



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
IIC2413 - BASES DE DATOS

Examen

Fecha: 12 de diciembre de 2024

2º semestre 2024 - Profesores: Eduardo Bustos - Christian Álvarez

INSTRUCCIONES

- a) La prueba tiene 4 preguntas, cada una vale 6 puntos ($6+1=7$) y una pregunta Bonus de 5 décimas sobre la nota final de la prueba. **Duración 2,5 horas.**
- b) **Lea toda la prueba primero**, las preguntas no están en orden de dificultad
- c) Al finalizar la prueba debe digitalizar cada pregunta en formato PDF VERTICAL y subirlas a Canvas **correctamente**.
- d) Cada plana debe contener su Nombre Completo y Número de Lista en la parte superior.
- e) No mezcle respuestas de diferentes preguntas en la misma plana. Sí puede responder una pregunta en una plana y otra en la otra plana de la misma hoja.
- f) Debe firmar la lista de asistencia.
- g) No seguir esta instrucción dificulta el trabajo de evaluación y por ende el tiempo de entrega de las notas, además los expone a un descuento de hasta 5 décimas de la nota final.

Pregunta 1 - Modelo ER + Diseño de BD + FN

La FIA desea implementar un sistema de gestión para llevar un control eficiente de los equipos, pilotos, carreras y resultados. El sistema debe permitir registrar información sobre los equipos participantes, los pilotos, las carreras programadas y los resultados de cada carrera. A continuación, se detallan los requisitos:

Equipos

- Cada equipo tiene un identificador único, nombre, país de origen y año de fundación.
- Un equipo puede tener varios pilotos.

Pilotos

- Cada piloto tiene un identificador único, nombre, fecha de nacimiento y nacionalidad.
- Un piloto está asociado a un equipo.

Carreras

- Cada carrera tiene un identificador único, nombre del Gran Premio, fecha, país y número de vueltas.
- Una carrera puede tener múltiples resultados, uno por cada piloto participante.

Resultados

- Cada resultado tiene un identificador único, posición final, tiempo total y puntos obtenidos.
- Un resultado está asociado a una carrera y a un piloto.

De acuerdo al enunciado anterior, responda las siguientes preguntas.

- a) Crea un diagrama entidad-relación que represente el sistema de gestión de la Fórmula 1. Especifique la cardinalidad de las relaciones y las llaves primarias (3 pts)

Respuesta en Figura 1 o Figura 2

- b) Genere el esquema de la BD, especificando los campos de las tablas, tipos de datos, llaves primarias y llaves foráneas (2 pts)

Solución:

```
Equipo(id_equipo: INT PRIMARY KEY,  
       nombre_equipo: STRING,  
       pais_origen: STRING,
```

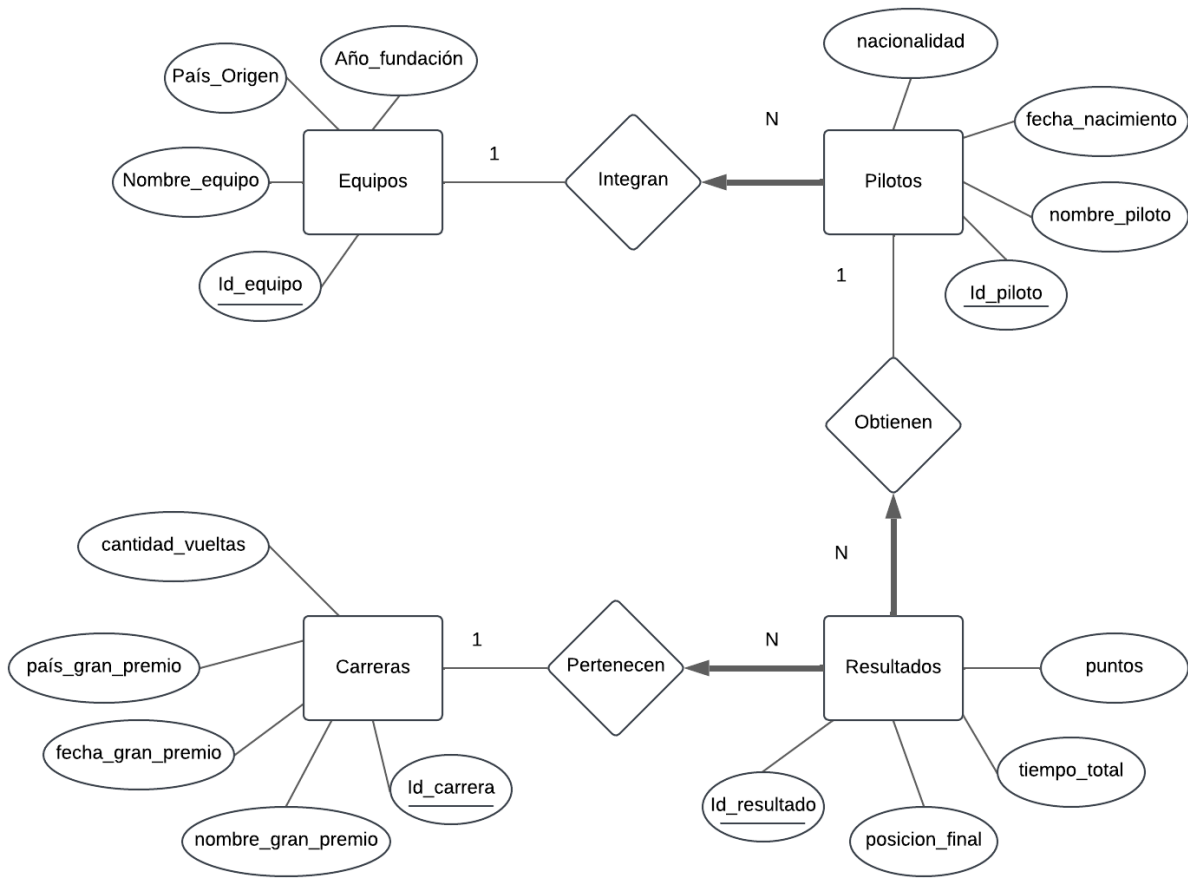


Figura 1: Digrama E/R F1

año_fundacion: DATE)

```

Piloto(id_piloto: INT PRIMARY KEY,
      nombre_piloto: STRING,
      fecha_nacimiento: DATE,
      nacionalidad: STRING,
      id_equipo INT,
      FOREIGN KEY (id_equipo) REFERENCES equipo(id_equipo))
  
```

```

Carrera(id_carrera: INT PRIMARY KEY,
        nombre_gran_premio: STRING,
        fecha_gran_premio: DATE,
        pais_gran_premio: STRING,
        cantidad_de_vueltas: INT)
  
```

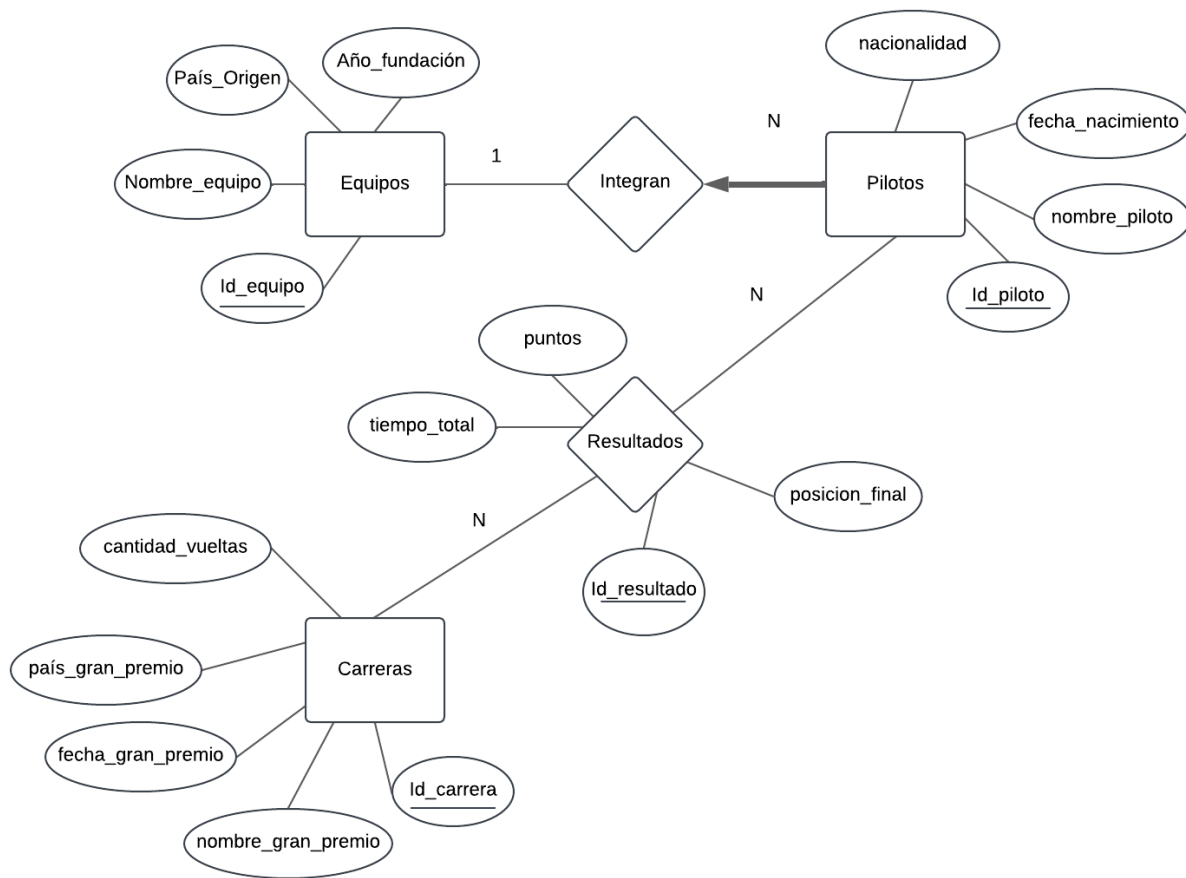


Figura 2: Digrama E/R F1-v2

```
Resultados(id_resultado: INT PRIMARY KEY,
posicion_final: INT,
tiempo_total: TIME (puede ser INT),
puntos: INT,
id_carrera: INT,
id_piloto: INT,
FOREIGN KEY (id_carrera) REFERENCES carrera(id_carrera),
FOREIGN KEY (id_piloto) REFERENCES piloto(id_piloto))
```

- c) Especifique si su modelo se encuentra en 3NF o BCNF. Justifique su respuesta y en caso de estar en 3NF, explique por qué no puede generarse un modelo en BCNF (1 pts).

Solución: El modelo anterior está en 3NF porque en todas las tablas, todos los campos dependen directamente desde la llave y no hay

dependencias transitivas.

El modelo no se encuentra en BCNF porque hay dependencias funcionales de elementos que no son parte de la llave. Por ejemplo en la tabla equipos, a partir del nombre del equipo se puede determinar su año de fundación o país de origen. Lo mismo pasa con la tabla de pilotos, donde a partir del nombre del piloto, se pueden determinar los otros atributos. Similarmente también en la tabla de carreras, ya que a partir del nombre del gran premio, se pueden determinar los otros atributos.

Pregunta 2 - SQL + Lógica en la BD

A continuación se describe un esquema de BD para gestionar campeonatos de gimnasia rítmica, considerando la participación individual y en conjunto, así como los instrumentos utilizados en las competiciones individuales.

Con respecto a las relaciones entre las tablas, se especifica lo siguiente:

- Una **competencia** puede tener múltiples **participaciones**.
- Un **gimnasta** puede participar en múltiples **participaciones**.
- Un **conjunto** puede participar en múltiples **participaciones**.
- Un **equipo** puede tener múltiples **conjuntos**.
- Un **conjunto** puede tener múltiples **gimnastas**.
- Una **participación** puede ser evaluada por múltiples **jueces**.
- Un **juez** puede evaluar múltiples **participaciones**.

Tabla: Competencia

- id_competencia (INT, PRIMARY KEY, AUTO_INCREMENT)
- nombre (VARCHAR(100), NOT NULL)
- fecha (DATE, NOT NULL)
- lugar (VARCHAR(100), NOT NULL)

Tabla: Gimnasta

- id_gimnasta (INT, PRIMARY KEY, AUTO_INCREMENT)
- nombre (VARCHAR(100), NOT NULL)
- fecha_nacimiento (DATE, NOT NULL)
- pais (VARCHAR(50), NOT NULL)
- id_equipo (INT, FOREIGN KEY REFERENCES Equipo(id_equipo))

Tabla: Equipo

- id_equipo (INT, PRIMARY KEY, AUTO_INCREMENT)
- nombre (VARCHAR(100), NOT NULL)
- pais (VARCHAR(50), NOT NULL)

Tabla: Conjunto

- id_conjunto (INT, PRIMARY KEY, AUTO_INCREMENT)
- nombre (VARCHAR(100), NOT NULL)

- id_equipo (INT, FOREIGN KEY REFERENCES Equipo(id_equipo))

Tabla: Gimnasta_Conjunto

- id_gimnasta (INT, FOREIGN KEY REFERENCES Gimnasta(id_gimnasta))
- id_conjunto (INT, FOREIGN KEY REFERENCES Conjunto(id_conjunto))
- PRIMARY KEY (id_gimnasta, id_conjunto)

Tabla: Participacion

- id_participacion (INT, PRIMARY KEY, AUTO_INCREMENT)
- id_competencia (INT, NOT NULL, FOREIGN KEY REFERENCES Competencia(id_competencia))
- id_gimnasta (INT, FOREIGN KEY REFERENCES Gimnasta(id_gimnasta), NULL)
- id_conjunto (INT, FOREIGN KEY REFERENCES Conjunto(id_conjunto), NULL)
- tipo_participacion (ENUM('individual', 'conjunto'), NOT NULL)
- instrumento (ENUM('pelota', 'clavas', 'cinta', 'manos libres'), NULL)
- puntaje (DECIMAL(5, 2), NOT NULL)

Tabla: Juez

- id_juez (INT, PRIMARY KEY, AUTO_INCREMENT)
- nombre (VARCHAR(100), NOT NULL)
- pais (VARCHAR(50), NOT NULL)

Tabla: Evaluacion

- id_evaluacion (INT, PRIMARY KEY, AUTO_INCREMENT)
- id_participacion (INT, NOT NULL, FOREIGN KEY REFERENCES Participacion(id_participacion))
- id_juez (INT, NOT NULL, FOREIGN KEY REFERENCES Juez(id_juez))
- puntaje (DECIMAL(5, 2), NOT NULL)
- comentarios (TEXT)

Dada la información anterior, responder las siguientes preguntas:

- a) Genere una consulta SQL que obtenga el puntaje de cada participación de los integrantes de un equipo (conjunto e individual) en la competencia con *id_competencia* = 4, La consulta debe incluir el nombre del equipo, tipo de participación e instrumento usado (3 pts).

```
-- Participaciones de conjunto
SELECT p.id_participacion,
       p.puntaje,
       p.tipo_participacion,
```

```

    p.instrumento,
    e.nombre AS nombre_equipo
FROM Participacion p JOIN Conjunto c
    ON p.id_conjunto = c.id_conjunto
JOIN Equipo e
    ON c.id_equipo = e.id_equipo
WHERE
    p.id_competencia = 4
AND p.tipo_participacion = 'conjunto'
UNION
    -- Participaciones individuales
SELECT p.id_participacion,
    p.puntaje,
    p.tipo_participacion,
    p.instrumento,
    e.nombre AS nombre_equipo
FROM Participacion p JOIN Gimnasta g
    ON p.id_gimnasta = g.id_gimnasta
JOIN Equipo e
    ON g.id_equipo = e.id_equipo
WHERE
    p.id_competencia = 4
AND p.tipo_participacion = 'individual';

```

- b) Genere un procedimiento almacenado que reciba como parámetro el *id_juez* y *id_competencia*, y entregue como resultado todas las evaluaciones realizadas por el juez en la competencia (1.5 pts)

```

CREATE OR REPLACE FUNCTION ObtenerEvaluacionesPorJuezYCompetencia(
    p_id_juez INT,
    p_id_competencia INT
)
RETURNS TABLE (
    id_evaluacion INT,
    id_participacion INT,
    puntaje DECIMAL,
    comentarios TEXT,
    id_competencia INT,
    id_gimnasta INT,
    id_conjunto INT,
    tipo_participacion VARCHAR,
    instrumento VARCHAR,

```

```

        puntaje_participacion DECIMAL
    ) AS $$
BEGIN
    RETURN QUERY
    SELECT
        e.id_evaluacion,
        e.id_participacion,
        e.puntaje,
        e.comentarios,
        p.id_competencia,
        p.id_gimnasta,
        p.id_conjunto,
        p.tipo_participacion,
        p.instrumento,
        p.puntaje AS puntaje_participacion
    FROM
        Evaluacion e JOIN Participacion p
        ON e.id_participacion = p.id_participacion
    WHERE
        e.id_juez = p_id_juez
        AND p.id_competencia = p_id_competencia;
END;
$$ LANGUAGE plpgsql;

```

- c) Genere un procedimiento almacenado que reciba como parámetro *id_competencia*, *tipo_participacion* e *instrumento* que entregue como resultado los primeros tres lugares de cada competencia, clasificada por *tipo_participacion* e *instrumento* (1.5 pts).

```

CREATE OR REPLACE FUNCTION ObtenerPrimerosTresLugares(
    p_id_competencia INT,
    p_tipo_participacion VARCHAR,
    p_instrumento VARCHAR)
RETURNS TABLE (
    id_participacion INT,
    id_competencia INT,
    id_gimnasta INT,
    id_conjunto INT,
    tipo_participacion VARCHAR,
    instrumento VARCHAR,
    puntaje DECIMAL
) AS $$

```

```
BEGIN
  RETURN QUERY
  SELECT
    p.id_participacion,
    p.id_competencia,
    p.id_gimnasta,
    p.id_conjunto,
    p.tipo_participacion,
    p.instrumento,
    p.puntaje
  FROM
    Participacion p
  WHERE
    p.id_competencia = p_id_competencia
    AND p.tipo_participacion = p_tipo_participacion
    AND p.instrumento = p_instrumento
  ORDER BY
    p.puntaje DESC
  LIMIT 3;
END;
$$ LANGUAGE plpgsql;
```

Pregunta 3 - Conceptos

En el contexto del proyecto semestral responda las siguientes preguntas:
Dadas las tablas:

- Persona(RUN:Integer;DV:char;Nombre:text;email_personal:text;email_institucional:text;...)
 - Estudiante(RUN:Integer FK Persona.RUN; Número de alumno:Integer; ...)
 - Notas(Número de alumno:Integer; Nombre:text; Cohorte:text;sigla:text;Asignatura:text; periodo:text; Nota:float;Calificación:text; ...)
- a) (2 pts) Indique brevemente el proceso (cada paso o tarea) que se realiza, desde la confección del modelo E/R pasando por la lectura de los datos (en Excel o CSV), hasta que los datos están cargados en la Base de Datos (detalle las tareas del cargador, en pseudocódigo)
- a.- Desde las reglas de negocio determinar la Entidades y Relaciones
 - b.- Transformar el modelo E/R en un esquema de la base de datos
 - c.- Normalizar el modelo relacional en 3FN
 - d.- Crear el esquema de base de datos usando DDL
 - e.- Por cada tab del Excel o archivo CSV se lee cada linea y
 - e.1.- Extraer los atributos y verificar la validez del formato
 - e.2.- Si hay error se trata de corregir, si no se puede se guarda en archivo de errores
 - e.3.- Se realizan los INSERT de los atributos en sus tablas correspondientes
 - e.4.- Si hay error se trata de corregir, si no se puede se guarda en archivo de errores
 - f.- Se analiza el archivo de errores y por cada uno se determina la acción entre: Corregir, Marcar como inválido, Eliminar en cascada
 - g.- Usando el cargador.php se cargan los registros corregidos
- b) (2 pts) Indique cómo se realiza en PHP+SQL una consulta para obtener todos nombres de los estudiantes que tienen 2 números de alumno.

1. Establecer la conexión

```
$php $db_connection = pg_connect("host=server dbname=dbname user=username password=password")
```

2. Construir el query

```
$query = " SELECT persona.RUN, persona.Nombre COUNT(estudiante.numero de alumno)
```

```
FROM Persona
```

```
Select Persona.nombre From Persona, Estudiante Where persona.RUN =
```

```
estudiante.RUN  
GROUP BY Persona.RUN  
HAVING COUNT(estudiante.numero de alumno) = 2  
$resultado = pg_query($db_connection, $query);
```

```
3. Desplegar el query  
while ($row = pg_fetch_row($resultado)) {  
    echo "$row[0] $row[1] ";  
    echo "<br />\n";  
}
```

- c) (0.5 pts) En la consulta `SELECT A FROM T WHERE C`, ¿qué efecto adverso tiene si `A="*;drop table Estudiante"`? ¿qué se debe implementar para evitar este efecto?

Si es que los permisos lo permiten se realizan los siguientes pasos
- Si estudiante tiene un atributo referencia de FK de alguna otra tabla da error
Sino borra la tabla
Se debe sanitizar el input de A de modo de no permitir la ejecución del drop el cual se distingue por el ";"

- d) (0.5 pts) Indique las llaves candidatas y primaria de las siguientes tablas

- Persona RUN, email institucional eventualmente puede ser también email personal
- Notas Numero de alumno-sigla-periodo candidata=primaria

- e) (0.5 pts) Indique los índices clustered y unclustered que se deben aplicar a la tabla estudiante para hacer las consultas más rápidas?

Clustered por número de alumno que es PK
Secundario unclustered por RUN (o nombre)

- f) (0.5 pts) Respecto a la transacción de carga de las notas del curso (E3) qué efecto tiene el poner el `CREATE TEMP TABLE` acta inmediatamente luego del `BEGIN TRANSACTION` y antes del `COMMIT` versus ponerlo antes del `BEGIN TRANSACTION - COMMIT`

En el primer caso la tabla temporal se crea al inicio de la transacción por lo que en caso de falla el undo deshace esa transacción, o sea la borra. En el segundo la tabla se crea independiente de la transacción por lo que si falla seguirá creada.

Pregunta 4 - Privacidad

- a) ¿En qué consiste el k-anonimato, l-dispersión y cómo actúan en conjunto para resguardar la privacidad? (2 pts)

Solución: Un conjunto de datos es k-anónimo si para cada combinación de atributos identificables se compartirá por al menos k individuos.

La l-diversidad exige es que cada "grupo" identificable k-anónimo contenga al menos l valores distintos para el atributo sensible.

- b) Relacione el concepto y su definición (2 pts)

- Identificador explícito (IdE)
- Cuasi identificador (CId)
- Atributos sensibles (AS)
- Atributos no-sensibles (AnS)

Definición	Concepto
Conjunto de atributos cuyo valor se desea proteger.	
Conjunto de atributos que individualmente identifican de manera explícita al "dueño" de un registro.	
Conjunto de atributos que no tienen un carácter privado (y por lo tanto no se busca proteger).	
Conjunto de atributos que colectivamente tienen el potencial de identificar al dueño de un registro.	

Solución

AS

IdE

AnS

CId

- c) En la siguiente tabla identifique los tipos de atributos y dé un ejemplo con k=3, l=2 (2 pts)

IdE ID

CId Edad Género Código Postal

AS Diagnóstico

AnS no hay

ID	Edad	Género	Código Postal	Diagnóstico
1	34	M	12345	Diabetes
2	45	F	12345	Hipertensión
3	34	F	12345	Asma
4	50	M	67890	Diabetes
5	45	M	67890	Hipertensión
6	34	F	67890	Asma

ID	Edad	Género	Código Postal	Diagnóstico
1	30-40	M	123**	Diabetes
2	40-50	F	123**	Hipertensión
3	30-40	F	123**	Asma
4	40-50	M	678**	Diabetes
5	40-50	M	678**	Hipertensión
6	30-40	F	678**	Asma

Bonus Full Text Search & retrieval - 1/2 punto

Tienes un conjunto de documentos y deseas calcular el TF (Frecuencia de Término) y el IDF (Frecuencia Inversa de Documento) para cada término en los documentos.

- **Documento 1:** “El gato come pescado”
- **Documento 2:** “El perro come carne”
- **Documento 3:** “El gato y el perro son amigos”

a) Calcular el TF (Frecuencia de Término): para los términos gato y perro (0,1 pts)

Solución: TF es Número de veces que aparece un Término en un documento, dividido por el número total de términos en el documento

Termino	gato	perro
Documento 1	1/4	0/4
Documento 2	0/4	1/4
Documento 3	1/7	1/7

b) Calcular el IDF (Frecuencia Inversa de Documento): para los términos gato y perro.
Nota: $\log(3/2) = 0,1761$; $\log(3) = 0,4771$ (0.1 pts)

El IDF se calcula como el logaritmo del número total de documentos dividido por el número de documentos que contienen el término.

Termino	gato	perro
IDF	$\log(3/2)$	$\log(3/2)$

c) Calcular el TF-IDF: para los términos gato y perro. (0.1 pts)

El TF-IDF se calcula multiplicando el TF por el IDF.

Termino	gato	perro
Documento 1	$0.25 \times 0.1761 = 0.0440$	$0 \times 0.1761 = 0$
Documento 2	$0 \times 0.1761 = 0$	$0.25 \times 0.1761 = 0.0440$
Documento 3	$0.14 \times 0.1761 = 0.0246$	$0.14 \times 0.1761 = 0.0246$

d) ¿Para qué sirve determinar TF-IDF? Dé un ejemplo usando los datos calculados en las preguntas anteriores (0.2 pts)

TF-IDF sirve para determinar si al buscar un término, que documentos se proponen como los más relevantes y en qué prioridad.

Si lo aplicamos a los datos usados, si buscáramos el término gato, el documento más relevante sería el Documento 1 y el segundo es el Documento 2, siendo irrelevante en el caso del Documento 3.

Torpedo

SQL

Sintaxis consulta SQL

```
SELECT [ALL | DISTINCT] * | [COUNT, MAX, MIN, SUM, AVG](col1), col2, ...
      [AS alias_column]
FROM table1
      [JOIN table2 ON table1.common_field = table2.common_field]
WHERE condition1 [AND | OR condition2]
      [BETWEEN value1 AND value2]
      [IN (value1, value2, ...)]
      [LIKE pattern]
      [= | <> | <= | >= | < | >] value
GROUP BY column1, column2, ...
HAVING aggregate_condition [COUNT, MAX, MIN, SUM, AVG]
ORDER BY column1 [ASC | DESC]
UNION | INTERSECT | EXCEPT
[SELECT column1 FROM another_table]
LIMIT row_count;
```

Sintaxis Stored Procedure

```
CREATE OR REPLACE FUNCTION <nombre_función> (<argumentos>)
RETURNS [VOID | TABLE(...) ] AS $$
DECLARE
    <declaración_variables>
BEGIN
    <sentencias_SQL>
END; $$ language plpgsql
```

Sintaxis PL/pgSQL

```
BEGIN
    IF <condición> THEN
        <cuerpo_if>
    ELSE
        <cuerpo_else>
    END IF;
    INSERT INTO <tabla> VALUES (<tupla_valores>);
    FOR <variable> IN <iterable>
    LOOP
        <cuerpo_loop>
    END LOOP;
    RETURN QUERY <consulta_SQL>;
END;
```