Base de Datos

Cambios hechos en este sprint:

Con respecto a la redundancia de datos, lo que se ha cambiado es la colección de pasajeros y la colección de vuelos, antes cada objeto pasajero de la colección pasajeros tenía un objeto vuelo con todos sus atributos y como también había una colección de vuelos se repetían los datos de vuelos, es decir que en la base de datos había dos objetos vuelo idénticos, ahora cada objeto pasajero ya no tiene el objeto vuelo, ahora lo que se ha cambiado es que los objetos vuelo de la colección vuelos, tienen una lista de objetos pasajero pero solo con su id y su nombre que son suficientes para poder identificar a un pasajero de un determinado vuelo.

Se ha añadido al objeto Node(coche) un nuevo atributo llamado destino.

Se ha añadido la documentación de la instalación de MongoDB en Debian.

Descripció general

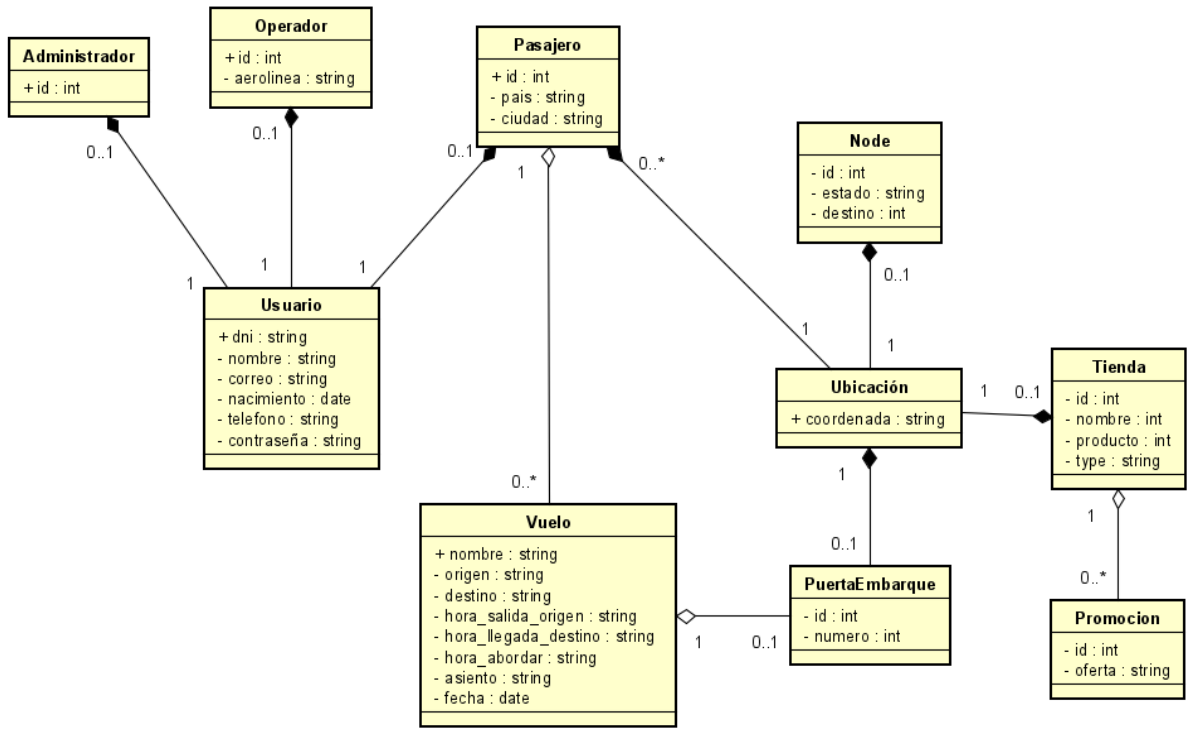
Hemos decidido por optar a un diseño de la base de datos básico. En el esquema que adjuntamos más abajo detallamos como tenemos planeado almacenar la información imprescindible y básica de nuestra Terminal 2 para los diferentes usuarios.

Conceptos

Explicación de lo que simboliza cada concepto dentro del diagrama conceptual que posteriormente se convertirán en información que se almacenará en nuestra base de datos.

Usuario: Persona común que utiliza un sistema informático.
Administrador: Usuario que tiene control absoluto de nuestro sistema inteligente.
Operador: Usuario encargado de gestionar los vuelos de las diferentes aerolíneas.
Pasajero: Usuario cliente al que facilitamos información de sus vuelos y datos.
Vuelo: Trayecto que realiza un avión de un sitio a otro.
Puerta_embarque: Información del lugar donde se accede al avión.
Ubicación: Lugar donde se encuentra alguna cosa(pasajero, puerta de embarque, tiendas, servicio)
Node: Concepto de coche
Tienda: Establecimiento donde se vende algún producto.
Promoción: Publicidad que se da de un determinado producto.

DMC



Descripción modelo conceptual no relacional.

Al ser un modelo conceptual no relacional, las líneas que hay entre dos conceptos no indican la relación que hay entre ellos sino como se se desglosa, las flechas con la punta negra indican obligación y las blancas indican que se puede desglosar o no.
Por ejemplo el concepto pasajero se desglosa en tres conceptos o entidades, pasajero se compone de sus propios atributos, debe tener un usuario, debe tener una ubicación y puede tener un vuelo asignado.

Explicació software/hardware utilitzat

El software que utilizaremos para el almacenamiento de datos es MongoDB, mongo viene de la palabra inglesa humongus(“enorme”), es un sistema de base de datos NOSQL(No relacional).
Lo hemos escogido porque es escalable, flexible, fácil de aprender, usar y sobretodo que la información que almacenaremos será la de un aeropuerto internacional, lo que nos lleva a almacenar grandes cantidades de datos que además irá creciendo constantemente.

Descripción modelo lógico no relacional en MongoDB

En las bases de datos no relaciones, el concepto de tabla y su relación mediante claves foráneas son muy diferentes, en mongoDB no existe una estructura de relaciones, en lugar de tablas se utilizan las colecciones(collections) que están formadas con atributos igual que las tablas de las bases de datos relacionales, pero lo que hace que sean no relaciones es que están orientadas a objetos, es decir que dentro de una colección se pueden añadir objetos, es como si en una base de datos SQL se pudiesen añadir tablas dentro de otras tablas, en este apartado se explicará la estructura de cada colección de nuestra base de datos.

Colecciones

passengers

Esta sería una colección de pasajeros, con un id(su documento de identidad). un nombre, el email, su fecha de nacimiento, el país y la ciudad por si en un futuro necesitáramos generar estadísticas de los pasajeros. También podemos observar que la contraseña de nuestro pasajero está cifrada con un hash y así mantener la privacidad de su contraseña, la ubicación es la posición actual donde se encuentra.

Operators

En el caso de los operadores tendría los mismo atributos del concepto Usuario de nuestro MC, solo que en este le añadimos el atributo aerolínea que es para la compañía que trabaja.

Administrators

La colección de Administrators tiene los mismo atributos que el concepto usuario porque hemos supuesto que no habrá muchos administradores por lo tanto no hace falta que tengamos demasiada información de ellos en nuestro sistema para estadísticas o similares.

Shops

Estructura básica de las tiendas, contiene información del nombre, el producto que vende, su localización para guiar a los pasajeros hacia las tiendas cercanas, un atributo llamado type que nos indicará de qué será este comercio, que pueden ser tiendas, restaurantes, de ocio y por último una lista de las ofertas que promocionan.

Node

La colección Node tiene solo estos atributos porque aún no sabemos qué datos relevantes necesitamos.

Flights

```
{
  {
    {
      name: "VL 203",
      From: "Vilanova",
      to: "London",
      Boarding_time: "18:30",
      departure_time: "19:10",
      arrival_time: "21:30",
      seat: "FA2"
      date:  new Date("YYYY-mm-dd"),
      gate: {
        name: "F4",
        location : "546,20"
      },
      passengers:{
        id: "37882964D",
        name: "Pedro Sanchez"
      },
      {
        id: "57686924E",
        name: "Jordi Gasol"
      }
    }
```

Estructura básica de la colección vuelo que guarda información del nombre del vuelo, el origen, destino del vuelo, la hora de embarque, hora a la que llegará a su destino como la hora a la que sale del origen, también tenemos la fecha del vuelo,la información de la puerta de embarque con su nombre y localización,por último tenemos todos los pasajeros que pueden subir a ese vuelo.

Instalacion de mongoDB en Debian

Importamos la clave pública utilizada por el sistema de gestión de paquetes.

```
wget -qO - https://www.mongodb.org/static/pgp/server-4.2.asc | sudo apt-key add -
```

Creamos un archivo en /etc/apt/sources.list.d/mongodb-org-4.2.list para mongoDB

```
echo "deb http://repo.mongodb.org/apt/debian buster/mongodb-org/4.2 main" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.2.list
```

Ahora actualizamos el paquete de base de datos.

```
sudo apt-get update
```

Por último solo nos queda instalar el paquete de mongoDB

```
sudo apt-get install -y mongodb-org
```

Iniciar mongoDB

```
sudo systemctl start mongod
```

Parar mongoDB

```
sudo systemctl stop mongod
```

Usar mongoDB

```
mongo
```

Para más información el link de instalación de mongoDB en Debian

https://docs.mongodb.com/manual/tutorial/install-mongodb-on-debian/

En caso de que mongo deje de funcionar
cfchown -R mongodb:mongodb /var/lib/mongodb
chown mongodb:mongodb /tmp/mongodb-27017.sock