

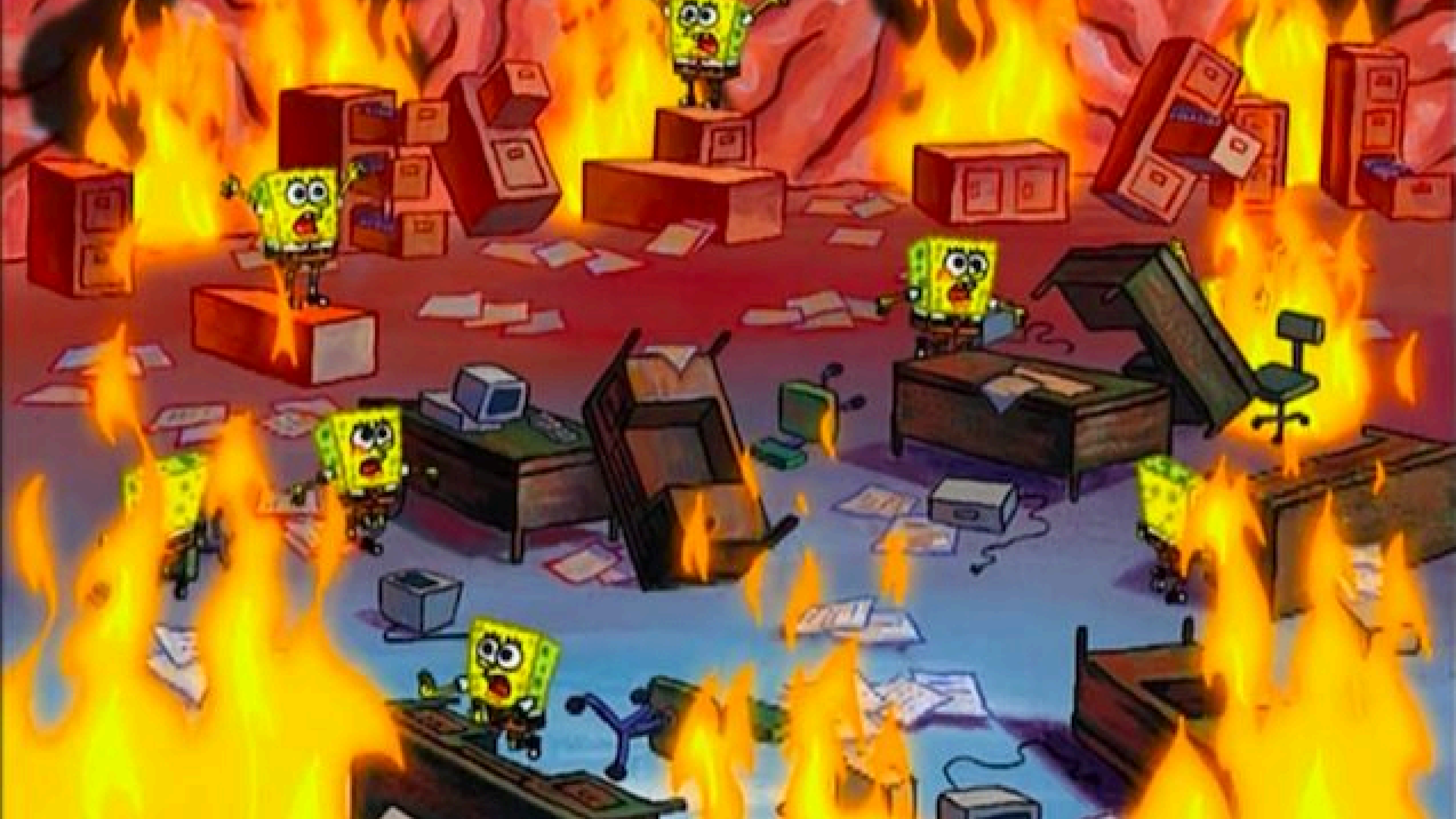


# CREACIÓN DE BB.DD ACADÉMICA

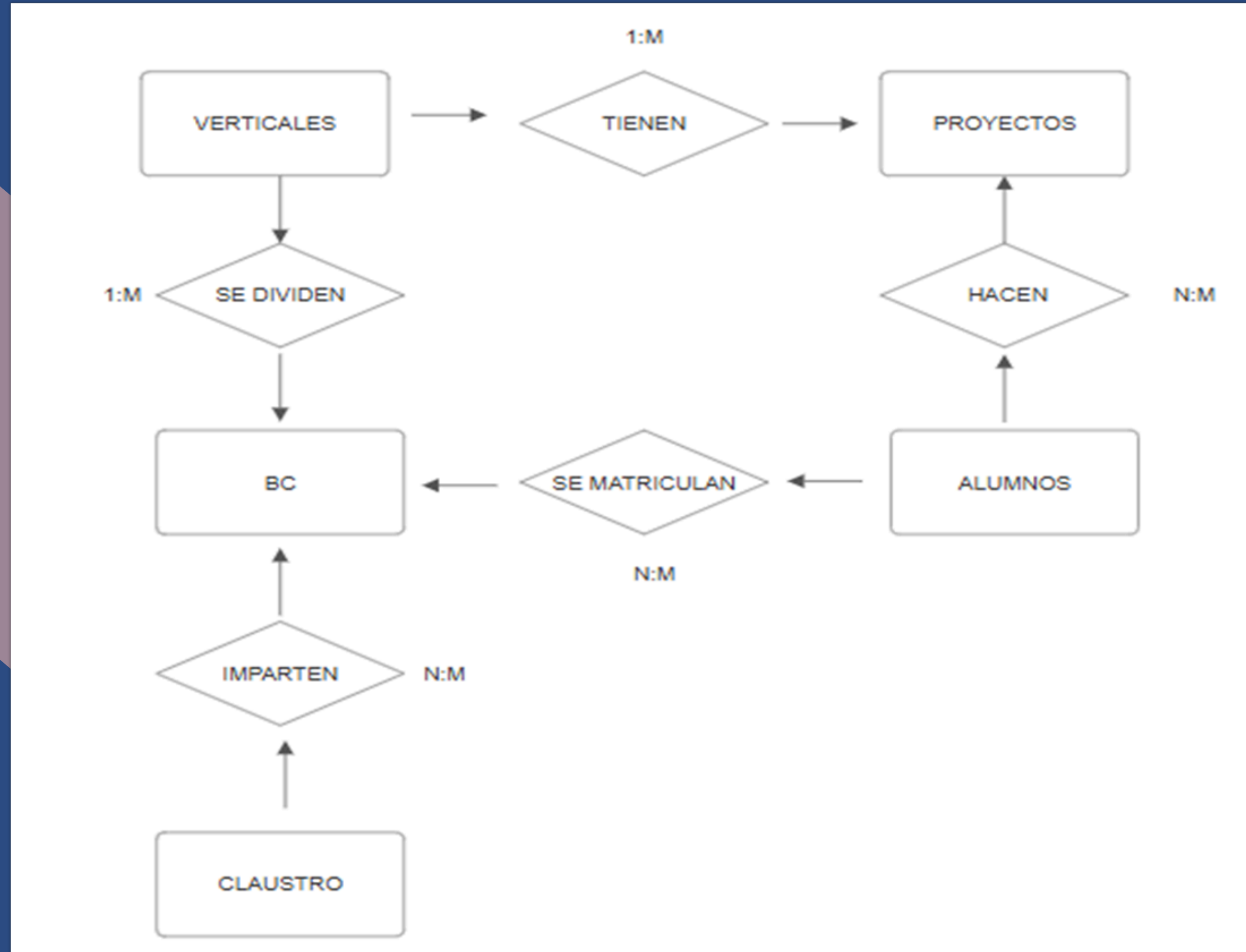
MIEMBROS: Nacho , Irene, Yanelis

# Objetivos

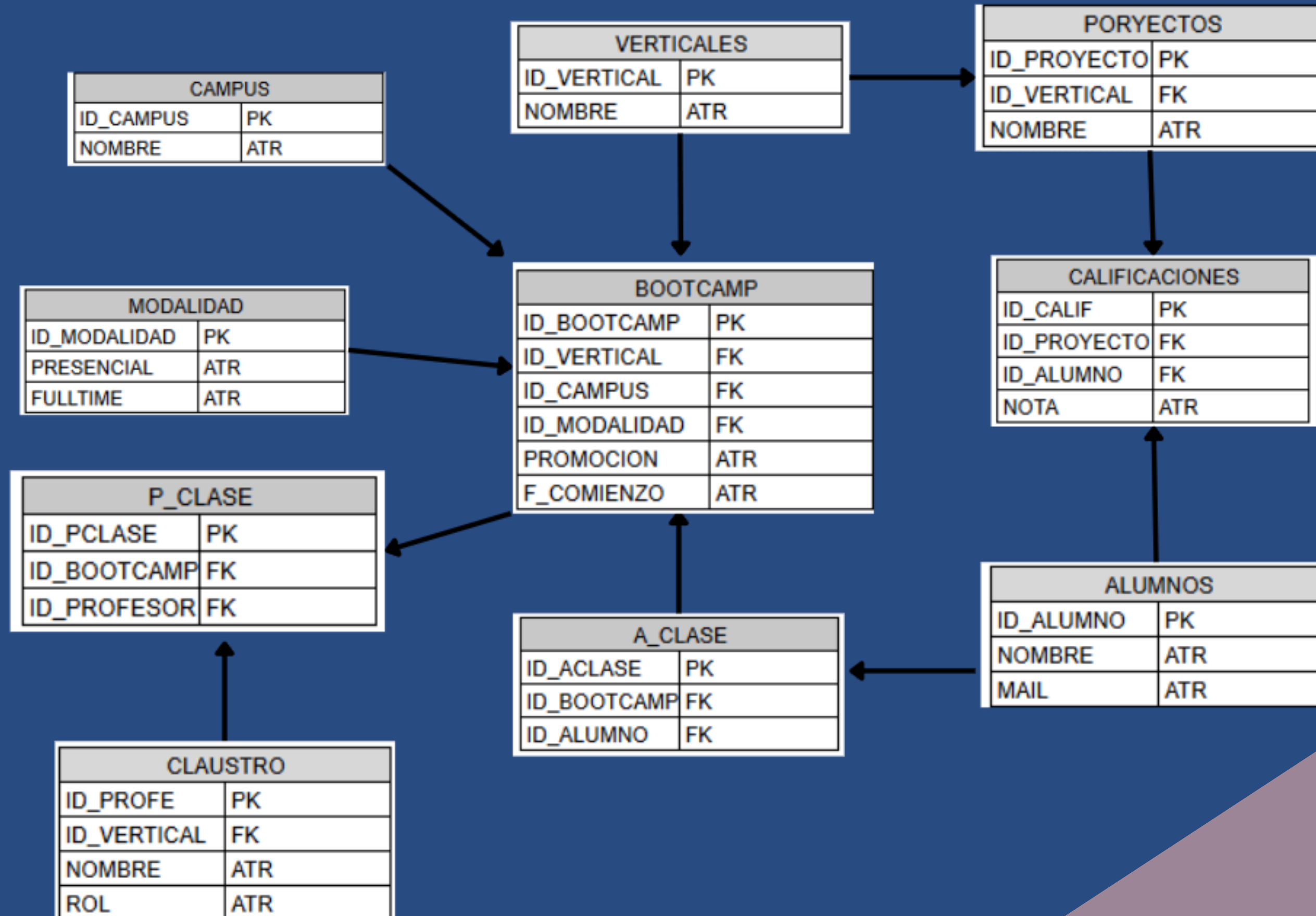
- Generar una base de datos en SQL para una institución académica.
- Optimizar el almacenamiento y la gestión de la información.
- Manejar datos esenciales de las diferentes entidades.
- Buscar solución escalable y eficiente que facilite la administración de los cursos (bootcamps).
- Mejorar la toma de decisiones mediante la consulta de datos precisos.
- Permitir la automatización de procesos



# DIAGRAMA ENTIDAD - RELACIÓN



# MODELO LÓGICO



# PASOS

- Creación de servidor y base de datos.
- Elaboración de tablas para carga.
- Importación de datos.

# CREACIÓN DE TABLAS

## 8. CALIFICACIONES

```
In [257]: total_clases["Calificación"] = total_clases["Calificación"] .replace({'Apto': True, 'No Apto': False})#cambiamos a bool los da
```

```
In [258]: calificaciones1=pd.DataFrame({"nombre_alumnos":total_clases["Nombre"],"nombre_proyecto":total_clases["Proyecto"],"Nota":total_
calificaciones1.insert(0, 'id_calificaciones', range(1, len(calificaciones1) + 1))
```

```
In [259]: calificaciones= pd.merge(calificaciones1, alumnos, left_on='nombre_alumnos', right_on='Nombre', how='left')
calificaciones= pd.merge(calificaciones, proyectos, left_on='nombre_proyecto', right_on='Proyecto', how='left')
calificaciones=calificaciones.loc[:,["id_calificaciones","id_alumnos","id_proyecto","Nota"]]
```

```
In [260]: calificaciones.head()
```

```
Out[260]:
```

	id_calificaciones	id_alumnos	id_proyecto	Nota
0	1	1	7	True
1	2	2	7	True
2	3	3	7	True
3	4	4	7	True
4	5	5	7	True

# QUERIES Y SCRIPT

```
)CREATE TABLE p_clase (  
    id_pclase    SMALLINT PRIMARY KEY,  
    id_bootcamp  SMALLINT,  
    id_profesor  VARCHAR(50),  
    CONSTRAINT fk_bootcamp FOREIGN KEY (id_bootcamp) REFERENCES bootcamp (id_bootcamp),  
    CONSTRAINT fk_profe FOREIGN KEY (id_profe) REFERENCES claustro (id_profe)  
)CREATE TABLE campus (  
    id_campus    SMALLINT PRIMARY KEY,  
    nombre       VARCHAR(50)  
)CREATE TABLE modalidad (  
    id_modalidad SMALLINT PRIMARY KEY,  
    presencial    BOOLEAN,  
    fulltime      BOOLEAN  
);
```

```
dic_sql={"campus":campus,  
        "verticales":verticales,  
        "modalidad":modalidad,  
        "claustro":claustro,  
        "bootcamp":bootcamp,  
        "proyectos":proyectos,  
        "p_clase":p_clase,  
        "alumnos":alumnos,  
        "calificaciones":calificaciones,  
        "a_clase":a_clase  
}
```

```
from sqlalchemy import create_engine  
engine = create_engine('postgresql://school_db_hkla_user:S8tv0M5YVinPoWddRVHG8Q4qMr3TFSek@dpg-cu8fnnl6l47c738hj8v0-a.frankfurt  
def upload_tables(dict, engine):  
    for table_name, dataframe in dict.items():  
        dataframe.to_sql(  
            name=table_name, # Use the key as the table name  
            con=engine,      # Database engine  
            if_exists="append", # If the table exists, append  
            index=False      # Do not write the index  
        )
```



# CONCLUSIONES

- Organización estructurada de la información.
- Normalización de datos.
- Relaciones claras entre tablas.
- Facilidad para gestionar la información.
- Escalabilidad y adaptabilidad.
- Seguridad e integridad de los datos.
- Posibilidades de análisis y reportes.
- Automatización y optimización de procesos.