

UNIVERSIDAD NACIONAL DE SAN AGUSTIN DE AREQUIPA
ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS
ASIGNATURA: TOPICOS AVANZADOS EN BASES DE DATOS (E)

NOMBRE : Yoset Cozco Mauri

CÓDIGO : 20153584

Dado el siguiente esquema conceptual diseñar una base de datos orientada a objetos.

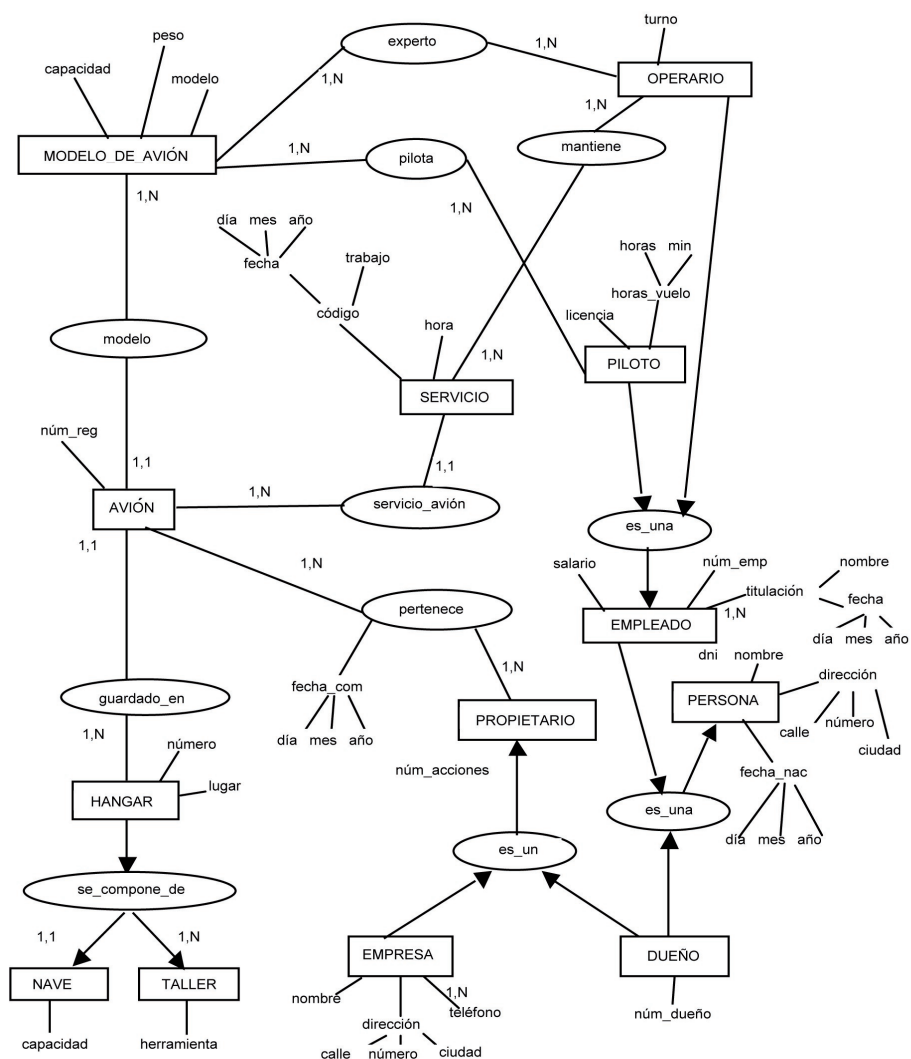


Tabla "Propietario":

- **id:** Identificador único del propietario.
- **num_acciones:** Número de acciones que posee el propietario.
- **fecha_com:** Fecha de compra.

Tabla "Persona":

- propietario_id: Identificador único del propietario asociado a la persona.
- nombre: Nombre de la persona.
- dni: Número de identificación personal.
- direccion_calle: Nombre de la calle de la dirección de la persona.
- direccion_numero: Número de la dirección de la persona.
- direccion_ciudad: Nombre de la ciudad de la dirección de la persona.
- fecha_nac: Fecha de nacimiento de la persona.

Tabla "Empresa":

- propietario_id: Identificador único del propietario asociado a la empresa.
- nombre: Nombre de la empresa.
- direccion_calle: Nombre de la calle de la dirección de la empresa.
- direccion_numero: Número de la dirección de la empresa.
- telefono: Número de teléfono de la empresa.

Tabla "Empleado":

- id: Identificador único del empleado.
- salario: Salario del empleado.
- num_emp: Número de empleado.
- titulacion_nombre: Nombre de la titulación del empleado.
- titulacion_fecha: Fecha de obtención de la titulación del empleado.

Tabla "Piloto":

- empleado_id: Identificador único del empleado asociado al piloto.
- licencia: Licencia del piloto.
- horas_vuelo: Horas de vuelo acumuladas por el piloto.
- minutos_vuelo: Minutos de vuelo acumulados por el piloto.

Tabla "ModeloAvion":

- id: Identificador único del modelo de avión.
- capacidad: Capacidad del avión (número de pasajeros, por ejemplo).
- peso: Peso del avión.
- modelo: Modelo del avión.

Tabla "ExpertosModelo":

- modelo_avion_id: Identificador único del modelo de avión asociado al experto.
- empleado_id: Identificador único del empleado experto en el modelo de avión.

Tabla "Hangar":

- id: Identificador único del hangar.
- numero: Número del hangar.
- lugar: Lugar donde se encuentra ubicado el hangar.

Tabla "Nave":

- hangar_id: Identificador único del hangar asociado a la nave.
- capacidad: Capacidad de la nave (número de aviones que puede albergar, por ejemplo).

Tabla "Taller":

- id: Identificador único del taller.
- hangar_id: Identificador único del hangar asociado al taller.
- herramienta: Descripción de la herramienta utilizada en el taller.

Tabla "ServicioAvion":

- id: Identificador único del servicio del avión.
- hora: Hora del servicio del avión.
- codigo_trabajo: Código del trabajo realizado en el avión.
- codigo_fecha: Fecha del código de trabajo del avión.

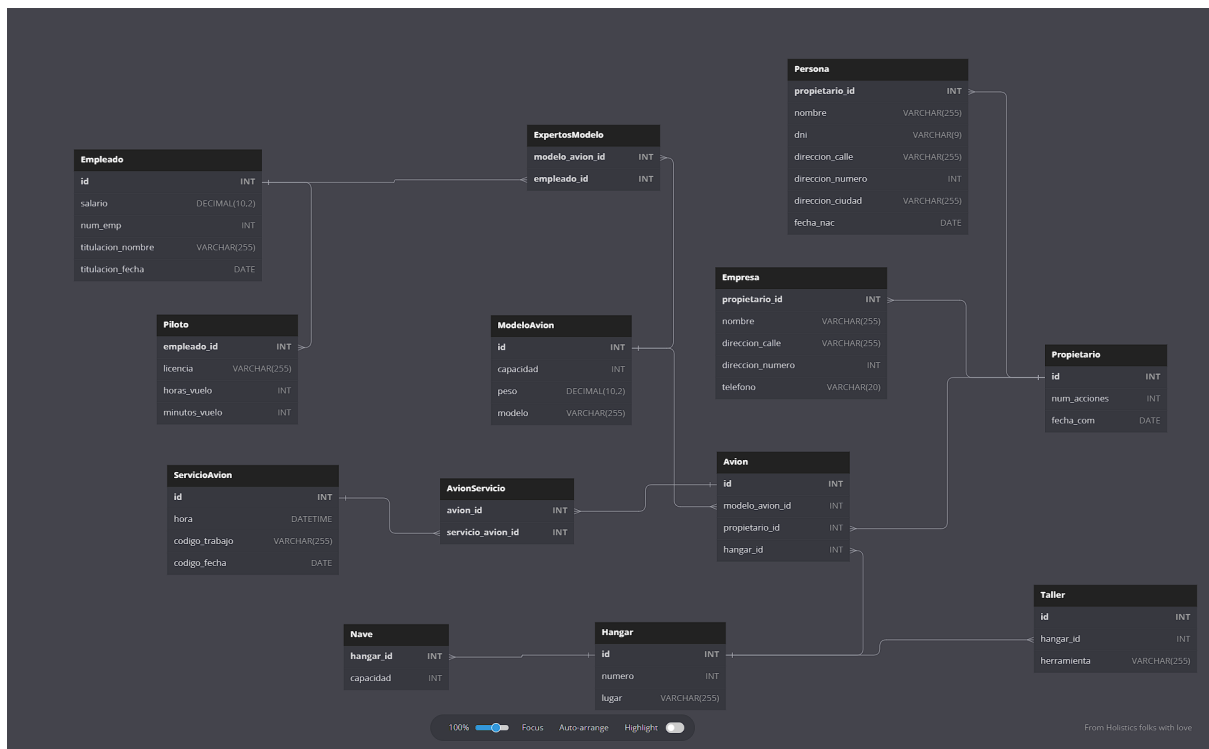
Tabla "Avion":

- id: Identificador único del avión.
- modelo_avion_id: Identificador único del modelo de avión asociado al avión.
- propietario_id: Identificador único del propietario asociado al avión.
- hangar_id: Identificador único del hangar asociado al avión.

Tabla "AvionServicio":

- avion_id: Identificador único del avión asociado al servicio del avión.
- servicio_avion_id: Identificador único del servicio del avión asociado al avión.

Diseño de la base de datos



Implementación en Python:

```
from datetime import datetime
from typing import List, Tuple

class ServicioAvion:
    def __init__(self, hora: datetime, codigo: Tuple[str, datetime]):
```

```

        self.hora = hora
        self.codigo = codigo

class Propietario:
    def __init__(self, num_acciones: int, fecha_com: datetime):
        self.num_acciones = num_acciones
        self.fecha_com = fecha_com

class Persona(Propietario):
    def __init__(self, nombre: str, dni: str, direccion: Tuple[str, int, str],
fecha_nac: datetime, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.nombre = nombre
        self.dni = dni
        self.direccion = direccion
        self.fecha_nac = fecha_nac

class Empresa(Propietario):
    def __init__(self, nombre: str, direccion: Tuple[str, int], telefono: str,
*args, **kwargs):
        super().__init__(*args, **kwargs)
        self.nombre = nombre
        self.direccion = direccion
        self.telefono = telefono

class Nave:
    def __init__(self, capacidad: int):
        self.capacidad = capacidad

class Taller:
    def __init__(self, herramienta: str):
        self.herramienta = herramienta

class Hangar:
    def __init__(self, numero: int, lugar: str, nave: Nave, talleres:
List[Taller]):
        self.numero = numero
        self.lugar = lugar
        self.nave = nave
        self.talleres = talleres

class Empleado:
    def __init__(self, salario: float, num_emp: int, titulacion: Tuple[str,
datetime]):

```

```

        self.salario = salario
        self.num_emp = num_emp
        self.titulacion = titulacion

class Piloto(Empleado):
    def __init__(self, licencia: str, horas_vuelo: Tuple[int, int], *args,
**kwargs):
        super().__init__(*args, **kwargs)
        self.licencia = licencia
        self.horas_vuelo = horas_vuelo

class ModeloAvion:
    def __init__(self, capacidad: int, peso: int, modelo: str, expertos:
List[Empleado], pilotos: List[Piloto]):
        self.capacidad = capacidad
        self.peso = peso
        self.modelo = modelo
        self.expertos = expertos
        self.pilotos = pilotos

class Avion:
    def __init__(self, modelo_avion: ModeloAvion, servicios:
List[ServicioAvion], propietario: Propietario, hangar: Hangar):
        self.modelo_avion = modelo_avion
        self.servicios = servicios
        self.propietario = propietario
        self.hangar = hangar

# Ejemplo de cómo usar las clases

# Crear un propietario persona en el contexto de Perú
propietario_persona = Persona(nombre="Luis Gonzales", dni="01234567",
direccion=("Av. Principal", 456, "Lima"), fecha_nac=datetime(1985, 8, 15),
num_acciones=20, fecha_com=datetime(2022, 10, 1))

# Crear un modelo de avión en el contexto de Perú
piloto = Piloto(licencia="ABC123", horas_vuelo=(1500, 45), salario=1000)

```