Implementación de CRUD utilizando Mongoose

CRUD es un acrónimo que se refiere a las siguientes cuatro funciones básicas que debe incluir todo modelo:

- Create: crea unas nuevas entradas
- Read: recupera la información de unas entradas
- Update: actualiza algunas entradas
- Delete: elimina ciertas entradas

Antes de poder implementar estas funciones se necesita saber cómo crear un esquema con mongoose. Los esquemas son objetos que definen una estructura según como se deben guardan los documentos en una colección particular en MongoDB.

Para crear un esquema seguimos la siguiente estructura:

```
const nombre_esquema = new mongoose.Schema(
  {
    campo1: tipo1,
    campo2: tipo2,
    campo3: tipo3
  }
)
```

Donde los tipos pueden ser String, Number, Date, Boolean, Array o ObjectId. Con respecto al último tipo, se utiliza cuando se hace referencia a otro documento. Usualmente, los campos con este tipo llevan están escritos de la siguiente manera:

```
    referencia: { //referencia a otro documento
      type: mongoose.Schema.Types.ObjectId, // hace referencia a otro documento
      ref: 'otra_coleccion',          // nombre de la coleccion a la cual referencia
    },
```

Aparte de definir el tipo de cada campo es posible definir algunas validaciones ya sea para asegurarse que el campo no este vacío, que sea único, etc. Al definir estas validaciones se pueden usar las siguientes estructuras:

```
const nombre_esquema = new mongoose.Schema(
  {
    campo1: {
      type: String,       // tipo de dato
      required: true,     // validacion: campo requerido
      unique: true,       // validacion: el valor debe ser unico
      trim: true,         // elimina espacios en blanco
      minlength: 3,       // longitud minima (min para numeros)
      maxlength: 50,      // longitud maxima (max para numeros)
      default: Hello  ,    //valor por defecto
    },
  },
);
```

Una vez definidos los esquemas se puede pasar a definir los modelos basados en sus esquemas correspondientes. Un modelo nos permite interactuar con los documentos y ejecutar las operaciones CRUD. Para crear un modelo se usa la siguiente estructura:

```
const modelo = mongoose.model('nombre_modelo', esquema, 'coleccion');
```

Con el modelo creado podemos pasar a definir sus operaciones CRUD respectivas.

Create

Empezando por la operación Create (Crear) se utiliza la siguiente estructura:

```
const m = new modelo({campo1: dato1, campo2: dato2, ..., campoN: datoN})
const resultado = await m.save()
```

La primera línea crea una nueva instancia de un modelo particular y provee los datos del documento. Luego la segunda línea utiliza la función save() para validar el documento según lo establecido en el esquema del cual esta basado. Si la función se termina de ejecutar exitosamente un nuevo documento es añadido a la base de datos.

Read

```
const m = await modelo.find()
```

La línea anterior es el método Read (Leer) para toda una colección asociada con un esquema. El método find() recopila todos los documentos en la colección y los coloca en el arreglo personas. Si se busca solo recopilar documentos que cumplen alguna condición se utiliza lo siguiente:

```
const m = await modelo.find({campo1: dato, campo2: {$operador: valorlimite}})
```

También se puede especificar que solo se obtengan algunos campos en lugar de todos:

```
const m = await modelo.find({}, {campo1: 1, campo2: 1, campo3: 0 });
```

Update

Al momento de actualizar (update) un documento se necesita saber la cantidad y que criterio deben cumplir los documentos que serán editados. Cuando se van a actualizar varios documentos se utiliza el método updateMany() y updateOne() para casos donde solo se busca actualizar un documento. El primer atributo que tienen ambas de estas funciones es el atributo mediante el cual se seleccionaran los documentos seguido por los campos que se cambiaran junto con su valor nuevo. Un ejemplo de la estructura que se sigue para editar documentos es la siguiente:

```
const m = await model.updateOne({campoidentificador: valor},
{
  $set: {
    campo1: nuevo1,
    campo2: nuevo2,
    ...,
    campoN: nuevoN
  }
})
```

Donde valor es el valor especifico que tiene el documento a editar en su campo identificador.

Delete

Al igual que la operación Update, la operación Delete (Eliminar) cuenta con dos métodos diseñados para eliminar varios documentos o un único documento, estos son deleteMany() y deleteOne() respectivamente. Además, también requiere especificar que criterio deben cumplir los elementos que serán eliminados.

```
const m = await modelo.deleteOne({campoidentificador: valor})
```

Ahora con todo lo discutido anteriormente se puede implementar las operaciones CRUD utilizando Mongoose en Express.js y podemos pasar a codificar middleware donde se utilizan estas operaciones.