

Web Scraper con Puppeteer y MongoDB

Este proyecto realiza **web scraping** en **eBay** utilizando **Puppeteer** para extraer productos relacionados con "Fernando Alonso".

Los datos recolectados incluyen: **nombre**, **precio** e **imagen del producto**, y se almacenan en una **base de datos MongoDB**.

Tecnologías utilizadas

- Node.js
- Puppeteer
- MongoDB con Mongoose
- JSON para almacenamiento local

Instalación y configuración

Clona este repositorio

```
git clone https://github.com/tuusuario/web-scraper-ebay.git
cd web-scraper-ebay
```

Instala las dependencias

```
npm install
```

Configura MongoDB

Crea un archivo `.env` en la raíz del proyecto y agrega tu cadena de conexión a MongoDB:

```
DB_URI=mongodb://localhost:27017/ebay_products
```

Si usas **MongoDB Atlas**, coloca tu URL de conexión en lugar de `localhost`.

Ejecuta el scraper

```
npm run start
```

El script **buscará productos** en **eBay**, **recorrerá todas las páginas** y **guardará los datos** en **MongoDB**.

Estructura del proyecto

```
web-scraper-ebay
├── models
│   └── Product.js # Modelo de MongoDB para productos
├── controllers
│   └── pageScraper.js # Lógica de scraping con Puppeteer
├── index.js # Punto de entrada del script
└── browser.js # Configuración del navegador Puppeteer
```

```
├ db.js          # Conexión a MongoDB
├ .env           # Variables de entorno (MongoDB URI)
├ package.json   # Dependencias y configuración de NPM
└ README.md      # Documentación del proyecto
```

Personalización

Si deseas modificar la búsqueda, edita `pageScraper.js` y cambia la palabra clave:

```
await page.type("input[aria-label='Search for anything']", "Fernando Alonso")
```

Consultar datos en MongoDB

Abre **MongoDB Compass** o usa la terminal:

```
mongo
use ebay_products
db.products.find().pretty()
```
