

Alumno: Agustín Mazza.

Criterios de aceptación:

- Aplicar aquellas metodologías mostradas durante las clases.
- Argumentar cada una de las decisiones a la hora de definir los modelos.
- Diagramar de forma legible con cada uno de los gráficos.
- Entregar el trabajo en formato PDF, con texto justificado, nombre y apellido del alumno en cuestión.

Actividad:

Contexto:

- Un comercio de suministros de limpieza requiere que cada uno de sus empleados pueda registrar sus ventas de forma individual. Además, deben tener acceso a la aplicación utilizando credenciales.
- El propietario del comercio necesita calcular el promedio de ventas de cada empleado mensualmente, ordenarlo de mayor a menor, y basándose en este ranking, seleccionar al empleado destacado del mes.
Basándose en el contexto dado, deberá:

- 1) Realizar un diagrama que represente cada una de las entidades, cuáles son las propiedades que tendrán cada una de ellas y cómo estará relacionada la información.
- 2) Realizar otro diagrama que permita apreciar de forma gráfica qué tipo de “relaciones” tendrán cada una de las colecciones con respecto a sus entidades, si serán embebidas o asociadas y deberán justificar su elección.
- 3) Crear los esquemas y modelos con Mongoose respetando las elecciones que hayan sido tomadas en los diagramas previos.

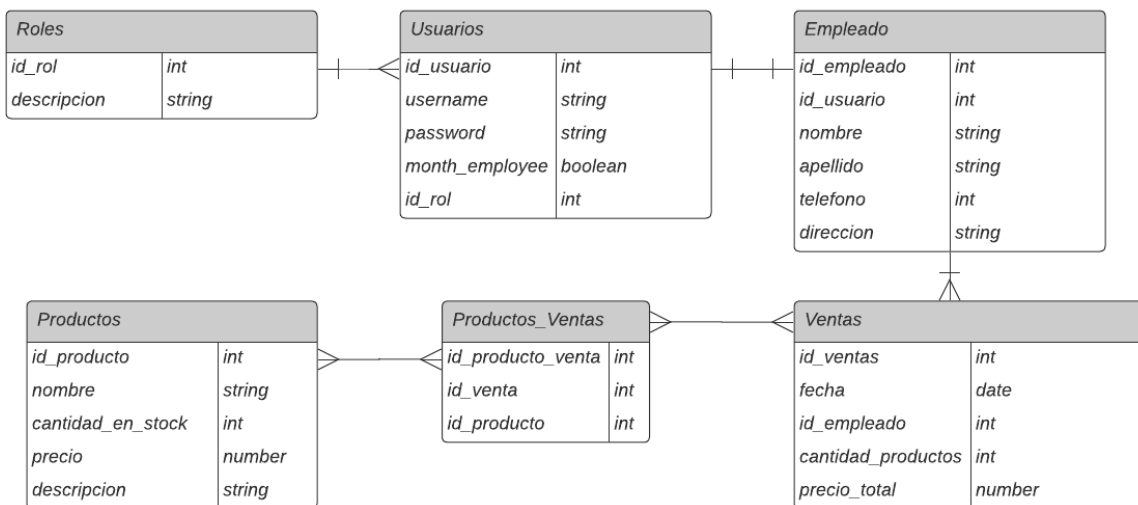
Desarrollo:

1) Entidades:

- Empleados
- Usuarios
- Roles
- Productos
- Ventas
- Productos-Ventas

Ejercicio de Modelado con Mongoose.

Agustin Mazza | September 21, 2023



Explicación:

- **Roles:** Habrán 2 roles, administrador y vendedor. Todos los empleados tendrán el rol de vendedor, el único con el rol de “administrador” será el propietario del comercio.

#Cardinalidad: Uno a muchos. Un rol puede pertenecer a muchos empleados y muchos empleados pueden tener un rol.

- **Usuarios:** Las credenciales para acceder a la aplicación serán el username y la password, estará relacionada con las entidades “Roles” y “Empleados”. “month_employee” será true o false dependiendo de un dato calculado, la cantidad de dinero generado por sus ventas, es decir, se sumarán todos los “precio_total” de la entidad “Ventas” correspondientes a cada empleado y el que posea un valor más alto será el empleado del mes.

#Cardinalidad: uno a uno. Un empleado puede tener un usuario y un usuario puede corresponder a solo un empleado.

- **Ventas:** Se registraron las ventas realizadas por los empleados, la fecha en la que se realizó, la cantidad de productos en dicha venta y el costo total de todos los productos de la venta correspondiente.

#Cardinalidad: uno a muchos. Muchas ventas pueden corresponder a un empleado y un empleado puede tener muchas ventas.

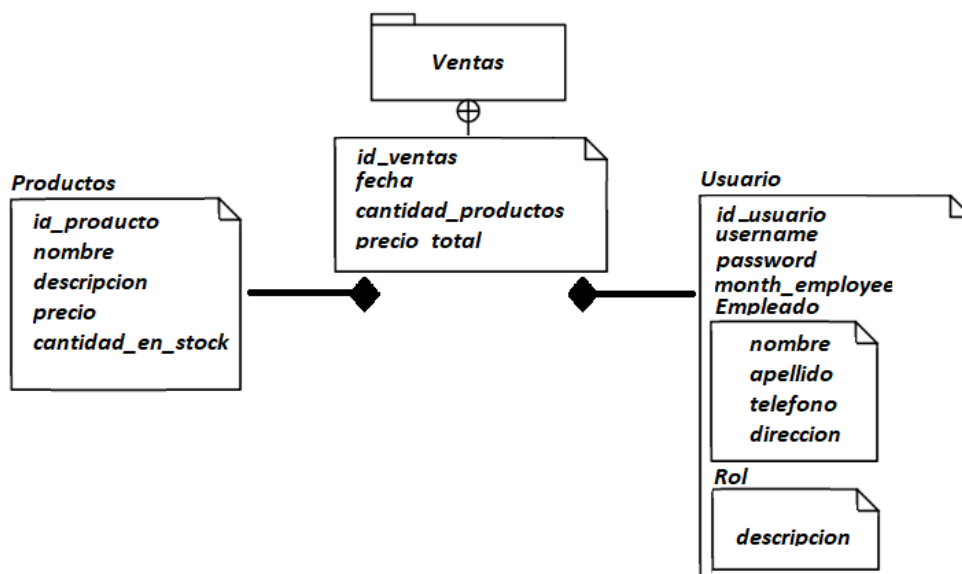
- **Productos:** Se registrará el nombre de los productos de limpieza, su descripción, costo y la cantidad de stock disponible del mismo.

#Cardinalidad:
Muchos a muchos. Relacionando productos con ventas en una tabla intermedia.

- **Productos-Ventas:** Tabla intermedio entre productos y ventas, se cargarán los registros de las ventas junto a los productos incluidos en la misma.

#Cardinalidad: Muchos a muchos. Muchas ventas pueden tener muchos productos. Muchos productos pueden corresponder a muchas ventas.

2)



Explicación: La colección será “Ventas”, la cual contará con 3 documentos.

- **Usuarios:** tendrá 2 documentos embebidos, “Empleado” y “Rol”, esto para tener todo lo relacionado a un usuario en un mismo documento, pudiendo acceder a toda su información de un mismo lugar.
- **Productos:** Contará con toda la información relacionada a los productos del comercio en cuestión.
- **Ventas:** Tendrá la información relacionada a una venta (fecha, precio total y cantidad de productos) y se “relaciona” por asociación a los productos y usuarios involucrados en una venta, es decir, referenciando al usuario(empleador) que realizó la venta y a los productos incluidos en la misma.

Esto para no cargar todos los productos ni a todos los usuarios en un mismo documento, con el objetivo de evitar una falta de consistencia en los datos.

3)

- **MODELO DE USUARIO**

```
JS user.model.js M X
models > JS user.model.js > userSchema > empleado
1 import mongoose, { Schema, model } from "mongoose";
2
3 const userSchema = new Schema({
4   username: {
5     type: String,
6     required: true,
7   },
8   password: {
9     type: String,
10    required: true,
11  },
12  month_employee: {
13    type: Boolean,
14    required: true,
15  },
16  empleado: {
17    type: Schema.Types.ObjectId,
18    ref: 'Empleado',
19    required: true,
20  },
21  rol: {
22    type: Schema.Types.ObjectId,
23    ref: "Rol",
24    required: true
25  },
26 }, {
27   timestamps: true
28 });
29
30 const Usuario = model('Usuario', userSchema);
31
32 export default Usuario;
```

- **MODELO DE ROL**

```
JS rol.model.js U X
models > JS rol.model.js > [⌘] default
1  import mongoose, { Schema, model } from 'mongoose';
2
3  const roleSchema = new Schema({
4    descripcion: {
5      type: String,
6      required: true,
7    },
8  });
9
10 const Rol = model('Rol', roleSchema);
11
12 export default Rol;
13
```

- **MODELO DE EMPLEADO**

```
JS employee.model.js U X
models > JS employee.model.js > [⌘] empleadosSchema > 🔗 telefono
1  import mongoose, { Schema, model } from "mongoose";
2
3  const empleadosSchema = new Schema({
4    nombre: {
5      type: String,
6      required: true,
7    },
8    apellido: {
9      type: String,
10     required: true,
11   },
12   telefono: {
13     type: Number,
14     required: true,
15   },
16   direccion: {
17     type: String,
18     required: true,
19   },
20 });
21
22 const Empleado = model('Empleado', empleadosSchema);
23
24 export default Empleado
25
```

- **MODELO DE PRODUCTO**

```
product.model.js M X
models > JS product.model.js > ...
1  import mongoose, { Schema, model } from 'mongoose';
2
3  const productosSchema = new Schema({
4    nombre: {
5      type: String,
6      required: true,
7    },
8    cantidad_en_stock: {
9      type: Number,
10     required: true,
11   },
12   precio: {
13     type: Number,
14     required: true,
15   },
16   descripcion: {
17     type: String,
18     required: true,
19   },
20 });
21
22 const Producto = model('Producto', productosSchema);
23
24 export default Producto;
25
```

- **MODELO DE VENTA**

```
sale.model.js M X
models > JS sale.model.js > ventasSchema
1  import mongoose, { Schema, model } from 'mongoose';
2
3  const ventasSchema = new Schema({
4    fecha: {
5      type: Date,
6      required: true,
7    },
8    cantidad_productos: {
9      type: Number,
10     required: true,
11   },
12   precio_total: {
13     type: Number,
14     required: true,
15   },
16   usuario: {
17     type: Schema.Types.ObjectId,
18     ref: 'Usuario',
19     required: true,
20   },
21   productos: [{
22     type: Schema.Types.ObjectId,
23     ref: 'Producto',
24     required: true,
25   }],
26 });
27
28 const Venta = model('Venta', ventasSchema);
29
30 export default Venta;
31
```