

**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERÍA**

**Escuela Profesional de Ingeniería de Sistemas**

**BD NO-RELACIONAL 03 - MONGODB**

Curso: Base de datos II

Docente: Ing. Juan Miguel Choque Flores

Integrantes:

***Ayma Choque, Erick Yoel (2021072616)***

***Poma Machicado, Fabiola Estefani (2021070030)***

***Tapia Vargas, Dylan Yariet (2021072630)***

**Tacna – Perú**

***2024***

**Desarrollo de una plataforma de comercio electrónico aplicando inteligencia artificial para aumentar las ventas de celulares en la empresa MijoStore**

**Diccionario de Datos**

**Versión *{1.0}***

**ÍNDICE GENERAL**

[**1.**](#_heading=h.gjdgxs) **Modelo Entidad / relación** 4

[**1.1.**](#_heading=h.30j0zll) **Diseño lógico** 4

[**1.2.**](#_heading=h.1fob9te) **Diseño Físico** 4

[**2.**](#_heading=h.3znysh7) **DICCIONARIO DE DATOS** 4

[**2.1.**](#_heading=h.2et92p0) **Tablas** 4

[**1.1.**](#_heading=h.tyjcwt) **Procedimientos Almacenados** 4

[**1.2.**](#_heading=h.3dy6vkm) **Lenguaje de Definición de Datos (DDL)** 5

[**1.3.**](#_heading=h.1t3h5sf) **Lenguaje de Manipulación de Datos (DML)** 5

**Diccionario de Datos**

1. **Modelo Entidad / relación**
   1. **Diseño lógico**

El diseño lógico en MongoDB se centra en cómo estructurar los datos, las colecciones y las relaciones entre los documentos. Incluye:

**Estructura separada por colecciones**

**1.1.1 Colección usuarios**

**{**

**"\_id": { "$oid": "64a29f4bc5e4b4f34b89a001" },**

**"nombre": "Juan",**

**"apellido": "Pérez",**

**"correo": "juan.perez@example.com",**

**"clave": "hashed\_password",**

**"rol": "admin",**

**"imagen": "url\_de\_imagen"**

**}**

**1.1.2 Colección categorias**

**{**

**"\_id": { "$oid": "64a29f4bc5e4b4f34b89a002" },**

**"nombre": "Electrónica"**

**}**

**1.1.3 Colección productos**

**{**

**"\_id": { "$oid": "64a29f4bc5e4b4f34b89a003" },**

**"categoria\_id": { "$oid": "64a29f4bc5e4b4f34b89a002" },**

**"nombre": "Smartphone XYZ",**

**"descripcion": "Un smartphone avanzado",**

**"precio": 999.99,**

**"stock": 100,**

**"descuento": "10",**

**"fecha": "2024-12-01",**

**"imagen": "url\_imagen\_producto",**

**"color": "Negro",**

**"caracteristicas": {**

**"sistema\_operativo": "Android",**

**"ram": "8GB",**

**"camara": "12MP",**

**"bateria": "4500mAh",**

**"almacenamiento": "128GB",**

**"pantalla": "6.5 pulgadas",**

**"procesador": "Snapdragon 888",**

**"carga\_rapida": "Sí"**

**}**

**}**

**1.1.4 Colección pedidos**

**{**

**"\_id": { "$oid": "64a29f4bc5e4b4f34b89a004" },**

**"usuario\_id": { "$oid": "64a29f4bc5e4b4f34b89a001" },**

**"direccion": {**

**"departamento": "Lima",**

**"provincia": "Lima",**

**"direccion\_detalle": "Av. Principal 123"**

**},**

**"precio\_total": 1999.99,**

**"estatus": "pendiente",**

**"fecha": "2024-12-01",**

**"hora": "14:30",**

**"productos": [**

**{**

**"producto\_id": { "$oid": "64a29f4bc5e4b4f34b89a003" },**

**"nombre": "Smartphone XYZ",**

**"unidades": 2,**

**"precio\_unitario": 999.99**

**}**

**]**

**}**

**1.1.5 Colección promociones**

**{**

**"\_id": { "$oid": "64a29f4bc5e4b4f34b89a005" },**

**"producto\_id": { "$oid": "64a29f4bc5e4b4f34b89a003" },**

**"fecha\_inicio": "2024-12-01",**

**"fecha\_fin": "2024-12-31",**

**"descuento\_porcentaje": 15.00,**

**"activo": true,**

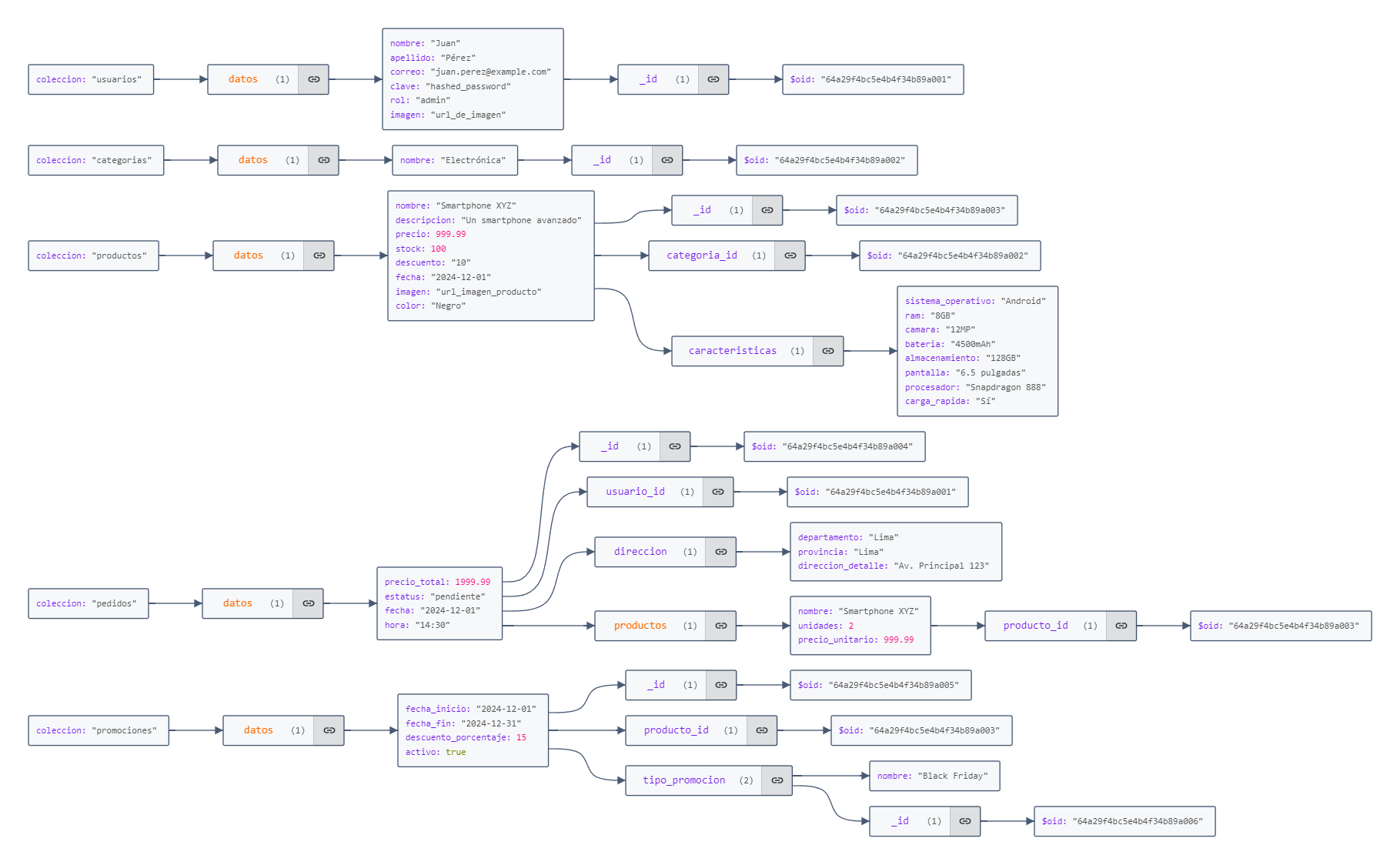
**"tipo\_promocion": {**

**"\_id": { "$oid": "64a29f4bc5e4b4f34b89a006" },**

**"nombre": "Black Friday"**

**}**

**}**



* 1. **Diseño Físico**

El diseño físico en MongoDB se centra en cómo los datos se almacenan y se accede a ellos. Aquí es donde optimizas para consultas, almacenamiento y escalabilidad.

**1.2.1 Creación de las colecciones**Utiliza el comando db.createCollection para cada entidad.

**1.2.2 Configuración de índices**

* Índices en campos de búsqueda frecuente, como correo en usuarios y nombre en productos.

db.usuarios.createIndex({ correo: 1 }, { unique: true });

db.productos.createIndex({ nombre: 1 });

**1.2.3 Insertar datos iniciales**Usa db.[nombreColeccion].insertMany([...]).

**1.2.4 Considerar esquema flexible o validaciones**Puedes definir validaciones opcionales para garantizar la estructura.

db.createCollection("usuarios", {

validator: {

$jsonSchema: {

bsonType: "object",

required: ["nombre", "correo", "clave"],

properties: {

nombre: { bsonType: "string" },

correo: { bsonType: "string", pattern: ".+@.+\\..+" },

clave: { bsonType: "string" }

}

}

}

});

1. **DICCIONARIO DE DATOS**
   1. **Colecciones**

| **Nombre de la Colección:** | | | **usuarios** | |
| --- | --- | --- | --- | --- |
| **Descripción de la colecciones:** | | | Almacena información sobre los usuarios registrados en el sistema, incluyendo datos personales, credenciales y rol. Este es el punto de entrada para la gestión de usuarios en el sistema. | |
| **Objetivo:** | | | Guardar los detalles de los usuarios para autenticar, autorizar y gestionar sus permisos en el sistema. | |
| **Relaciones con otras colecciones:** | | | **Colección pedidos**: Cada usuario puede tener varios pedidos asociados, referenciados a través del campo usuario\_id. | |
| **Descripción de los campos** | | | | |
| **Nro.** | **Nombre del campo** | **Tipo dato longitud** | **Permite nulos** | **Descripción del campo** |
|
| 1 | \_id | ObjectId | no | Identificador único del usuario (generado automáticamente por MongoDB) |
| 2 | nombre | string (100) | no | Nombre del usuario |
| 3 | apellido | string (100) | no | Apellido del usuario |
| 4 | correo | string (100) | no | Correo electrónico del usuario |
| 5 | clave | string | no | Contraseña cifrada del usuario |
| 6 | rol | string (50) | no | Rol del usuario (ej. admin, cliente) |
| 7 | imagen | string | si | URL de la imagen de perfil del usuario |

| **Nombre de la Colección:** | | | **categorias** | |
| --- | --- | --- | --- | --- |
| **Descripción de la colecciones:** | | | Contiene categorías de productos en el sistema, cada una con un nombre que permite clasificar los productos por marcas. | |
| **Objetivo:** | | | Almacenar las categorías bajo las cuales se organizan los productos. | |
| **Relaciones con otras colecciones:** | | | **Colección productos**: Los productos están categorizados por el campo categoria\_id, que hace referencia a esta colección. | |
| **Descripción de los campos** | | | | |
| **Nro.** | **Nombre del campo** | **Tipo dato longitud** | **Permite nulos** | **Descripción del campo** |
|
| 1 | \_id | ObjectId | no | Identificador único de la categoría (generado automáticamente por MongoDB) |
| 2 | nombre | string (100) | no | Nombre de la categoría (ej. Samsung, Apple |

| **Nombre de la Colección:** | | | **productos** | |
| --- | --- | --- | --- | --- |
| **Descripción de la colecciones:** | | | Contiene los productos disponibles en el sistema, con atributos como nombre, descripción, precio, stock, etc. Los productos están relacionados con categorías específicas. | |
| **Objetivo:** | | | Almacenar los detalles de los productos, incluyendo sus características y precios. | |
| **Relaciones con otras colecciones:** | | | **Colección categorias**: La relación se establece mediante el campo categoria\_id, que hace referencia a la categoría del producto. | |
| **Descripción de los campos** | | | | |
| **Nro.** | **Nombre del campo** | **Tipo dato longitud** | **Permite nulos** | **Descripción del campo** |
|
| 1 | \_id | ObjectId | no | Identificador único del producto (generado automáticamente por MongoDB) |
| 2 | categoria\_id | ObjectId | no | Identificador de la categoría del producto |
| 3 | nombre | string (100) | no | Nombre del producto |
| 4 | descripcion | string (255) | si | Descripción breve del producto |
| 5 | precio | float | no | Precio unitario del producto |
| 6 | stock | int | no | Número de unidades disponibles del producto |
| 7 | descuento | string (50) | si | Porcentaje de descuento aplicable |
| 8 | fecha | date | no | Fecha de creación del producto |
| 9 | imagen | string | si | URL de la imagen del producto |
| 10 | color | string (50) | si | Color del producto |
| 11 | caracteristicas | object | si | Características técnicas del producto (ej. sistema operativo, RAM, cámara, etc.) |

| **Nombre de la Colección:** | | | **pedidos** | |
| --- | --- | --- | --- | --- |
| **Descripción de la colecciones:** | | | Almacena los pedidos realizados por los usuarios. Cada pedido tiene detalles sobre los productos adquiridos, la dirección de entrega, el precio total y su estatus. | |
| **Objetivo:** | | | Gestionar y almacenar los pedidos realizados por los usuarios. | |
| **Relaciones con otras colecciones:** | | | **Colección usuarios**: Se hace referencia al usuario que realizó el pedido mediante el campo usuario\_id.  **Colección productos**: Los productos en el pedido se encuentran referenciados a través del campo productos.producto\_id. | |
| **Descripción de los campos** | | | | |
| **Nro.** | **Nombre del campo** | **Tipo dato longitud** | **Permite nulos** | **Descripción del campo** |
|
| 1 | \_id | ObjectId | no | Identificador único del pedido (generado automáticamente por MongoDB) |
| 2 | usuario\_id | ObjectId | no | Identificador del usuario que realizó el pedido |
| 3 | direccion | object | no | Dirección de entrega del pedido (departamento, provincia, detalle) |
| 4 | precio\_total | float | no | Precio total del pedido |
| 5 | estatus | string (50) | no | Estatus del pedido (pendiente, enviado, entregado) |
| 6 | fecha | date | no | Fecha del pedido |
| 7 | hora | string (5) | no | Hora del pedido (ej. 14:30) |
| 8 | productos | array | no | Lista de productos comprados, con su cantidad y precio unitario |

| **Nombre de la Colección:** | | | **promociones** | |
| --- | --- | --- | --- | --- |
| **Descripción de la colecciones:** | | | Almacena las promociones activas en el sistema, asociadas a productos específicos, con detalles como el porcentaje de descuento, fechas de inicio y fin, y tipo de promoción. | |
| **Objetivo:** | | | Gestionar las promociones de productos, como descuentos y ofertas especiales. | |
| **Relaciones con otras colecciones:** | | | **Colección productos**: Se refiere al producto que tiene la promoción activa mediante el campo producto\_id.  **Colección tipos\_promocion**: Los tipos de promoción se definen en una colección separada que se hace referencia mediante el campo tipo\_promocion. | |
| **Descripción de los campos** | | | | |
| **Nro.** | **Nombre del campo** | **Tipo dato longitud** | **Permite nulos** | **Descripción del campo** |
|
| 1 | \_id | ObjectId | no | Identificador único de la promoción (generado automáticamente por MongoDB) |
| 2 | producto\_id | ObjectId | no | Identificador del producto al que se le aplica la promoción |
| 3 | fecha\_inicio | date | no | Fecha de inicio de la promoción |
| 4 | fecha\_fin | date | no | Fecha de fin de la promoción |
| 5 | descuento\_porcentaje | float | no | Porcentaje de descuento aplicado |
| 6 | activo | boolean | no | Indica si la promoción está activa |
| 7 | tipo\_promocion | object | no | Tipo de promoción (ej. Black Friday) |

* 1. **Métodos**

| **Nombre de método:** | | | guardarMongoDB | | |
| --- | --- | --- | --- | --- | --- |
| **Descripción del método:** | | | Guardar un usuario en la base de datos MongoDB. | | |
| **Objetivo:** | | | Insertar un nuevo documento en la colección usuarios de la base de datos MiTiendaDB. | | |
| **Relaciones con otros método:** | | | Ninguno | | |
| **Descripción de los campos** | | | | | |
| **Nro.** | **Nombre del campo** | **Tipo dato** | **IN** | **OUT** | **Descripción del campo** |
| 1 | nombre | String | Si |  | Nombre del usuario |
| 2 | apellido | String | Si |  | Apellido del usuario |
| 3 | correo | String | Si |  | Correo electrónico del usuario |
| 4 | clave | String | Si |  | Contraseña del usuario |
| 5 | rol | String | Si |  | Rol del usuario (usuario por defecto) |
| 6 | fecha\_registro | Date | Si |  | Fecha de registro, automática |

| **Nombre de método:** | | | leerMongoDB | | |
| --- | --- | --- | --- | --- | --- |
| **Descripción del método:** | | | Leer un usuario de la base de datos MongoDB por su correo. | | |
| **Objetivo:** | | | Consultar un documento en la colección usuarios de la base de datos MiTiendaDB. | | |
| **Relaciones con otros método:** | | | Ninguno | | |
| **Descripción de los campos** | | | | | |
| **Nro.** | **Nombre del campo** | **Tipo dato** | **IN** | **OUT** | **Descripción del campo** |
| 1 | correo | String | Si | No | Correo electrónico del usuario a consultar. |
| 2 | resultado | Document | No | Si | Documento encontrado o null si no existe. |

| **Nombre de método:** | | | actualizarMongoD | | |
| --- | --- | --- | --- | --- | --- |
| **Descripción del método:** | | | Actualizar la información de un usuario en MongoDB. | | |
| **Objetivo:** | | | Modificar campos específicos del usuario en la colección usuarios. | | |
| **Relaciones con otros método:** | | | Ninguno | | |
| **Descripción de los campos** | | | | | |
| **Nro.** | **Nombre del campo** | **Tipo dato** | **IN** | **OUT** | **Descripción del campo** |
| 1 | nombre | String | Si | No | Correo electrónico del usuario a actualizar. |
| 2 | apellido | String | No | No | Nombre del usuario, si se actualiza. |
| 3 | correo | String | No | No | Apellido del usuario, si se actualiza. |
| 4 | clave | String | No | No | Nueva contraseña, si se actualiza. |
| 5 | rol | String | No | No | Nuevo rol del usuario, si se actualiza. |
| 6 | resultado | Boolean | No | Si | true si se actualizó exitosamente, false si no. |

| **Nombre de método:** | | | eliminarMongoDB | | |
| --- | --- | --- | --- | --- | --- |
| **Descripción del método:** | | | Eliminar un usuario en MongoDB por su correo. | | |
| **Objetivo:** | | | Borrar un documento de la colección usuarios en la base de datos MiTiendaDB. | | |
| **Relaciones con otros método:** | | | Ninguno | | |
| **Descripción de los campos** | | | | | |
| **Nro.** | **Nombre del campo** | **Tipo dato** | **IN** | **OUT** | **Descripción del campo** |
| 1 | correo | String | Si | No | Correo electrónico del usuario a eliminar. |
| 2 | resultado | Boolean | No | Si | true si se eliminó exitosamente, false si no. |

* 1. **Estructura y Datos Iniciales en MongoDB**

**Estructura combinada en un solo archivo**

**[**

**{**

**"coleccion": "usuarios",**

**"datos": [**

**{**

**"\_id": { "$oid": "64a29f4bc5e4b4f34b89a001" },**

**"nombre": "Juan",**

**"apellido": "Pérez",**

**"correo": "juan.perez@example.com",**

**"clave": "hashed\_password",**

**"rol": "admin",**

**"imagen": "url\_de\_imagen"**

**}**

**]**

**},**

**{**

**"coleccion": "categorias",**

**"datos": [**

**{**

**"\_id": { "$oid": "64a29f4bc5e4b4f34b89a002" },**

**"nombre": "Electrónica"**

**}**

**]**

**},**

**{**

**"coleccion": "productos",**

**"datos": [**

**{**

**"\_id": { "$oid": "64a29f4bc5e4b4f34b89a003" },**

**"categoria\_id": { "$oid": "64a29f4bc5e4b4f34b89a002" },**

**"nombre": "Smartphone XYZ",**

**"descripcion": "Un smartphone avanzado",**

**"precio": 999.99,**

**"stock": 100,**

**"descuento": "10",**

**"fecha": "2024-12-01",**

**"imagen": "url\_imagen\_producto",**

**"color": "Negro",**

**"caracteristicas": {**

**"sistema\_operativo": "Android",**

**"ram": "8GB",**

**"camara": "12MP",**

**"bateria": "4500mAh",**

**"almacenamiento": "128GB",**

**"pantalla": "6.5 pulgadas",**

**"procesador": "Snapdragon 888",**

**"carga\_rapida": "Sí"**

**}**

**}**

**]**

**},**

**{**

**"coleccion": "pedidos",**

**"datos": [**

**{**

**"\_id": { "$oid": "64a29f4bc5e4b4f34b89a004" },**

**"usuario\_id": { "$oid": "64a29f4bc5e4b4f34b89a001" },**

**"direccion": {**

**"departamento": "Lima",**

**"provincia": "Lima",**

**"direccion\_detalle": "Av. Principal 123"**

**},**

**"precio\_total": 1999.99,**

**"estatus": "pendiente",**

**"fecha": "2024-12-01",**

**"hora": "14:30",**

**"productos": [**

**{**

**"producto\_id": { "$oid": "64a29f4bc5e4b4f34b89a003" },**

**"nombre": "Smartphone XYZ",**

**"unidades": 2,**

**"precio\_unitario": 999.99**

**}**

**]**

**}**

**]**

**},**

**{**

**"coleccion": "promociones",**

**"datos": [**

**{**

**"\_id": { "$oid": "64a29f4bc5e4b4f34b89a005" },**

**"producto\_id": { "$oid": "64a29f4bc5e4b4f34b89a003" },**

**"fecha\_inicio": "2024-12-01",**

**"fecha\_fin": "2024-12-31",**

**"descuento\_porcentaje": 15.00,**

**"activo": true,**

**"tipo\_promocion": {**

**"\_id": { "$oid": "64a29f4bc5e4b4f34b89a006" },**

**"nombre": "Black Friday"**

**}**

**}**

**]**

**}**

**]**

* 1. **Manipulación de Datos en Colecciones NoSQL**

**Método: GuardarMongoDB**

public function guardarMongoDB() {

// Conectar a MongoDB

$client = new Client("mongodb://localhost:27017");

$collection = $client->MiTiendaDB->usuarios; // Asume que la base de datos es MiTiendaDB y la colección es 'usuarios'

// Encriptar la contraseña antes de guardarla

$clave\_hash = password\_hash($this->getClave(), PASSWORD\_BCRYPT);

// Crear un documento para MongoDB

$documento = [

'nombre' => $this->getNombre(),

'apellido' => $this->getApellido(),

'correo' => $this->getCorreo(),

'clave' => $clave\_hash,

'rol' => 'usuario',

'fecha\_registro' => new MongoDB\BSON\UTCDateTime()

];

// Insertar el documento en MongoDB

try {

$resultado = $collection->insertOne($documento);

return $resultado->getInsertedCount() > 0; // Si la inserción es exitosa, devuelve true

} catch (Exception $e) {

// En caso de error, devuelve false

return false;

}

}

**Método: leerMongoDB**

**public function leerMongoDB($correo) {**

**$client = new Client("mongodb://localhost:27017");**

**$collection = $client->MiTiendaDB->usuarios;**

**try {**

**$resultado = $collection->findOne(['correo' => $correo]);**

**return $resultado ?: null; // Devuelve el documento encontrado o null si no existe**

**} catch (Exception $e) {**

**return null;**

**}**

**}**

**Método: actualizarMongoDB**

**public function actualizarMongoDB($correo, $nuevosDatos) {**

**$client = new Client("mongodb://localhost:27017");**

**$collection = $client->MiTiendaDB->usuarios;**

**try {**

**$resultado = $collection->updateOne(**

**['correo' => $correo], // Filtro**

**['$set' => $nuevosDatos] // Datos a actualizar**

**);**

**return $resultado->getModifiedCount() > 0; // Devuelve true si se modificó algo**

**} catch (Exception $e) {**

**return false;**

**}**

**}**

**Método: eliminarMongoDB**

**public function eliminarMongoDB($correo) {**

**$client = new Client("mongodb://localhost:27017");**

**$collection = $client->MiTiendaDB->usuarios;**

**try {**

**$resultado = $collection->deleteOne(['correo' => $correo]);**

**return $resultado->getDeletedCount() > 0; // Devuelve true si se eliminó el documento**

**} catch (Exception $e) {**

**return false;**

**}**

**}**