Proyecto

Creación de Base de Datos Relacional

Miguel Jiménez De La Torre
Juan Pablo Rizzi
Ángel Gómez Alonso

Fabián González Martín
Rebeca Díaz-Montenegro Sánchez

Datos de entrada

Nombre	Rol	Vertical	Promocion	Campus	Modalidad
Noa Yáñez	TA	DS	Septiembre	Madrid	Presencial
Saturnina Benitez	TA	DS	Septiembre	Madrid	Presencial
Anna Feliu	TA	FS	Septiembre	Madrid	Presencial
Rosalva Ayuso	TA	FS	Septiembre	Valencia	Presencial
Ana Sofía Ferrer	TA	FS	Febrero	Valencia	Presencial

Nombre	Email	Promoción	Fecha_comienzo	Campus	Proyecto_HLF	Proyecto_EDA	Proyecto_BBDD
Jafet Casals	Jafet_Casals@gmail.com	Septiembre	18/09/2023	Madrid	Apto	No Apto	Apto
Jorge Manzanares	Jorge_Manzanares@gmail.com	Septiembre	18/09/2023	Madrid	Apto	No Apto	Apto
Onofre Adadia	Onofre_Adadia@gmail.com	Septiembre	18/09/2023	Madrid	Apto	Apto	Apto
Merche Prada	Merche_Prada@gmail.com	Septiembre	18/09/2023	Madrid	Apto	No Apto	No Apto
Pilar Abella	Pilar_Abella@gmail.com	Septiembre	18/09/2023	Madrid	Apto	No Apto	Apto

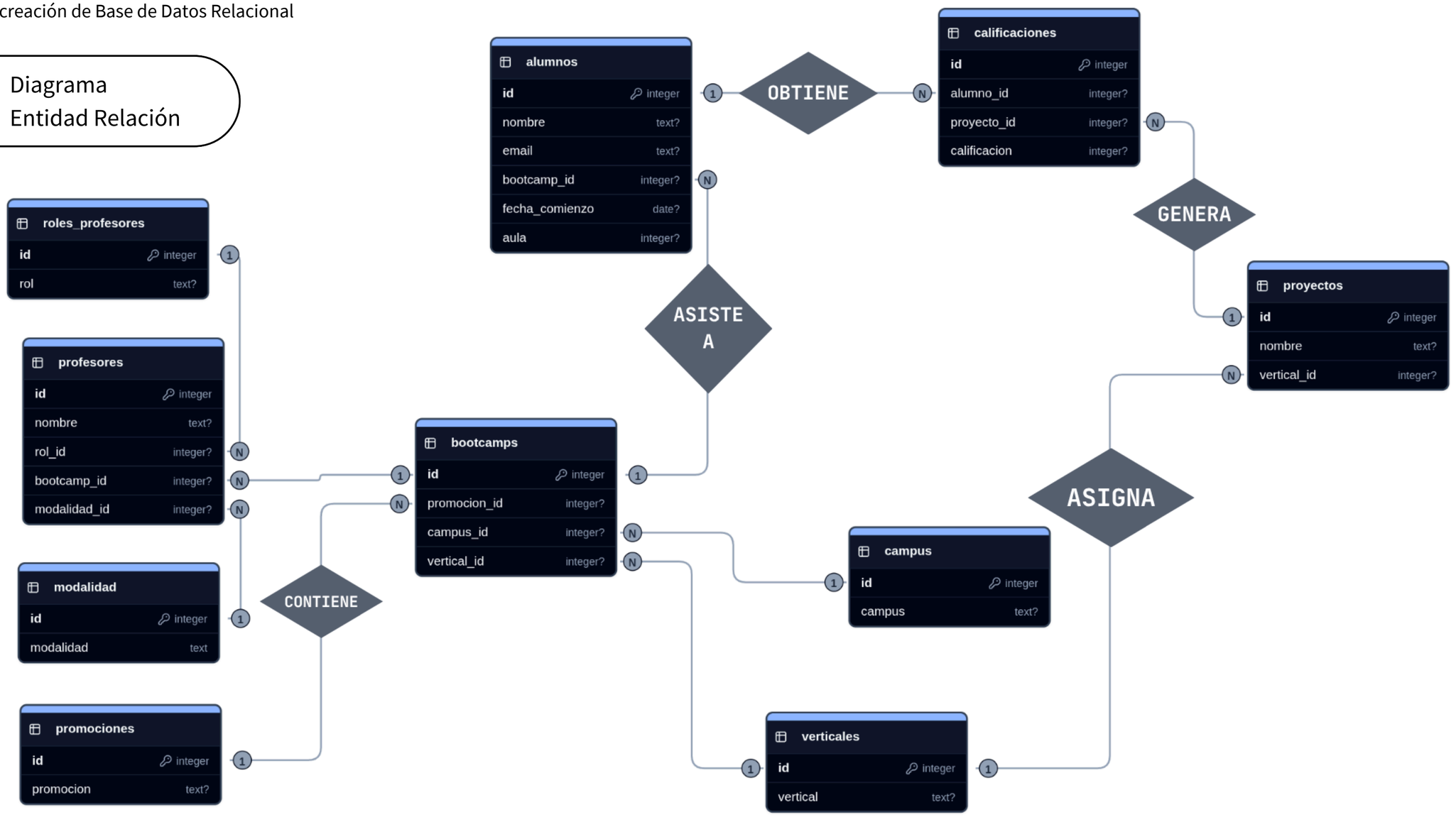
Proyecto de creación de Base de Datos Relacional

Tareas desarrolladas

1. Diagrama Entidad Relación
2. Modelo Lógico
3. Base de datos
4. Consultas

Proyecto de creación de Base de Datos Relacional

01. Diagrama Entidad Relación



Proyecto de creación de Base de Datos Relacional

01. Diagrama Entidad Relación

02. Modelo lógico

Proyecto de creación de Base de Datos Relacional

01. Diagrama Entidad Relación

02. Modelo lógico

03. Base de datos

```
CREATE TABLE IF NOT EXISTS alumnos (
    id INTEGER NOT NULL PRIMARY KEY,
    nombre VARCHAR(50),
    email VARCHAR(50),
    bootcamp_id INTEGER NOT NULL,
    fecha_comienzo DATE,
    aula INTEGER,
    FOREIGN KEY (bootcamp_id) REFERENCES bootcamps(id)
)
```

```
CREATE TABLE IF NOT EXISTS profesores (
    id SERIAL PRIMARY KEY,
    nombre VARCHAR(50),
    rol_id INTEGER,
    bootcamp_id INTEGER,
    modalidad_id INTEGER,
    FOREIGN KEY (rol_id) REFERENCES roles_profesores(id),
    FOREIGN KEY (bootcamp_id) REFERENCES bootcamps(id),
    FOREIGN KEY (modalidad_id) REFERENCES modalidad(id)
)
```

```
CREATE TABLE IF NOT EXISTS modalidades (
    id SERIAL PRIMARY KEY,
    modalidad VARCHAR(20) NOT NULL
```

```
CREATE TABLE IF NOT EXISTS verticales (
    id SERIAL PRIMARY KEY,
    verticales VARCHAR(10)
)
```

```
CREATE TABLE IF NOT EXISTS proyectos (
    id INTEGER NOT NULL PRIMARY KEY,
    nombre VARCHAR(50),
    vertical_id INTEGER NOT NULL,
    FOREIGN KEY (vertical_id) REFERENCES verticales(id)
)
```

Proyecto de creación de Base de Datos Relacional

01. Diagrama Entidad Relación

02. Modelo lógico

03. Base de datos

04. Consultas

¿Cuantos alumnos diferentes hay en la edición de septiembre?

¿Que campus ha acogido más alumnos?

```sql
-- Cuantos alumnos hay en la promocion de septiembre
SELECT DISTINCT (COUNT(a.id))
FROM Bootcamps b
INNER JOIN Promociones p ON b.promocion_id = p.id
INNER JOIN Alumnos a ON b.id = a.bootcamp_id
WHERE p.promocion = 'Septiembre'
```

```sql
--Que campus ha acogido más alumnos
SELECT c.campus,count(a.id)
FROM campus c
LEFT JOIN Bootcamps b ON c.id = b.campus_id
LEFT JOIN Alumnos a ON b.id = a.bootcamp_id
GROUP BY c.campus
```

Proyecto de creación de Base de Datos Relacional

01. Diagrama Entidad Relación

02. Modelo lógico

03. Base de datos

04. Consultas

- ¿Cuantos alumnos diferentes hay en la edición de septiembre?
- ¿Que campus ha acogido más alumnos?
- Alumnos con al menos un "No Apto"

```sql
-- Alumnos con al menos un "No Apto"
WITH alumnos_con_no_apto AS (    -- inicia el cte (common table expression), con el nombre de la tabla temporal
    SELECT DISTINCT alumno_id  --selecciona sin duplicados
    FROM calificaciones
    WHERE calificacion = 'No Apto'
)
SELECT
    a.id,
    a.nombre,
    a.email,
    a.bootcamp_id,
    a.aula
FROM alumnos a
INNER JOIN alumnos_con_no_apto na ON a.id = na.alumno_id -- usamos la cte (tabla temporal), con alias "na", el ON es la condicion de union
ORDER BY a.nombre;
```

Proyecto de creación de Base de Datos Relacional

01. Diagrama Entidad Relación

02. Modelo lógico

03. Base de datos

04. Consultas

- ¿Cuantos alumnos diferentes hay en la edición de septiembre?
- ¿Que campus ha acogido más alumnos?
- Alumnos con al menos un "No Apto"
- Suma de entregas por alumno
- Porcentaje de aprobados y suspensos por campus

```sql
--suma de entregas por alumno
SELECT a.nombre, count(ca.id) AS suma_entregas
FROM alumnos a
LEFT JOIN calificaciones ca ON a.id = ca.alumno_id
GROUP BY a.nombre
ORDER BY suma_entregas DESC;
```

```sql
-- % de aprobados y suspensos por campus
SELECT
    c.campus,
    ROUND(100.0 * SUM(CASE WHEN LOWER(ca.calificacion) = 'apto' THEN 1 ELSE 0 END) / COUNT(*), 2) AS porcentaje_apto,
    ROUND(100.0 * SUM(CASE WHEN LOWER(ca.calificacion) = 'no apto' THEN 1 ELSE 0 END) / COUNT(*), 2) AS porcentaje_no_apto
FROM Alumnos a
JOIN Bootcamps b ON a.bootcamp_id = b.id
LEFT JOIN Calificaciones ca ON a.id = ca.alumno_id
LEFT JOIN Campus c ON b.campus_id = c.id
GROUP BY c.campus;
```

Proyecto de creación de Base de Datos Relacional

01. Diagrama Entidad Relación

02. Modelo lógico

03. Base de datos

04. Consultas

Gracias por la atención

Miguel Jiménez De La Torre
Juan Pablo Rizzi
Ángel Gómez Alonso

Fabián González Martín
Rebeca Díaz-Montenegro Sánchez