

Datos

//Creación de personas

```
CREATE ( Y: Persona { apellido: "Lopez", nombre: "Yoel" } );
CREATE ( A: Persona { apellido: "Bassi", nombre: "Agustín" } );
CREATE ( D: Persona { apellido: "Lara", nombre: "Mauricio" } );
CREATE ( M: Persona { apellido: "Pacher", nombre: "Federico" } );
CREATE ( J: Persona { apellido: "Sarche", nombre: "Felipe" } );
CREATE ( C: Persona { apellido: "Ducca", nombre: "Brian" } );
```

//Creación de materias

```
CREATE ( G: Materia { nombre: "Gestión de grandes volúmenes de
datos", short_name: "GdV", obligatoria: true } );
CREATE ( I: Materia { nombre: "Infraestructura para la
implementación de sistemas", short_name: "IIS", obligatoria:
false } );
CREATE ( P: Materia { nombre: "Arquitectura de protocolos",
short_name: "AdP", obligatoria: true } );
CREATE ( D: Materia { nombre: "Desarrollo de aplicaciones web",
short_name: "DdA", obligatoria: true } );
CREATE ( Ci: Materia { nombre: "Ciberseguridad en internet de las
cosas", short_name: "CibS", obligatoria: true } );
CREATE ( T: Materia { nombre: "Testing de Sistemas de Internet de
las Cosas", short_name: "TSIoT", obligatoria: false } );
```

//Creación de relaciones DICTO (docentes)

```
MATCH (A: Persona { apellido: "Bassi"}), (P: Materia { short_name:
"AdP"}) CREATE (A)-[:DICTO {cohorte: 3, bimestre: 2 } ]->(P);
MATCH (Y: Persona { apellido: "Lopez"}), (G: Materia { short_name:
"GdV"}) CREATE (Y)-[:DICTO {cohorte: 3, bimestre: 3 } ]->(G);
MATCH (Y: Persona { apellido: "Lopez"}), (I: Materia { short_name:
"IIS"}) CREATE (Y)-[:DICTO {cohorte: 3, bimestre: 4 } ]->(I);
MATCH (C: Persona { apellido: "Ducca"}), (Ci: Materia
{ short_name: "CibS"}) CREATE (C)-[:DICTO {cohorte: 3, bimestre:
3 } ]->(Ci);
```

//Creación de relaciones CURSO (alumnos)

```
MATCH (B: Persona { apellido: "Lara"}), (P: Materia { short_name:
"AdP"}) CREATE (B)-[:CURSO {calificacion: 8, cohorte: 3, bimestre:
2 } ]->(P);
MATCH (B: Persona { apellido: "Lara"}), (G: Materia { short_name:
"GdV"}) CREATE (B)-[:CURSO {calificacion: 9, cohorte: 3, bimestre:
3, grupo: 1 } ]->(G);
MATCH (M: Persona { apellido: "Pacher"}), (G: Materia
{ short_name: "GdV"}) CREATE (M)-[:CURSO {calificacion: 9, cohorte:
3, bimestre: 3, grupo: 2 } ]->(G);
MATCH (J: Persona { apellido: "Sarche"}), (G: Materia
{ short_name: "GdV"}) CREATE (J)-[:CURSO {calificacion: 9, cohorte:
3, bimestre: 3, grupo: 1 } ]->(G);
MATCH (J: Persona { apellido: "Sarche"}), (T: Materia
```

```

{ short_name: "TSIoT" } ) CREATE (J)-[:CURSO { cohorte: 3, bimestre:
4 } ]->(T);
MATCH (B: Persona { apellido: "Lara"}), (I: Materia { short_name:
"IIS" } ) CREATE (B)-[:CURSO { cohorte: 3, bimestre: 4 } ]->(I);
MATCH (M: Persona { apellido: "Pacher"}), (I: Materia
{ short_name: "IIS" } ) CREATE (M)-[:CURSO { cohorte: 3, bimestre:
4 } ]->(I);
MATCH (C: Persona { apellido: "Ducca"}), (D: Materia
{ short_name: "DdA" } ) CREATE (C)-[:CURSO { cohorte: 3, bimestre:
1 } ]->(D);
//Creación de relaciones CONOCE
MATCH (B: Persona { apellido: "Lara"}), (M: Persona { apellido:
"Castello" } ) CREATE (B)-[:CONOCE]->(M);
MATCH (M: Persona { apellido: "Pacher"}), (J: Persona { apellido:
"Castiñeiras" } ) CREATE (M)-[:CONOCE]->(J);
MATCH (J: Persona { apellido: "Sarche"}), (A: Persona
{ apellido: "Bassi" } ) CREATE (J)-[:CONOCE]->(A);

```

Query

```
//0. Listado de alumnos (personas que cursan materias)
MATCH (p:Persona)-[:CURSO]->(m:Materia)
RETURN p;

//1. Listado de alumnos que cursaron la misma materia, pero en
grupos distintos
MATCH (a:Persona)-[ac:CURSO]->()-[bc:CURSO]-(b:Persona)
WHERE ac.grupo <> bc.grupo
RETURN DISTINCT a.nombre, a.apellido, b.nombre, b.apellido;

//2. Listado de personas que dictaron más de una materia
MATCH (p:Persona)-[d:DICTO]->()
WITH p, count(d) AS cnt
WHERE cnt > 1
RETURN p;

//3. Calculo de mi promedio
MATCH (p:Persona)-[c:CURSO]->()
WHERE p.apellido = "Lara"
RETURN avg(c.calificacion);

//4. Recomendación de personas que cursaron juntas pero no se
conocen
MATCH (a:Persona)-[ac:CURSO]->()-[bc:CURSO]-(b:Persona)
WHERE ac.cohorte = bc.cohorte
AND ac.bimestre = bc.bimestre
AND NOT (a)-[:CONOCE]-(b)
RETURN DISTINCT a.apellido, b.apellido;

//5.a. Conocidos de mis conocidos hasta longitud 2
MATCH (a:Persona)-[s:CONOCE*..2]->(b)
WHERE a.apellido = "Lara"
RETURN DISTINCT a,b;

//5.b. Conocidos de mis conocidos hasta longitud indefinida
MATCH (a:Persona)-[s:CONOCE*..]->(b)
WHERE a.apellido = "Lara"
RETURN DISTINCT a,b;

//6. Alumnos que también son docentes (dictaron y cursaron materias)
MATCH (a:Persona)
WHERE EXISTS ((a:Persona)-[:DICTO]-())
AND EXISTS ((a:Persona)-[:CURSO]-())
RETURN a;

//7. Materias optativas que no he cursado y que curso alguien con
quien yo he cursado
MATCH (a:Persona)-[ac:CURSO]-(am:Materia), (b:Persona)-[bc:CURSO]-
(bm:Materia)
WHERE a.apellido = "Lara"
```

```
AND NOT EXISTS ((a:Persona)-[:CURSO]-(bm:Materia))
AND ac.cohorte = bc.cohorte
AND ac.bimestre = bc.bimestre
AND bm.obligatoria = false
RETURN bm.nombre;

//8. Materias optativas que no he cursado y que curso alguien con
quien yo he cursado
//y es conocido directo o indirecto
MATCH (a:Persona)-[ac:CURSO]-(am:Materia), (b:Persona)-[bc:CURSO]-(bm:Materia)
WHERE a.apellido = "Lara"
AND NOT EXISTS ((a:Persona)-[:CURSO]-(bm:Materia))
AND ac.cohorte = bc.cohorte
AND ac.bimestre = bc.bimestre
AND bm.obligatoria = false
AND EXISTS ((a:Persona)-[:CONOCE*..]-(b:Persona))
RETURN bm.nombre;
```