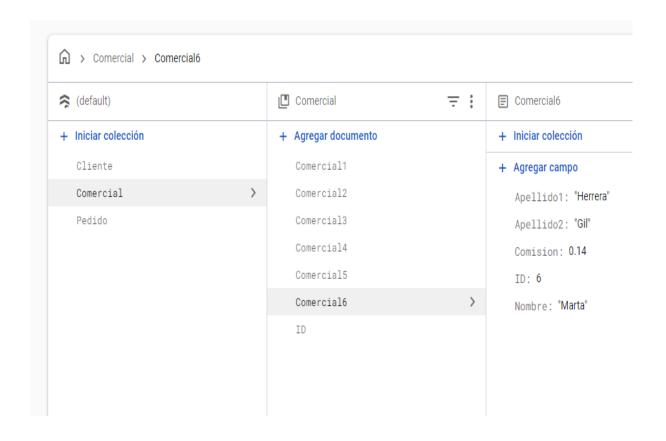
```
INSERT INTO cliente VALUES(1, 'Aarón', 'Rivero', 'Gómez', 'Almería', 100);
INSERT INTO cliente VALUES(2, 'Adela', 'Salas', 'Díaz', 'Granada', 200);
INSERT INTO cliente VALUES(3, 'Adolfo', 'Rubio', 'Flores', 'Sevilla', NULL);
INSERT INTO cliente VALUES(4, 'Adrián', 'Suárez', NULL, 'Jaén', 300);
INSERT INTO cliente VALUES(5, 'Marcos', 'Loyola', 'Méndez', 'Almeria', 200);
INSERT INTO cliente VALUES(6, 'Maria', 'Santana', 'Moreno', 'Cádiz, 100);
INSERT INTO cliente VALUES(7, 'Pilar', 'Ruiz', NULL, 'Sevilla', 300);
INSERT INTO cliente VALUES(8, 'Pepe', 'Ruiz', 'Santana', 'Huelva', 200);
INSERT INTO cliente VALUES(9, 'Guillermo', 'López', 'Gómez', 'Granada', 225);
```

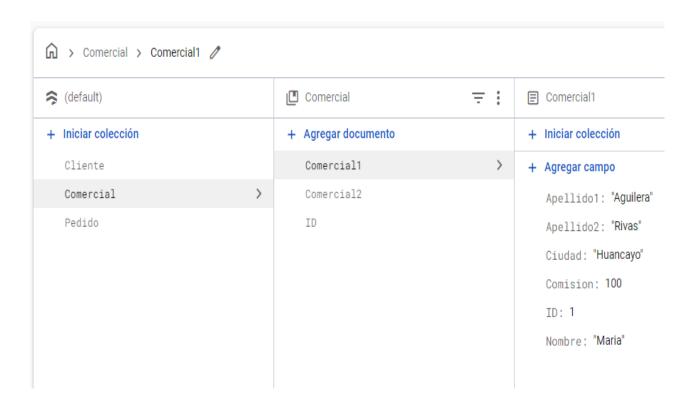
COMERCIAL:











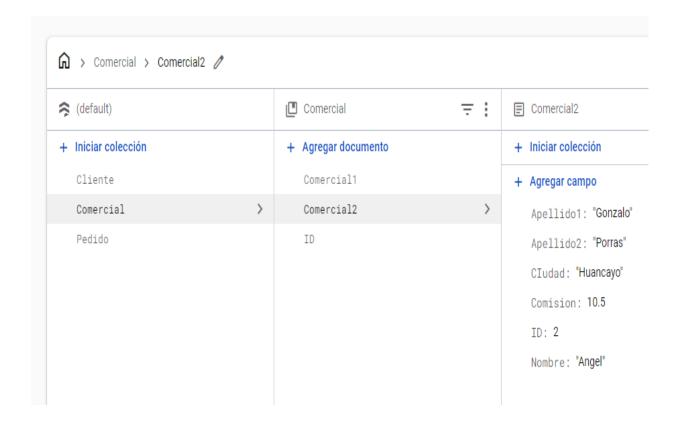
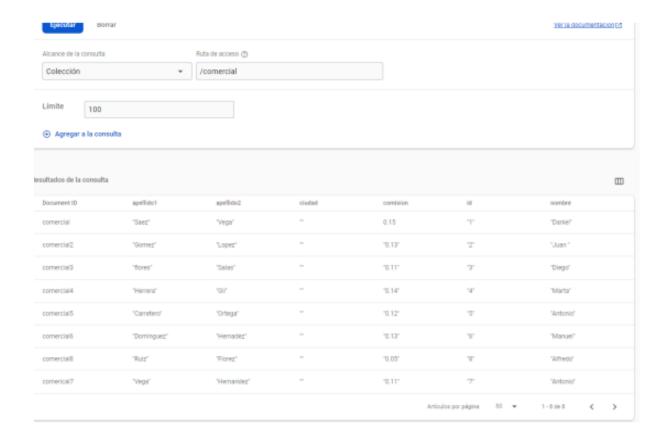


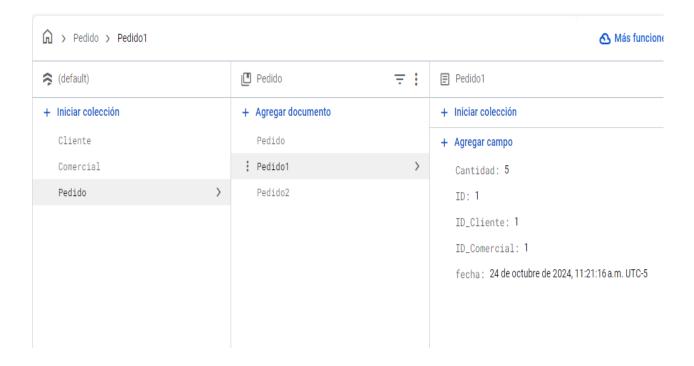
Tabla final de COMERCIAL con todos los datos:

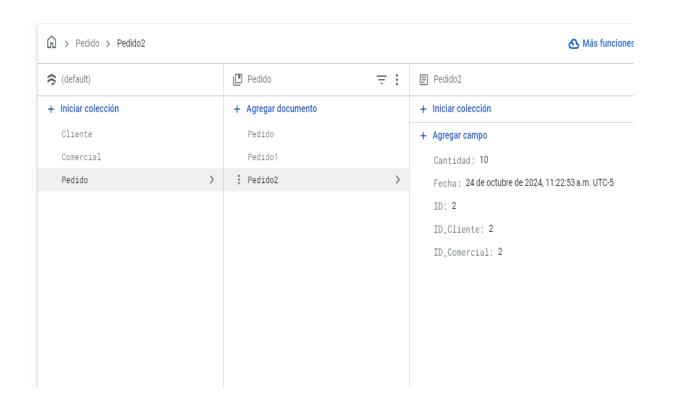


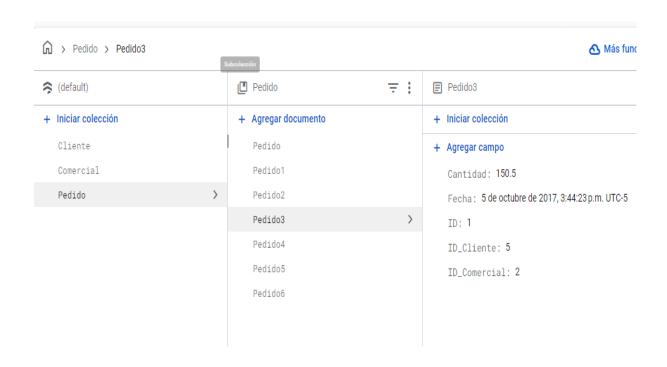
CÓDIGO COMERCIAL:

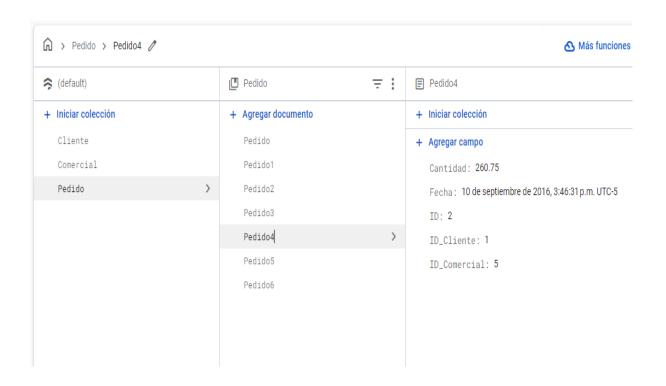
```
INSERT INTO comercial VALUES(1, 'Daniel', 'Sáez', 'Vega', 0.15);
INSERT INTO comercial VALUES(2, 'Juan', 'Gómez', 'López, 0.13);
INSERT INTO comercial VALUES(3, 'Diego', 'Flores', 'Salas', 0.11);
INSERT INTO comercial VALUES(4, 'Marta', 'Herrera', 'Gil', 0.14);
INSERT INTO comercial VALUES(5, 'Antonio', 'Carretero', 'Ortega', 0.12);
INSERT INTO comercial VALUES(6, 'Manuel', 'Domínguez', 'Hemández', 0.13);
INSERT INTO comercial VALUES(7, 'Antonio', 'Vega', 'Hernández', 0.11);
INSERT INTO comercial VALUES(8, 'Alfredo', 'Ruiz', 'Flores', 0.05);
```

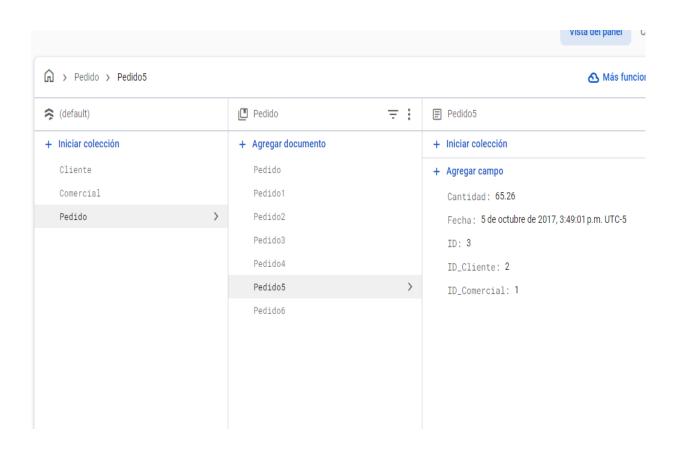
PEDIDO:











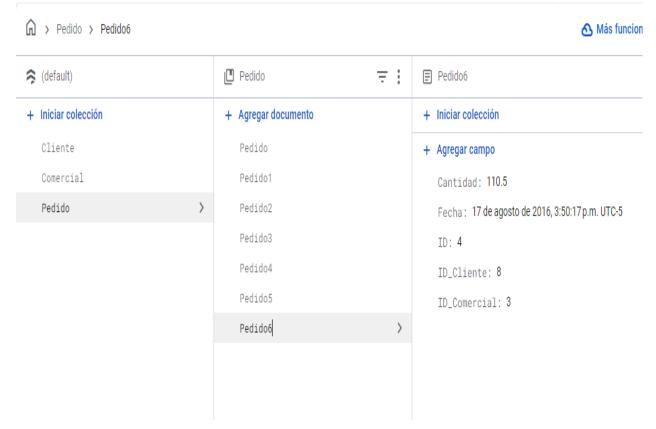


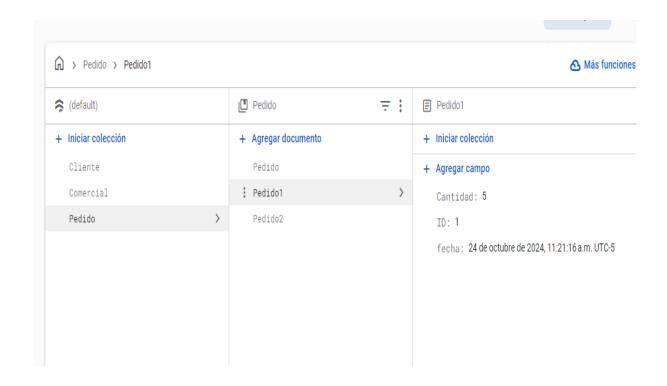
Tabla final PEDIDO con todos los datos:

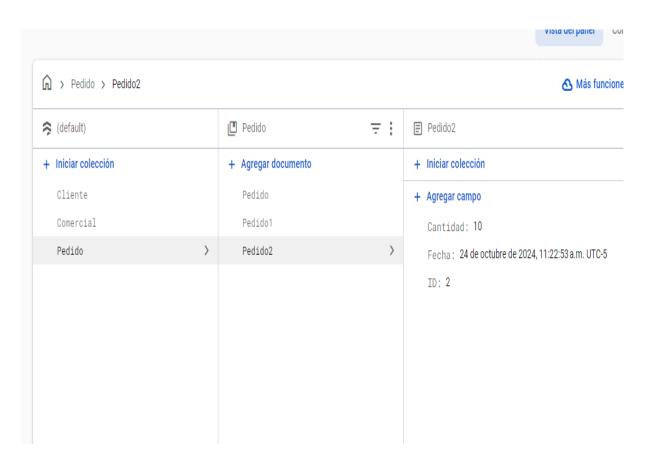
Resultados de la consul	ta					
Document ID	contided	fecha	id	H.	id_cliente	id_comercial
pedido1	"150.5"	5 de octubre de 2017, 12:00:00 a.m. UTC-5	77		27	2.
pedido10	"250.45"	27 de junio de 2015, 12:00:00 a.m. UTC-5	70"		T	2"
pedido11	'75.29'	17 de agosto de 2016, 12:00:00a.m. UTC-5	711"		.3.	7
pedido12	'3045.6'	25 de abril de 2017, 12:00:00 a.m. UTC-5	12		7	71"
pedido13	"545.75"	25 de enero de 2019, 12:00:00 a.m. UTC 5	13"		.6.	7*
pedido14	"145.82"	2 de febrero de 2017, 12:00:00 a.m. UTC-5	74"		ж.	7"
pedido15	-		-	-		
pedido16	"1951"	2 de febrero de 2017, 12:00:00 a.m. UTC-5	16"		71"	'8'
pedido2	"270.65"	10 de septiembre de 2016, 12:00:00 a.m. UTO 5	"2"		71"	T
pedido3	'65.26'	5 de octubre de 2017, 12:00:00 a.m. UTC-5	-3-		2"	7.
pedido4	"110.5"	5 de agosto de 2016, 12:00:00 a.m. UTC-5	4"		ж.	3"
pedido5	'940.5'	10 de septiembre de 2017, 12:00:00 a.m. UTG-5	'5'		'5'	T
pedido6	"2400.6"	27 de julio de 2016, 12:00:00 a.m. UTC-5	16"		ア	7.
pedido7	"5760"	10 de septiembre de 2015, 12:00:00 a.m. UTO-5	7		2"	7"
pedido8	"1983.43"	10 de octubre de 2017, 12:00:00 a.m. UTC-5	-8-		-8-	3"
pedido9	"2490.4"	10 de octubre de 2016, 12:00:00 a.m. UTO-5	191		ъ.	3.
				Articulos por pégine	50 w	1-16 de 16

CÓDIGO PEDIDO:

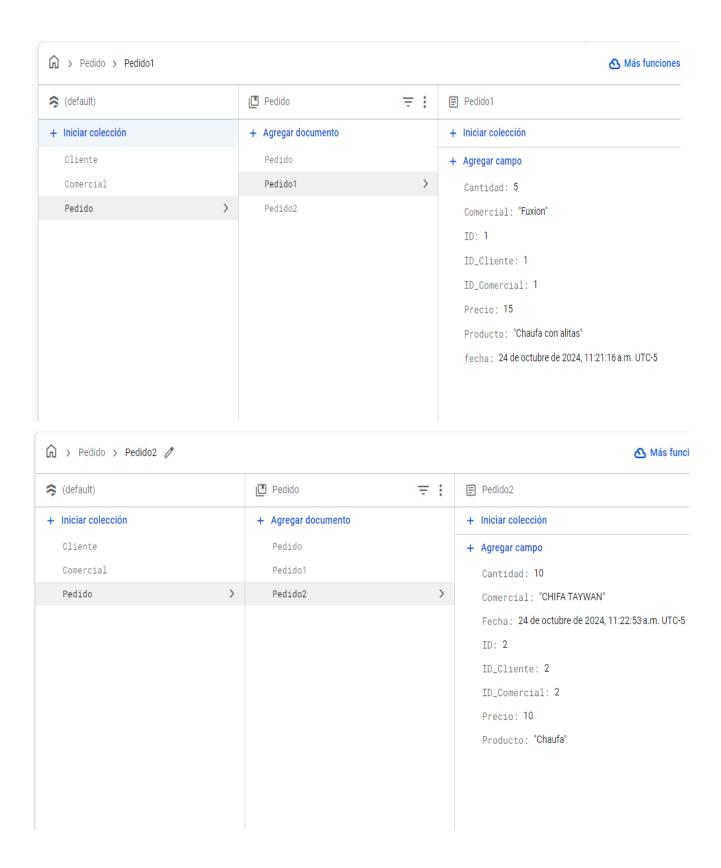
```
INSERT INTO pedido VALUES (1, 150.5, "2017-10-05", 5, 2);
INSERT INTO pedido VALUES(2, 270.65, '2016-09-10', 1, 5);
INSERT INTO pedido VALUES(3, 65.26, '2017-10-05, 2, 1);
INSERT INTO pedido VALUES(4, 110.5, '2016-08-17", 8, 3);
INSERT INTO pedido VALUES(5, 948.5, '2017-09-10, 5, 2);
INSERT INTO pedido VALUES(6, 2400.6, '2016-07-27', 7, 1);
INSERT INTO pedido VALUES(7, 5760, '2015-09-10', 2, 1);
INSERT INTO pedido VALUES(8, 1983.43, '2017-10-10', 4, 6);
INSERT INTO pedido VALUES(9, 2480.4, '2016-10-10', 8, 3);
INSERT INTO pedido VALUES(10, 250.45, '2015-06-27', 8, 2);
INSERT INTO pedido VALUES(11, 75.29, '2016-08-17, 3, 7);
INSERT INTO pedido VALUES(12, 3045.6, '2017-04-25', 2, 1);
INSERT INTO pedido VALUES(13, 545.75, '2019-01-25', 6, 1);
INSERT INTO pedido VALUES(14, 145.82, '2017-02-02', 6, 1);
INSERT INTO pedido VALUES (15, 370.85, '2019-03-11', 1, 5);
INSERT INTO pedido VALUES (16, 2389.23, '2019-03-11', 1, 5)
```

4 Eliminar 2 filas de la tabla pedido :





5.Modificar 2 tablas de la tabla pedido:



6, Consultas Básicas

1. Consulta a) Listado de pedidos ordenados por fecha de realización:

```
// Firestore: Listado de pedidos ordenados por fecha de
realización (más recientes primero)
db.collection("pedidos")
  .orderBy("fechaRealizacion", "desc") // Ordena por fecha de
forma descendente
get()
then((snapshot) => {
    snapshot.
forEach((doc) => {
console.log(doc.data()); // Imprime cada pedido
   });
  })
   });
  })
   });
catch((error) => {
   console.error("Error obteniendo los pedidos: ", error);
  });
```

2. Consulta b) Los dos pedidos de mayor valor:

```
// Firestore: Obtener los dos pedidos de mayor valor
db.collection("pedidos")
orderBy("valor", "desc") // Ordena por valor descendente
limit(2) // Limita a los 2 pedidos más caros
  .get()
then((snapshot) => {
    snapshot.
    snapshot
forEach((doc) => {
     console.log(doc.data()); // Imprime cada pedido
   });
  })
  });
  })
   });
catch((error) => {
console.error("Error obteniendo los pedidos: ", error);
 });
```

3. Consulta c) Clientes con pedidos (sin repetir IDs):

```
// Firestore: Clientes con pedidos, sin repetir
identificadores
db.collection("pedidos")
get()
  .then((snapshot) => {
    let clientes = new Set(); // Set para evitar duplicados
    snapshot.
    snaps
    sn
forEach((doc) => {
      clientes.
      clie
add(doc.data().cliente); // Agrega el cliente ID al set
    });
    });
    });
console.log([...clientes]); // Convierte Set en array y lo
imprime
 })
  })
catch((error) => {
```

```
console.error("Error obteniendo los clientes: ", error);
});
```

4. Consulta de) Clientes con segundo apellido no nulo, ordenados alfabéticamente:

```
// Firestore: Clientes cuyo segundo apellido no es NULL,
ordenados por apellido y nombre
db.
collection("clientes")
  .where("segundo Apellido", "!=", null) // Filtra por segundo
apellido no nulo
  .orderBy("apellido") // Ordena por apellido
  .orderBy("nombre") // Ordena por nombre
  .get()
  .then((snapshot) => {
    snapshot.forEach((doc) => {
      cons
console.log(doc.data()); // Imprime cada cliente
    });
  })
   });
  })
catch((error) => {
    cons
console.error("Error obteniendo los clientes: ", error);
```

5. Consulta de) Clientes cuyos nombres empiezan por "A" o "P" y terminan en "y":

```
// Firestore: Clientes cuyos nombres empiezan por "A" o "P" y
terminan en "y"
let clientes = [];
// Clientes que comienzan con "A"
db.
db
collection("clientes")
orderBy("nombre")
  .startAt("A")
endAt("A\uf8ff") // Buscar nombres que empiezan con "A"
  .get()
  .then((snapshot) => {
   snapshot.
   snapshot.
    sn
forEach((doc) => {
      clientes.
      clientes.
      clien
```

```
push(doc.data());
    });
    });
    })
// Clientes que comienzan con "P"
    retu
return db.collection("clientes")
orderBy("nombre")
startAt("P")
endAt("P\uf8ff") // Buscar nombres que empiezan con "P"
      .get();
  })
  })
then((snapshot) => {
    snapshot.
```

```
forEach((doc) => {
      clientes.
      clientes
push(doc.data());
    });
    });
    });
// Filtrar los clientes cuyo nombre termine en "y"
    let filtrados = clientes.filter((cliente) =>
cliente.nombre.endsWith("y"));
    consol
    con
console.log(filtrados);
 })
  .catch((error) => {
   console.error("Error obteniendo los clientes: ", error);
  });
```

6. Consulta f) Clientes cuyos nombres no empiezan por "A":

```
// Firestore: Clientes cuyos nombres NO empiezan por "A"
```

```
db.collection("clientes")
.
.
get()
.then((snapshot) => {
    let clientes = snapshot.docs.filter((doc) =>
!doc.data().nombre.startsWith("A"));
    c
console.log(clientes.map((doc) => doc.data())); // Imprime clientes filtrados
    })
.catch((error) => {
    consol

console.error("Error obteniendo los clientes: ", error);
    });
```

7. Consulta g) Comerciales cuyos nombres terminan en "o":

```
// Firestore: Comerciales cuyos nombres terminan en "o"
db.collection("comerciales")
.

get()
   .then((snapshot) => {

let comerciales = snapshot.docs.filter((doc) => doc.data().nombre.endsWith("o"));
   con
```

```
console.log(comerciales.map((doc) => doc.data())); // Imprime
comerciales filtrados
    })
    .
    })
    catch((error) => {
        c
        console.error("Error obteniendo los comerciales: ", error);
    });
```