Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Administración y diseño de bases de datos:

Sistema de reserva de vuelos

Alicia Guadalupe Cruz Pérez (alu0101420868@ull.edu.es)
Iván Texenery Díaz García (alu0101429762@ull.edu.es)
Javier Padilla Pío (alu0101410463@ull.edu.es)

Índice:

1. Introducción.

Los sistemas de reservas de vuelos son herramientas tremendamente extendidas en el mundo de la aviación comercial. Es por ello que creemos puede ser un buen proyecto de una base de datos completa y funcional, en el que se puedan desarrollar todos los conocimientos adquiridos en la asignatura.

2. Objetivos.

Entre los objetivos de este sistema de reserva de vuelos podemos especificar los siguientes:

- Poder realizar tanto reservas como cancelaciones en cualquier vuelo.
- Poder crear, modificar y borrar clientes para poder realizar reservas.
- Poder “dar de alta” y “dar de baja” a nuevos vuelos.
- Poder crear y borrar modificaciones.
- Poder añadir y borrar aeropuertos.
- Poder realizar operaciones con las aerolíneas y sus entidades dependientes(tarifas).

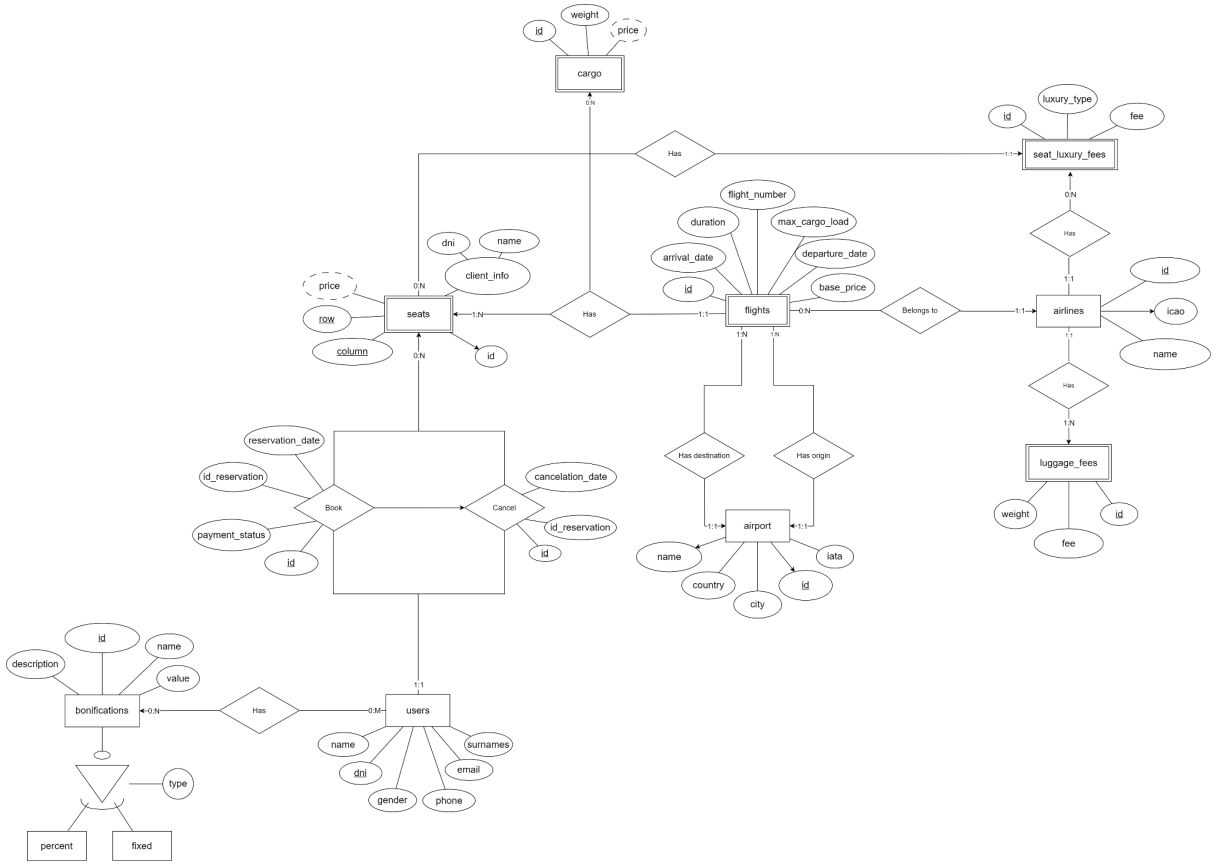
Se pretende hacer una API Rest fácil de usar sin perder de vista la funcionalidad de la misma. En este mismo documento, se proporcionará un esquema para dejar constancia de la funcionalidad de la API.

3. Especificación de requisitos.

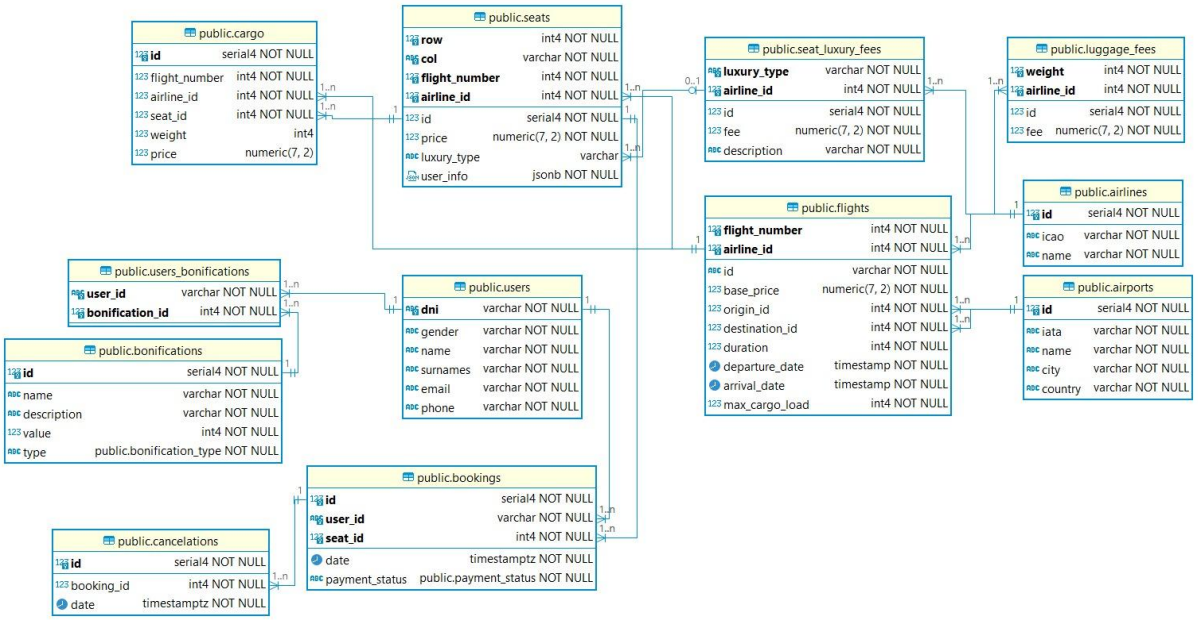
Entre los requisitos principales se establecen:

- Hacer un correcto manejo de errores, informando al usuario debidamente.
- Aprovechar lo máximo posible las url, usando los diferentes métodos http.
- Implementar en el modelo entidad-relación las restricciones solicitadas.

4. Modelo entidad-relación.



5. Modelo relacional.



6. Descripción de las entidades

- **airlines:** Información acerca de una aerolínea.
- **flights:** Información acerca de un vuelo
- **seats:** Todos los sitios de un vuelo junto con algunos metadatos como el precio del mismo, la clase e información del usuario.
- **airports:** Información de los aeropuertos registrados en el sistema.
- **cargo:** Todas las maletas asociadas a un asiento de un vuelo. Se guarda el peso de la maleta junto con su precio, calculado por medio de un trigger.
- **bookings:** Reservas realizadas en el sistema. Referencia a un asiento y un usuario.
- **cancelations:** Referencia aquellas reservas que hayan sido canceladas.
- **users:** Información de un usuario.
- **bonifications:** Bonificación disponibles en la aplicación (descuentos).
- **user_bonifications:** Relaciones entre un usuario y las bonificaciones.
- **seat_luxury_fees:** Tarifas aplicadas a los distintos tipos de asiento. Se almacena el tipo junto con la tarifa y una pequeña descripción.
- **luggage_fees:** Tarifas aplicadas a las maletas facturadas. Se almacena el peso máximo y la tarifa aplicada

7. Consultas

Vamos a comenzar insertando un nuevo vuelo para ver la creación automática del identificador.

```
INSERT INTO flights (airline_id,
                     origin_id,
                     destination_id,
                     departure_date,
                     arrival_date,
                     duration,
                     flight_number,
                     base_price,
                     max_cargo_load)
values (1, 1, 2, 'Thu, 21 Dec 2023 20:03:16 GMT', 'Thu, 21 Dec 2023 23:03:16 GMT', 180, 57, 45.0, 400);

select * from flights;
```

flights 1

select * from flights | Enter a SQL expression to filter results (use Ctrl+Space)

	id	flight_number	base_price	airline_id	origin_id	destination_id	duration	departure_date	arrival_date	max_cargo_load
1	ICAO0001101	101	200	1	1	2	120	2023-01-01 08:00:00.000	2023-01-01 10:00:00.000	500
2	ICAO002202	202	250	2	2	3	180	2023-01-02 12:00:00.000	2023-01-02 15:00:00.000	600
3	ICAO003303	303	180	3	3	4	150	2023-01-03 16:00:00.000	2023-01-03 18:30:00.000	450
4	ICAO004404	404	220	4	4	1	200	2023-01-04 20:00:00.000	2023-01-05 00:00:00.000	550
5	ICAO0001506	506	50	1	1	2	150	2023-12-21 20:03:16.000	2023-12-21 20:03:16.000	5.000
6	ICAO0001507	507	50	1	1	2	150	2023-12-21 20:03:16.000	2023-12-21 20:03:16.000	5.000
7	ICAO00156	56	45	1	1	2	180	2023-12-21 20:03:16.000	2023-12-21 23:03:16.000	400
8	ICAO00157	57	45	1	1	2	180	2023-12-21 20:03:16.000	2023-12-21 23:03:16.000	400

En la consulta anterior podemos ver como el trigger ‘update_flight_id’ ha creado automáticamente el id del vuelo a partir del número de vuelo y el ICAO de la correspondiente aerolínea.

A continuación vamos a añadir algunos asientos a este vuelo:

```
select * from flights;

insert into seats (airline_id,
                   flight_number,
                   col,
                   "row")
values
(1, 57, 'A', 1),
(1, 57, 'B', 1),
(1, 57, 'C', 1);
```

```
insert into seats (airline_id,
                   flight_number,
                   col,
                   "row",
                   luxury_type)
values
(1, 57, 'D', 1, 'FirstClass');

select * from seats where flight_number = 57;
```

seats 1

select * from seats where flight_number = 57 | Enter a SQL expression to filter results (use Ctrl+Space)

	id	row	col	price	flight_number	airline_id	luxury_type	user_info
1	30	1	A	45	57	1	[NULL]	{}
2	31	1	B	45	57	1	[NULL]	{}
3	32	1	C	45	57	1	[NULL]	{}
4	33	1	D	195	57	1	FirstClass	{}

Se puede apreciar como los precios se han rellenado automáticamente en función del precio base establecido anteriormente al crear el vuelo. Además si nos fijamos al asiento D1 se le ha aplicado también la tarifa FirstClass.

Ahora reservaremos uno de estos asientos y añadiremos algunas maletas.

```
INSERT INTO bookings (user_id, seat_id)
values ('123456789A', 30);

select * from bookings;
```

bookings 1

select * from bookings | Enter a SQL expression to filter results (use Ctrl+Space)

	id	user_id	seat_id	date	payment_status
1	1	123456789A	1	2023-01-01 00:00:00.000 +0000	fulfilled
2	2	987654321B	2	2023-01-02 00:00:00.000 +0000	fulfilled
3	3	555555555C	3	2023-01-03 00:00:00.000 +0000	pending
4	4	111111111D	4	2023-01-04 00:00:00.000 +0000	fulfilled
5	7	123456789A	30	2023-12-21 22:59:42.441 +0000	pending

```
insert into cargo (airline_id,
                   flight_number,
                   seat_id,
                   weight)
values (1, 57, 30, 20);

select * from cargo;
```

cargo 1

select * from cargo | Enter a SQL expression to filter results (use Ctrl+Space)

Grilla / Texto

	id	flight_number	airline_id	seat_id	weight	price
1	1	101	1	1	15	30
2	2	202	2	2	20	40
3	3	303	3	3	25	50
4	4	404	4	4	30	60
5	5	101	1	21	50	70
6	6	57	1	30	20	50

A continuación vamos a proceder a cancelar la reserva.

```
insert into cancelations (booking_id)
values (7);

select * from cancelations;
```

ncelations 1

lect * from cancelations | Enter a SQL expression to filter results (use Ctrl+Space)

id	booking_id	date
1	1	2023-01-02 00:00:00.000 +0000
2	3	2023-01-04 00:00:00.000 +0000
4	7	2023-12-21 23:02:36.531 +0000

Ahora que se han creado todos estos registros, si eliminamos el vuelo veremos como todo lo dependiente de este se borrará también.

```
delete from flights where flight_number = 57;

select * from seats where flight_number = 57;

select * from cargo where flight_number = 57;

select * from bookings;

select * from cancelations;
```

seats 1 | cargo 1 (2) | bookings 1 (3) | cancelations 1 (4) | Estadísticas 1

delete from flights where flight_number = 57; select * from seat | Data filter is not supported

Grilla

id	row	col	price	flight_number	airline_id	luxury_type

seats 1 | cargo 1 (2) | bookings 1 (3) | cancelations 1 (4) | Estadísticas 1

delete from flights where flight_number = 57; select * from seat | Data filter is not supported

Grilla

id	flight_number	airline_id	seat_id	weight	price

seats 1 | cargo 1 (2) | bookings 1 (3) | cancelations 1 (4) | Estadísticas 1

delete from flights where flight_number = 57; select * from seat | Data filter is not supported

Grilla / Texto

	id	user_id	seat_id	date	payment_status
1	1	123456789A	1	2023-01-01 00:00:00.000 +0000	fulfilled
2	2	987654321B	2	2023-01-02 00:00:00.000 +0000	fulfilled
3	3	555555555C	3	2023-01-03 00:00:00.000 +0000	pending
4	4	111111111D	4	2023-01-04 00:00:00.000 +0000	fulfilled

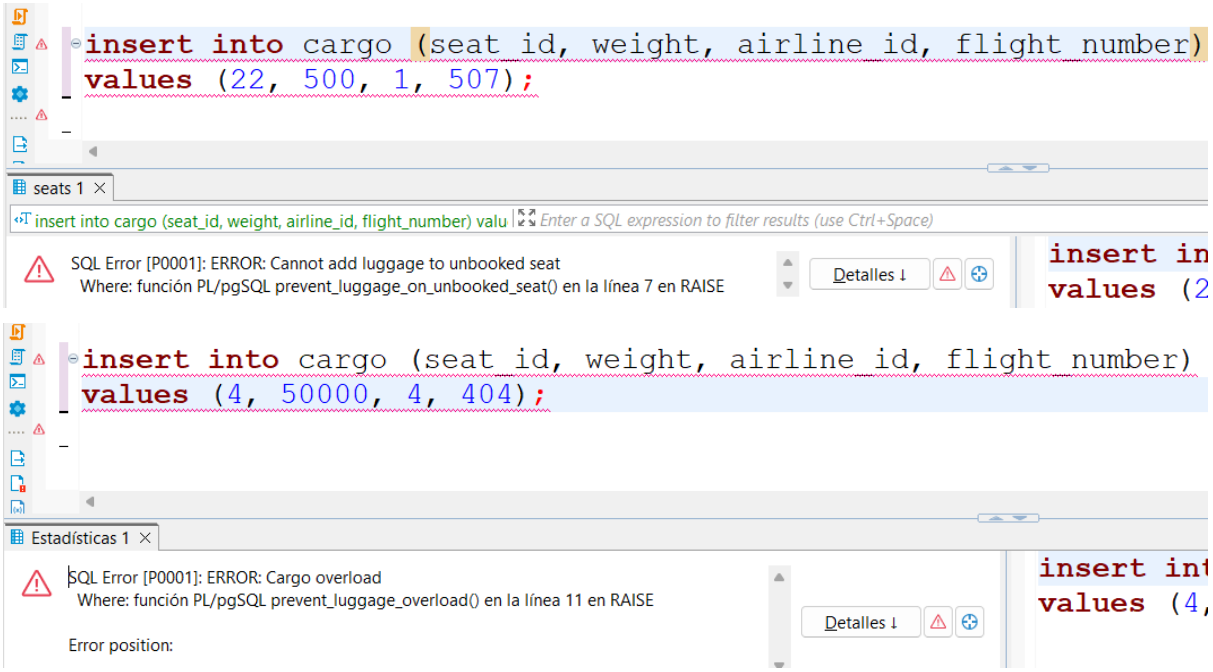
seats 1 | cargo 1 (2) | bookings 1 (3) | cancelations 1 (4) | Estadísticas 1

delete from flights where flight_number = 57; select * from seat | Data filter is not supported

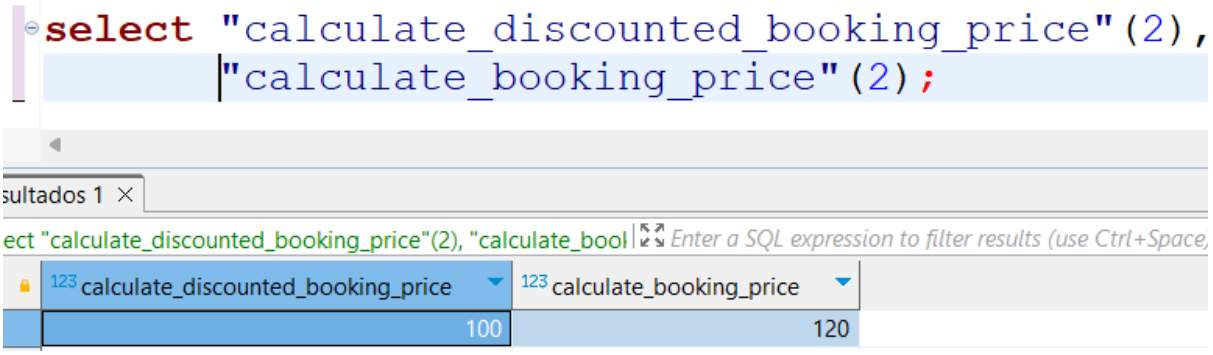
Grilla

	id	booking_id	date
1	1	1	2023-01-02 00:00:00.000 +0000
2	2	3	2023-01-04 00:00:00.000 +0000

Se puede apreciar cómo automáticamente se han eliminado todos los registros dependientes. Otras comprobaciones realizadas con los triggers son evitar añadir maletas a un asiento que no haya sido reservado y evitar que el peso de la bodega supere el peso neto del avión.



Por último se han creado dos funciones dentro de la base de datos para calcular el precio de una reserva sin aplicar las bonificaciones y aplicándolas.



8. Funcionalidades

Endpoint	**Acción**
`GET /users`	Lista los usuarios existentes
`POST /users`	Crea un nuevo usuario
`GET /users/<dni>`	Obtiene la información de un usuario
`PUT /users/<dni>`	Actualiza un usuario
`DELETE /users/<dni>`	Borra un usuario
`GET /bookings`	Lista reservas
`POST /bookings`	Crea una reserva
`GET /bookings/<id>`	Obtiene la información de una reserva
`DELETE /bookings/<id>`	Borra una reserva
`GET /airlines`	Lista las aerolíneas
`POST /airlines`	Crea una nueva aerolínea
`GET /airlines/<airline_id>`	Obtiene la información de una aerolínea
`GET /airlines/<airline_id>/luggage_fees`	Lista las tarifas de equipaje de una aerolínea
`POST /airlines/<airline_id>/luggage_fees`	Añade una tarifa de equipaje a una aerolínea
`PUT /airlines/<airline_id>/luggage_fees`	Modifica una tarifa de equipaje de una aerolínea
`DELETE /airlines/<airline_id>/luggage_fees`	Borra una tarifa de equipaje a una aerolínea

`GET /airlines/<airline_id>/luxury_fees`	Lista las tarifas del tipo de clase de una aerolínea
`POST /airlines/<airline_id>/luxury_fees`	Añade una tarifa del tipo de clase a una aerolínea
`PUT /airlines/<airline_id>/luxury_fees`	Modifica una tarifa de tipo de clase de una aerolínea
`DELETE /airlines/<airline_id>/luxury_fees`	Borra una tarifa de tipo de clase a una aerolínea
`GET /airlines/<airline_id>/flights`	Lista todos los vuelos de una aerolínea
`POST /airlines/<airline_id>/flights`	Añade un vuelo a una aerolínea
`GET /airlines/<airline_id>/flights/<id>`	Obtiene la información de un vuelo dado su número de vuelo
`DELETE /airlines/<airline_id>/flights/<id>`	Elimina un vuelo de una aerolínea dado su número de vuelo
`GET /airports`	Lista los aeropuertos
`POST /airports`	Crea un nuevo aeropuerto
`GET /bonifications`	Lista las bonificaciones
`POST /bonifications`	Crea una nueva bonificación