

Webir

(Curso 2019)

TheBestChoice

Grupo 13

Laura Peralta - 4.680.843-2

Alejo Salvo - 4.660.801-8

Martin Santini - 4.917.306-2

Introducción	3
Problema y Motivación	3
Enfoque de la solución	4
Diseño e Implementación	5
Funcionalidades y Uso	10
Evaluación y resultado	12
Conclusiones	13
Trabajo a Futuro	13
Referencias	14

Introducción

Este informe toma un problema de interés compartido y real y busca aproximar una solución utilizando los conocimientos aprendidos a lo largo del curso. Se describe el proceso de construcción de una herramienta, con la finalidad de mostrar el mejor precio de un mismo dispositivo entre dos páginas de tecnología. Se podrá observar como las técnicas vistas en clase en conjunto con otras herramientas existentes son de gran utilidad en la actualidad para la resolución de problemas.

Problema y Motivación

Actualmente, la tecnología cumple un rol fundamental, su uso se extiende desde el entretenimiento hasta la educación y/o herramienta de trabajo. Cada día hay más marcas especializadas en dispositivos electrónicos y nuevos modelos aparecen.

Al momento de comprar un aparato electrónico específico (auriculares, celular, computadora, etc) resulta casi imposible conocer todo lo que existe y a veces el usuario efectúa una compra, sin conocer otras posibilidades de mayor relación calidad-precio. Esta problemática, se vuelve aún más visible cuando un mismo producto o muy similar, se encuentra a un precio menor en otro lugar al que se efectuó la compra.

Hoy en día, no existe una herramienta sencilla para que los usuarios de sitios web de tecnología decidan dónde les conviene comprar distintos dispositivos electrónicos. Probablemente, la mayoría de las compras, se realizan en el sitio más conocido, sin conocer la diferencia de precio del mismo producto en dos sitios web diferentes como se mencionó anteriormente.

Con la motivación de contribuir con una solución a esta situación, este trabajo se enfoca en la implementación de una herramienta que unifique la información de precio de celulares de distintos sitios de tecnología, desde la ingeniería y estructuración de datos a la presentación para usuarios finales.

Enfoque de la solución

Se pretende construir un sitio web, accesible tanto desde pc, tablet o celular, con un diseño sencillo y directo, que no se aleje en sus conceptos de sitios comunes conocidos de empresas tecnológicas, para que su uso sea intuitivo.

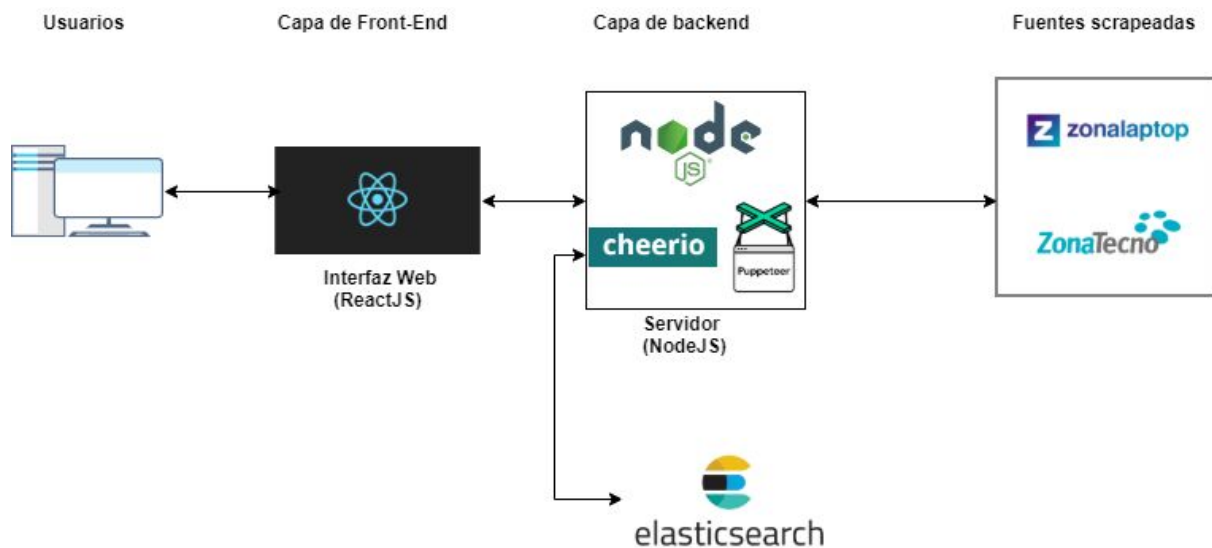
Investigando las distintas páginas y aplicaciones existentes, se definieron como fuentes de información los siguientes sitios uruguayos: Zona Laptop <https://zonalaptop.com.uy> y Zona Tecno <https://www.zonatecno.com.uy/>.

Se hizo un estudio en detalle de varios sitios, eligiendo finalmente los mencionados anteriormente, ya que, hacían más factible el resultado esperado según la estructura requerida de nuestro problema. Se observó que fuentes como Mercado Libre tienen escritura libre tanto para publicaciones como solicitudes de productos y sus características. Esto hace que los datos presentados al usuario tal vez no sean los buscados e incluso, podría no estar mostrando resultados que sí cumplan los criterios de búsqueda.

Se implementará un filtro de búsqueda primario donde el usuario deberá elegir entre las marcas conocidas con más demanda del dispositivo que busca , luego debe seleccionar la memoria que desea que tenga y un rango de precio que maneja. Acorde a estas selecciones el sistema buscará dentro de sus documentos los que cumplan estas especificaciones y mostrará el resultado del mejor precio para los distintos modelos.

Diseño e Implementación

El sistema tendrá una arquitectura cliente/servidor con los componentes que se comunican como se puede observar en el siguiente diagrama:



Representación del diagrama de componentes

La implementación del programa se realiza mediante la interacción de las diversas técnicas la recuperación de datos de las páginas web mencionadas anteriormente.

Se extrae la información necesaria del sitio web, esta información luego será utilizada para obtener cuál de los dos sitios web ofrece un menor precio por el dispositivo buscado.

Se describen a continuación los componentes y las técnicas utilizadas:

Usuarios:

Cualquier usuario podrá acceder a la aplicación desde su celular o computadora

Aplicación Web (ReactJS):

Para el front-end se usa ReactJS es un framework de javascript que ayuda a construir aplicaciones web de una manera más ordenada, sencilla, y fácil de combinar.

Servidor (NodeJS):

Para el servidor en la capa de backend decidimos utilizar NodeJS y express para implementar una API REST por la experiencia del equipo en esta tecnología y también por las herramientas que tenemos disponibles para scrapear datos.

Las librerías utilizadas para lograr scrapear datos de las fuentes mencionadas fueron:

- Puppeteer,
- Cheerio,

- Elasticsearch Client,

Elastic Search

Elasticsearch es un motor de búsqueda de texto completo utilizado para almacenar los datos recolectados de sitios web con grandes cantidades de información y hacer las consultas. Se decidió hacer uso de este dado que reúne grandes ventajas de otras herramientas similares y a su vez incluye otras características. Es un software libre desarrollado en Java y publicado como código abierto bajo las condiciones de la licencia Apache.

La comunicación con Elasticsearch se hace a través de la API REST, por lo que se necesita un cliente adicional. Usamos a su vez, Kibana, otro software de código abierto que permite utilizar Elasticsearch directamente desde el navegador.

La recuperación de los datos requiere como paso principal la creación de un índice que se construye con la siguiente estructura:

```
await esClient.index({
  index: indexName,
  type: 'apple',
  body: {
    description: descriptionText,
    link,
    image: imageLink,
    price: parseFloat(priceTextParsed),
    memory,
    model,
    page: "zonalaptop"
  }
})
```

La búsqueda se realiza comprobando este índice de documentos donde el contenido se almacenó de forma preparada. Debido a esto, este proceso requiere mucho menos tiempo que la consulta de todos los documentos. Se construye una Query de búsqueda según los filtros de los parámetros seleccionados por el usuario, con la combinación de Bool Query con Range Query siendo similar a los ejemplos a continuación:

```
const { body } = await esClient.search({
  index: indexName,
  body: {
    size: 500,
    query: {
      "bool": {
        "must": [ ],
        "filter": [
          {
```

1. Para encontrar por ejemplo la marca y memoria se arma una expresión similar a:

```
"bool": {
  "filter": [
    {
      "bool": {
        "should": [
          {
            "match": {
              "_type": brand
            }
          }
        ],
        "minimum_should_match": 1
      }
    },
    {
      "bool": {
        "filter": [
          {
            "bool": {
              "should": [
                {
                  "match": {
                    "memory": memory
                  }
                }
              ]
            }
          }
        ]
      }
    }
  ]
}
```

2. Para el rango de precio seleccionado, sería:

```
{
  "bool": {
    "should": [
      {
        "range": {
          "price": {
            "gte": lowest_price
          }
        }
      }
    ],
    "minimum_should_match": 1
  }
},
{
  "bool": {
    "filter": [
      {
        "bool": {
          "should": [
            {
              "range": {
                "price": {
                  "lte": highest_price
                }
              }
            }
          ]
        }
      }
    ]
  }
}
```

3. Dado que en las páginas bajo el mismo filtro de celulares existen otras ventas como "cases", "carcasas", "protectores", etc se debió excluir en la misma query:

```

"bool": {
  "must_not": {
    "bool": {
      "should": [
        {
          "match": {
            "description": "case"
          }
        }
      ]
    }
  }
}

```

La respuesta de Elasticsearch nos proporciona el contenido real y metadata adicional como los aciertos, hits, cada entrada que coincida con los criterios de búsqueda será considerada un éxito por Elasticsearch.

```

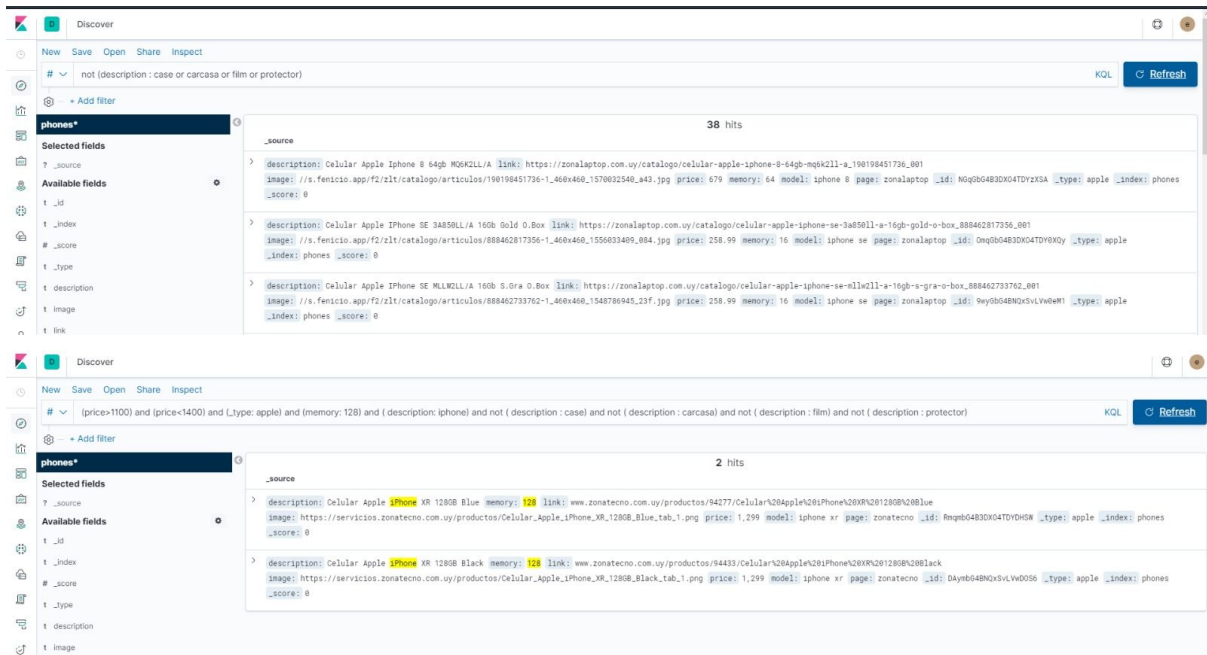
body: [
  {
    _index: 'phones',
    _type: 'apple',
    _id: 'NGGgbG4B3DXO4TDYzXSA',
    _score: 0,
    _source: {
      description: 'Celular Apple iPhone 8 64gb MQ6K2LL/A',
      link: 'https://zonalaptop.com.uy/catalogo/celular-apple-iphone-8-64gb-mq6k211-a_190198451736_001',
      image: '///s.fenicio.app/f2/zlt/catalogo/articulos/190198451736-1_460x460_1570032540_a43.jpg',
      price: 679,
      memory: '64',
      model: 'iphone 8',
      page: 'zonalaptop'
    }
  },
  {
    _index: 'phones',
    _type: 'apple',
    _id: 'QGqmbG4B3DXO4TDYDHRc',
    _score: 0,
    _source: {
      description: 'Celular Apple iPhone 8 64GB Gold',
      link: 'www.zonatecno.com.uy/productos/92498/Celular%20Apple%20iPhone%208%2064GB%20Gold',
      image: 'https://servicios.zonatecno.com.uy/productos/Celulares_Apple_iPhone_8_64GB_Gold_tab_1.png',
      price: 955,
      memory: '64',
      model: 'iphone 8',
      page: 'zonatecno'
    }
  },
  {
    _index: 'phones',
    _type: 'apple',
    _id: '_AymbG4BNQxSvLVwDONd',
    _score: 0,
    _source: {
      description: 'Celular Apple iPhone 8 64GB Silver',
      link: 'www.zonatecno.com.uy/productos/92516/Celular%20Apple%20iPhone%208%2064GB%20Silver',
      image: 'https://servicios.zonatecno.com.uy/productos/Celulares_Apple_iPhone_8_64GB_Silver_tab_1.png',
      price: 955,
      memory: '64',
      model: 'iphone 8',
      page: 'zonatecno'
    }
  }
],

```


Es entonces, una herramienta muy potente, rápida, flexible, dinámica, escalable. A su vez se puede ingresar y consultar a través de un navegador.

ElasticSearch cloud (<https://cloud.elastic.co/>):

Se utilizó al igual que Kibana para la visualización y análisis de los datos recuperados.



Fuentes Scrapeadas:

- Zona Laptop,
 - <https://zonalaptop.com.uy/>
- Zona Tecno,
 - <http://zonatecno.com.uy/>

Funcionalidades y Uso

La página web muestra al usuario la opción del aparato celular con mejor precio entre dos páginas acorde a la selección de filtros que desea consultar.

Se tienen las siguientes opciones continuas:

- Marca: Samsung o Apple
- Rango de Precios que maneja:
 - Sin rango
 - Entre USD 200 - USD 600
 - Entre USD 1000 - USD 1400
 - Entre USD 1400 en adelante
- Memoria: 32GB, 64GB, 128GB, 256GB

en una interfaz de usuario similar a la siguiente:

The image shows a dark-themed user interface for selecting smartphone filters. It is divided into three main sections:

- SELECT BRAND:** Two buttons are shown. The left button is light gray with the word "SAMSUNG" in blue. The right button is bright green with a black Apple logo.
- SELECT PRICE RANGE:** Five buttons are shown in a row: "No range" (green), "200 - 600", "600 - 1000", "1000 - 1400", and "1400 - *" (all gray).
- SELECT MEMORY SIZE:** Four buttons are shown in a row: "32" (green), "64", "128", and "256" (all gray).

At the bottom center, there is a button labeled "Get best prices!".

Al presionar el botón en el margen inferior central, la aplicación se conecta con la capa lógica del programa enviando el resultado de los parámetros de búsqueda de forma adecuada para la comunicación con el motor Elasticsearch que procesa y devuelve una lista de los mejores precios de cada uno de modelos que cumplen con las características.

← PHONES



Iphone 6s 32gb

USD 299

[Open zonalaptop](#)

[USD 419 in zonatecno](#)



Iphone 7 32gb

USD 729

[Open zonatecno](#)



Iphone 6 32gb

USD 429

[Open zonatecno](#)

Evaluación y resultado

A grandes rasgos el sitio web desarrollado cumple con los requerimientos y las funcionalidades pensadas para este proyecto. Desde el punto de vista de la velocidad y la interacción con el usuario es veloz e intuitiva.

Sin embargo, surgieron varios desafíos en el proceso de construcción del mismo, algunos resueltos adecuadamente y otros que exceden nuestro alcance, cómo se evalúa a continuación:

- Primero, comprendimos que las tecnologías elegidas en una primera instancia, no se acomodan a la naturaleza de lo que debíamos implementar. Por lo tanto, se decidió cambiar a las tecnologías anteriormente descritas.
- Para la parte de recolección de data, se observó que para comparar dos dispositivos iguales se debe normalizar las publicaciones ya que es ahí donde las páginas tienen los datos y características e incluso con distinta nomenclatura en cada página.
- El formato de las publicaciones de las páginas requirió identificar manualmente cuando se trataba de modelos (y memoria) distintos entre los dispositivos. Se utilizaron expresiones regulares para estas distinciones.
- Frente a diferencias en los resultados esperados se utilizó Kibana para explorar , visualizar y asegurarse que los datos fueron ingresados correctamente.
- Las páginas elegidas al comienzo del proyecto, resultaron con pocos dispositivos en común para comparar con el paso de los días.

A pesar de los obstáculos mencionados, se pudo comprobar la validez de los datos recabados y se verificó que corresponde al resultado correcto.

Por otro lado, se pudo valorar la enorme utilidad que tiene una herramienta desarrollada con las características del curso, así como también comprender sus limitaciones dada la gran complejidad de la data dispersa en la web.

Conclusiones

Principalmente, se pudo construir un programa correcto que resuelve el problema objetivo del trabajo, que consume información de páginas web, la almacena en elasticsearch y brinda una interfaz amigable para poder visualizarla.

Se eligió este proyecto de corto alcance para poner en práctica los conocimientos teóricos aprendidos en el curso y hacer uso de las diferentes tecnologías y técnicas de recuperación, procesamiento y presentación de información de la web

Resultó muy útil, a pesar de la curva de aprendizaje, hacer uso de herramientas como Elasticsearch que es una herramienta completa y robusta que facilitó el tanto el almacenamiento como las búsquedas de datos.

Trabajo a Futuro

Consideramos que existen bastantes mejoras que se pueden incorporar al trabajo en un futuro tanto para la eficiencia como para la calidad y utilidad del mismo.

Una de las principales y más importantes sería la incorporación de mayor cantidad de tiendas tecnológicas en la web para la recolección de datos. Creemos que sería además un buen desafío incorporar paginas como Mercado Libre o grupos de Facebook con características que presentan más complejidad.

Una segunda mejora sería incorporar un filtro libre por modelo de celular, dado que entendemos que, aunque el rango de precio es una de las principales restricciones de un usuario al momento de decidir una compra, en muchos casos el usuario podría desear un modelo específico de celular.

En tercer lugar, reconocemos muy útil, el hecho de agregar información de los feedbacks y reviews de artículos, si las páginas cuentan con eso, y permitir que el usuario pueda filtrar por estos datos. De este modo se está tomando en cuenta no solo la opinión sino la experiencia de un comprador.

Evidentemente observamos también que una proyección futura del trabajo tomaría en cuenta todas las marcas de celulares existentes en la actualidad y sin dudas, la incorporación de otros dispositivos como notebooks, tablets, auriculares y demás conformaría ya una mucho mejor solución al problema original.

Referencias

[1] Webir - Curso de recuperación de información y recomendaciones de la web:

<https://eva.fing.edu.uy/course/view.php?id=821>

[2] Framework para desarrollo en javascript: <https://es.reactjs.org/>

[3] Elasticsearch: el motor de búsqueda flexible:

<https://www.ionos.es/digitalguide/servidores/configuracion/que-es-elasticsearch/>

[4] Puppeteer: Documentación oficial: <https://pptr.dev/>

[5] Cheerio: <https://cheerio.js.org/>

[6] Kibana:

<https://www.ionos.es/digitalguide/online-marketing/analisis-web/tutorial-de-kibana/>

[7] ElasticSearch documentación oficial: <https://www.elastic.co/es/>