

HW17_MongoImplementation ☆ 📁

Archivo Editar Ver Insertar Formato Herramientas Extensiones Ayuda

100% Texto nor... Arial 11 B I U A

BRAYAN SEBASTIAN CRIOLLO VEGA
ALEX DARIO CUZCO YAMASCA

Create

```
const mongoose = require('mongoose');
const { Schema } = mongoose;
const { ObjectId } = Schema.Types.ObjectId;

const categorySchema = Schema({
  name: String,
  products: [ObjectId]
});

const productSchema = Schema({
  name: String,
  price: Number,
  category: ObjectId
});

const Category = mongoose.model('Category', categorySchema);
const Product = mongoose.model('Product', productSchema);

// Create a new category
const createCategory = async (name) => {
  const category = new Category({ name });
  await category.save();
  return category;
};

// Create a new product
const createProduct = async (categoryName, productName, price) => {
  const category = await Category.findOne({ name: categoryName });
  if (!category) {
    throw new Error('Category not found');
  }
  const product = new Product({
    name: productName,
    price: price,
    category: category._id
  });
  await product.save();
  return product;
};
```

HW17_MongoImplementation ☆ 📁

Archivo Editar Ver Insertar Formato Herramientas Extensiones Ayuda

100% Texto nor... Arial 11 B I U A

Read

```
const mongoose = require('mongoose');
const { Schema } = mongoose;
const { ObjectId } = Schema.Types.ObjectId;

const categorySchema = Schema({
  name: String,
  products: [ObjectId]
});

const productSchema = Schema({
  name: String,
  price: Number,
  category: ObjectId
});

const Category = mongoose.model('Category', categorySchema);
const Product = mongoose.model('Product', productSchema);

// Read all categories
const readCategories = async () => {
  const categories = await Category.find().exec();
  return categories;
};

// Read a category by name
const readCategory = async (name) => {
  const category = await Category.findOne({ name }).exec();
  if (!category) {
    throw new Error('Category not found');
  }
  return category;
};

// Read all products
const readProducts = async () => {
  const products = await Product.find().exec();
  return products;
};

// Read a product by name and category
const readProduct = async (categoryName, productName) => {
  const category = await Category.findOne({ name: categoryName });
  if (!category) {
    throw new Error('Category not found');
  }
  const product = await Product.findOne({
    name: productName,
    category: category._id
  }).exec();
  if (!product) {
    throw new Error('Product not found');
  }
  return product;
};
```

HW17_MongoImplementation ☆ 📁 Guardado en Drive

Archivo Editar Ver Insertar Formato Herramientas Extensiones Ayuda

100% Texto nor... Arial 11 B I U A

Update

```
const mongoose = require('mongoose');
const { Schema } = mongoose;
const { ObjectId } = Schema.Types.ObjectId;

const categorySchema = Schema({
  name: String,
  products: [ObjectId]
});

const productSchema = Schema({
  name: String,
  price: Number,
  category: ObjectId
});

const Category = mongoose.model('Category', categorySchema);
const Product = mongoose.model('Product', productSchema);

// Update a product
const updateProduct = async (categoryName, productName, newPrice) => {
  const category = await Category.findOne({ name: categoryName });
  if (!category) {
    throw new Error('Category not found');
  }
  const product = await Product.findOne({
    name: productName,
    category: category._id
  });
  if (!product) {
    throw new Error('Product not found');
  }
  product.price = newPrice;
  await product.save();
  return product;
};
```