



IES Luis Vives – Leganés (Madrid)

Ciclo Formativo de Grado Superior de Desarrollo de Aplicaciones Web

Bases de Datos

Autor/a:

Adrián García-Heras Herrero

Ariel Orlando Cahuaza Huanca

Víctor Marín Escribano

Curso 2024-25

Tabla de contenido

Imagen de la base de datos..... 1

Introducción..... 2

Características de la Tabla..... 2

Justificación de Diseño 4

Imagen de la base de datos

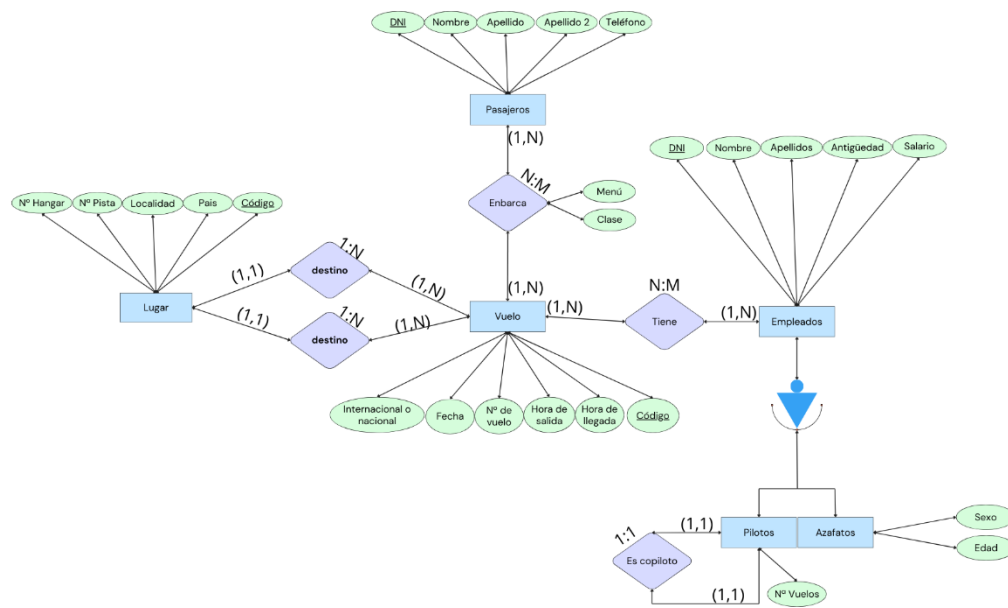


Figura 1. Modelo E/R

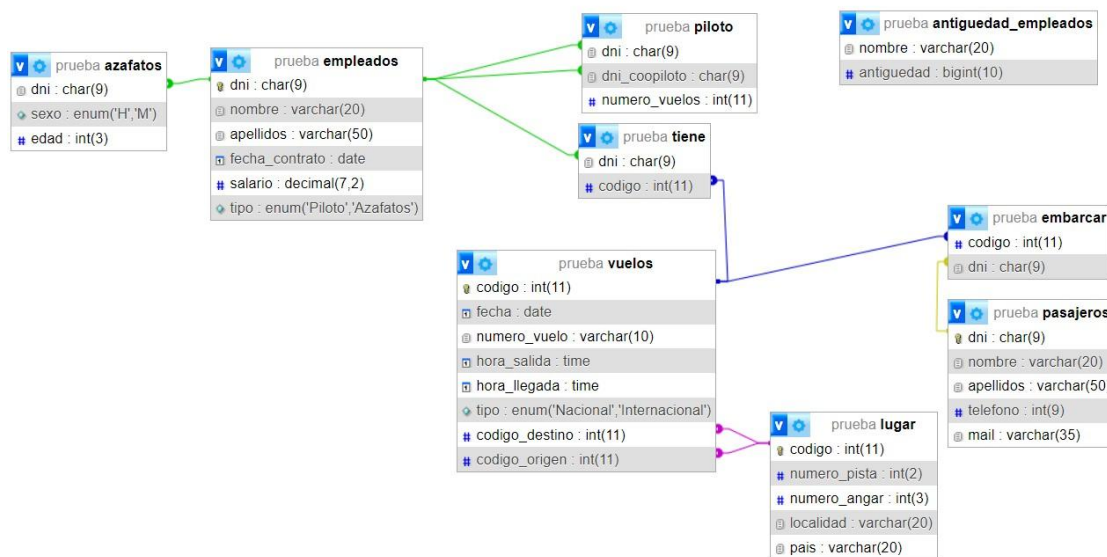


Figura 2. Esquema Tablas y relaciones foráneas

Introducción

En esta base de datos está diseñado para gestionar la información relacionada con los vuelos, pasajeros, empleados y lugares en sistema de aviación. Con la base de datos podremos gestionar de manera eficiente el seguimiento de los vuelos, las relaciones entre los pasajeros y vuelos, gestionar los datos de los empleados involucrados en las operaciones de vuelo.

Descripción del Tema Elegido y Necesidades Cubiertas

El tema elegido es como se gestionaría la información de los vuelos en un sistema de aviación. Las necesidades que va a cubrir la base de datos son:

- Almacenar y gestionar de los detalles de los vuelos.
- Registrar la información de pasajeros.
- Seguimiento de los embarques de los pasajeros en vuelos.
- Gestionar la información de empleados, incluyendo pilotos y azafatos.
- Registro de lugares como orígenes y destinos de los vuelos.

Características de la Tabla

1. Tabla vuelos

- Propósito: Almacenar información sobre cada vuelo.
- Atributos:
 - código (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único del vuelo. Genera un valor automático para cada vuelo.
 - fecha (DATE): Fecha del vuelo.
 - numero_vuelo (VARCHAR(10)): Número del vuelo. Se a elegido que sea varchar porque quiere un código con números y letras.
 - hora_salida (TIME): Hora de salida del vuelo.
 - hora_llegada (TIME): Hora de llegada del vuelo.
 - tipo (ENUM('Nacional', 'Internacional')): Tipo de vuelo.

- código_destino (INT): Referencia al lugar de destino.
- código_origen (INT): Referencia al lugar de origen.
- Claves foráneas: código_destino y código_origen estas claves hacen referencia a la tabla de **Vuelos**.

2. Tabla pasajeros

- Propósito: Almacenar información sobre los pasajeros.
- Atributos:
 - dni (CHAR(9), PRIMARY KEY): Documento de identidad del pasajero. Esta limitado a 9 caracteres porque por el sistema de generación del DNI.
 - nombre (VARCHAR(20)): Nombre del pasajero.
 - apellidos (VARCHAR(50)): Apellidos del pasajero.
 - teléfono (INT(9)): Teléfono del pasajero. Se limitan a 9 dígitos para que abarque los teléfonos sin prefijo.
 - mail (VARCHAR(35)): Correo electrónico del pasajero.

3. Tabla embarcar

- Propósito: Relacionar pasajeros con vuelos en los que embarcan.
- Atributos:
 - código (INT): Referencia al vuelo.
 - dni (CHAR(9)): Referencia al pasajero.
- Claves foráneas: código y DNI hacen referencia a las claves de las tablas **Vuelos** y **Pasajeros** consecutivamente.

4. Tabla empleados

- Propósito: Almacenar información sobre los empleados del aeropuerto.
- Atributos:
 - dni (CHAR(9), PRIMARY KEY): Documento de identidad del empleado. . Esta limitado a 9 caracteres porque por el sistema de generación del DNI.
 - nombre (VARCHAR(20)): Nombre del empleado.
 - apellidos (VARCHAR(50)): Apellidos del empleado.
 - fecha_contrato (DATE): Fecha de contratación del empleado.
 - salario (NUMERIC(7,2)): Salario del empleado. Hemos considerado limitarlo a 7 dígitos de los cuales 2 son decimales.
 - tipo (ENUM('Piloto', 'Azafatos')): Tipo de empleado.

5. Tabla piloto

- Propósito: Almacenar información específica sobre los pilotos.
- Atributos:
 - dni (CHAR(9)): Referencia al piloto. . Esta limitado a 9 caracteres porque por el sistema de generación del DNI.
 - dni_copiloto (CHAR(9)): Referencia al copiloto. . Esta limitado a 9 caracteres porque por el sistema de generación del DNI.

- numero_vuelos (INT): Número de vuelos realizados.
 - Claves foráneas: DNI y dni_copiloto hacen referencia a la clave DNI de la tabla **empleados**.
6. Tabla azafatos
- Propósito: Almacenar información específica sobre los azafatos.
 - Atributos:
 - dni (CHAR(9)): Referencia al azafato. . Esta limitado a 9 caracteres porque por el sistema de generación del DNI.
 - sexo (ENUM('H', 'M')): Sexo del azafato.
 - edad (INT(2)): Edad del azafato. Se a limitado de a 2 dígitos porque consideramos que no habrá alguien con mas de 70 años.
 - Claves foráneas: DNI hace referencia a la clave DNI de la tabla **Empleados**.
7. Tabla tiene
- Propósito: Relacionar empleados con los vuelos en los que participan.
 - Atributos:
 - dni (CHAR(9)): Referencia al empleado. . Esta limitado a 9 caracteres porque por el sistema de generación del DNI.
 - código (INT): Referencia al vuelo.
 - Claves foráneas: DNI hace referencia a la tabla **Empleados**, y código a la tabla **Vuelos**.
8. Tabla lugar
- Propósito: Almacenar información sobre los lugares (aeropuertos).
 - Atributos:
 - código (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único del lugar. Se genera un código automático para identificar cada lugar que se añade a la base de datos.
 - numero_pista (INT(2)): Número de pista. No se contempla un aeropuerto con mas de 99 pistas.
 - numero_hangar (INT(3)): Número de hangar. No se contempla un aeropuerto con mas de 999 hangares.
 - localidad (VARCHAR(20)): Localidad del lugar.
 - país (VARCHAR(20)): País del lugar.

Justificación de Diseño

Las decisiones de diseño elegidas para la base de datos se escogieron buscando la integridad, eficiencia y consistencia de los datos, como reflejar las relaciones entre entidades en la base de datos.

1. Integridad Referencial: Las claves foráneas aseguran que las relaciones entre tablas sean coherentes y previenen la eliminación o modificación de datos que estén en uso en otras tablas.
2. Normalización: La base de datos está normalizada para reducir redundancias y asegurar la consistencia de los datos.
3. Vistas: Se ha creado una vista (antigüedad_empleados) para facilitar las consultas y mejorar el uso de la base de datos.

Esta estructura de base de datos nos permite gestionar de manera eficaz la información relacionada con vuelos, asegurando la integridad y consistencia de los datos, y facilitando las consultas y operaciones diarias de la base de datos.

Azafatos y pilotos tienen **ON DELETE CASCADE Y ON UPDATE CASCADE** haciendo referencia a la tabla Empleados debido a que, si eliminamos un empleado, no necesitamos almacenar la información de si fue piloto o azafato.

La tabla Tiene, usa un **ON DELETE RESTRICT Y ON UPDATE CASCADE** hacia la tabla Empleados debido a que necesitamos almacenar la información de que empleados estuvieron en cada vuelo por seguridad en caso de investigación. Y un **ON DELETE RESTRICT Y ON UPDATE CASCADE** hacia la tabla Vuelos debido a que, si un vuelo se cancela y eliminamos sus datos, los empleados siguen trabajando en la empresa por lo tanto hay que asignarlos antes de eliminar los datos del vuelo.

La tabla Embarcar, tiene un **ON DELETE RESTRICT Y ON UPDATE CASCADE** hacia la tabla Vuelo, debido a que necesitamos asignar a los pasajeros a otro vuelo si se cancelase el actual, o si quieren una devolución. Y utilizamos un **ON DELETE CASCADE Y ON UPDATE CASCADE** hacia la tabla Pasajeros debido a que si un pasajero elimina su cuenta no necesitamos guardar sus datos sobre que vuelo compro.

La tabla Vuelos tiene dos claves foráneas de la tabla Lugar, estas siendo, **ON DELETE RESTRICT Y ON UPDATE CASCADE** debido a que haya unas condiciones meteorológicas extremas u otros casos que impidan el uso de esas pistas los Vuelos serán reasignados, pero nos interesa guardar los datos de las pistas hasta que se solucione dicho problema.