*Instituto Tecnológico de Costa Rica*

*Unidad de Computación*

*Introducción al Desarrollo de Aplicaciones Web*

*Investigación – Web Scraping*

*Jose Alberto Hidalgo Bonilla*

*Sede San Carlos*

*15/05/2017*

Tabla de contenido

[Resumen 2](#_Toc479942366)

[Introducción 3](#_Toc479942367)

[Desarrollo 3](#_Toc479942368)

[Conclusiones 10](#_Toc479942369)

[Recomendaciones 11](#_Toc479942370)

[Bibliografía 11](#_Toc479942371)

# Resumen

El uso de una serie de técnicas para obtener información de diferentes páginas web en lugar de manualmente copiarlas se le conoce como web scraping. El objetivo principal de un web scraper consiste en buscar cierta información necesaria y almacenarla de manera que pueda ser usada con algún propósito. Para esto existen varias técnicas en donde el usuario debe tener bien claro cuál es el objetivo del proyecto, de manera tal que el uso de alguna de estas técnicas llegue a ser la más adecuada y genere realmente los datos necesarios. Todo esto junto con el conocimiento de la información e implicaciones legales que implica el realizar web scraping y cuales conocimientos previos se deberían tener para una mayor optimización del proceso.

# Introducción

Cuando se habla de web scraping se dice que consiste en que tomar los datos más importantes de una o varias páginas web para el uso personal de estos datos o incluso hasta para su análisis posterior. Dicho concepto posee diferentes terminologías como ‘web crawling’, ‘screen scraping’, ‘web extraction’, ‘web-bot’, ‘web-spider’, entre otros.

Un web crawler o web spider generalmente rastrean páginas web utilizando algoritmos recursivos, en los cuales se escanea la primera página, se encuentran los enlaces de dicha página y se almacenan en algún tipo de estructura de datos, luego se realiza una búsqueda de los vínculos de los enlaces almacenados, se abre la página web de cada vínculo y se vuelve a almacenar en la misma estructura de datos y recursivamente se repite el proceso hasta que todos los enlaces hayan sido rastreados.

Por otro lado, el concepto de screen scraping consiste más en la extracción de la información de una página web en el momento de ser visualizada y para esto se necesitan motores como el de JavaScript y de CSS para la utilización de herramientas como headless browser.

# Desarrollo

El uso de web scraping está bien claro, con él, obtener grandes cantidades de datos (Big Data) sin tener que esforzarse mucho es muy sencillo, ya que con algoritmos de búsqueda se puede rastrear la información necesaria de miles de páginas web.

Existen varias técnicas para poder realizar web scraping, una de ellas y considerada la más fácil pero menos óptima es el uso de servicios web scraping. Mediante el uso de uno de estos servicios se puede obtener mucha información de manera estructurada de las páginas web y más aún si la página web también se encuentra bien estructurada, de esta forma la inteligencia del crawler hará que se ahorre más tiempo recolectando la información requerida, mientras que si no es así, se deberán definir las reglas de extracción de datos. Estos servicios poseen varias ventajas, una de ellas es que si alguien desea extraer datos de una página a la cual ya se le han extraído los datos anteriormente, las reglas ya estarán definidas y el trabajo a realizar será mínimo. Además tampoco existe la preocupación por el anti-scraping que implementan las páginas web ya que el servicio usado cuenta con sus propios mecanismos para evitarlo.

Por otro lado, el hecho de no poseer el control del procedimiento de extracción es una gran desventaja ya que las funcionalidades pueden estar limitadas y los datos hacerse públicos, además de que para realizar un web scraping de páginas extensas se deberá obtener algún plan de pago primero.

Algunos servicios de web scraping pueden ser Scrapinghub, 80legs, import.io, entre otros.

Otra técnica el uso de soluciones desktop para realizar web scraping localmente, desde el computador utiliziando navegadores ya se Mozilla Firefox y Google Chrome con alguna extensión de para screen scraping se pueden generar las reglas necesarias para el web crawler. Realizar web scraping localmente tiene ventajas cuando son operaciones puntuales y no se deben ejecutar automáticamente. Existen muchas aplicaciones desktop gratuitas que brindan mayor flexibilidad, mayor potencia y control sobre los datos y reglas de extracción que los servicios online, además que muchas traen una interfaz de usuario muy amigable que facilita más su uso. El problema es que se debe tener el computador encendido para realizar el proceso y suelen ser muy fácil de bloquear por parte de las páginas web.

Una técnica más para realizar web scraping es programar el algoritmo. Existen muchos lenguajes de programaciones orientados a esta técnica, como por ejemplo: Scrapy, Guzzle, GasperJS, SpookyJS, Simple HTML DOM, PHP-Spider, entre otros.

El hecho de poder realizar uno mismo el algoritmo hace que esta técnica sea la más óptima y aporte mayor flexibilidad e integración, ya que la extracción de datos se realiza con la frecuencia que se desee, se pueden gestionar los errores y adaptarse a los cambios de la página web. Teniendo los recursos necesarios el coste no tiene porqué ser mayor al de las otras técnicas, solamente que se tiene que tener cuidado con las medidas implementadas para evitar ser bloqueado por las páginas web a scrapear.

A la hora de realizar web scraping se tiene que tener mucho cuidado con las implicaciones legales. Se han tenido casos en el que un negocio por hacer uso de web sacraping ha sido denunciado por competencia desleal o temas de protección de datos.

Para esto, es importante revisar las condiciones legales de los sitios web a scrapear, ya que se podrían violar la propiedad intelectual, el uso de una marca registrada o los derechos de autor.

La legalidad de realizar web scraping sigue siedo poco clara basado en las opiniones y puntos de vista de diferentes personas, lo único que se puede concluir sobre la legalidad del caso es que si no se está haciendo ningún daño a la página web y no se venden los datos recolectados, entonces es legal el uso de estas técnicas.

Diagrama de la metodología

Para la extracción de datos se va a utilizar el lenguaje de programación python por la sencilla curva de aprendizaje y el eficiente procesamiento de múltiples tareas que tiene, además la información será almacenada en una base de datos PostgreSQL. La página web será <https://www.casas.cr/c/casas/> .

Extracción de datos

* Cada página posee una serie de casa, porque lo que se deberá scrapear todas las casa de una página e ir moviéndose entre páginas.
* En cada página se deberá obtener el link de cada casa para poder obtener su html y recolectar la información.
* El proceso se repite para cada casa y para cada página.

Parseo de datos

* Una vez recolectada la información se generará el query para insertar los datos.

Almacenamiento de datos

* Los datos recolectados serán almacenados en una base de datos PostgreSQL alojada en Heroku de manera estructurada.
* De dicha forma se podrán realizar consultas para saber los precios de las casas más baratas o cuáles han sido más vistas, entre otras.

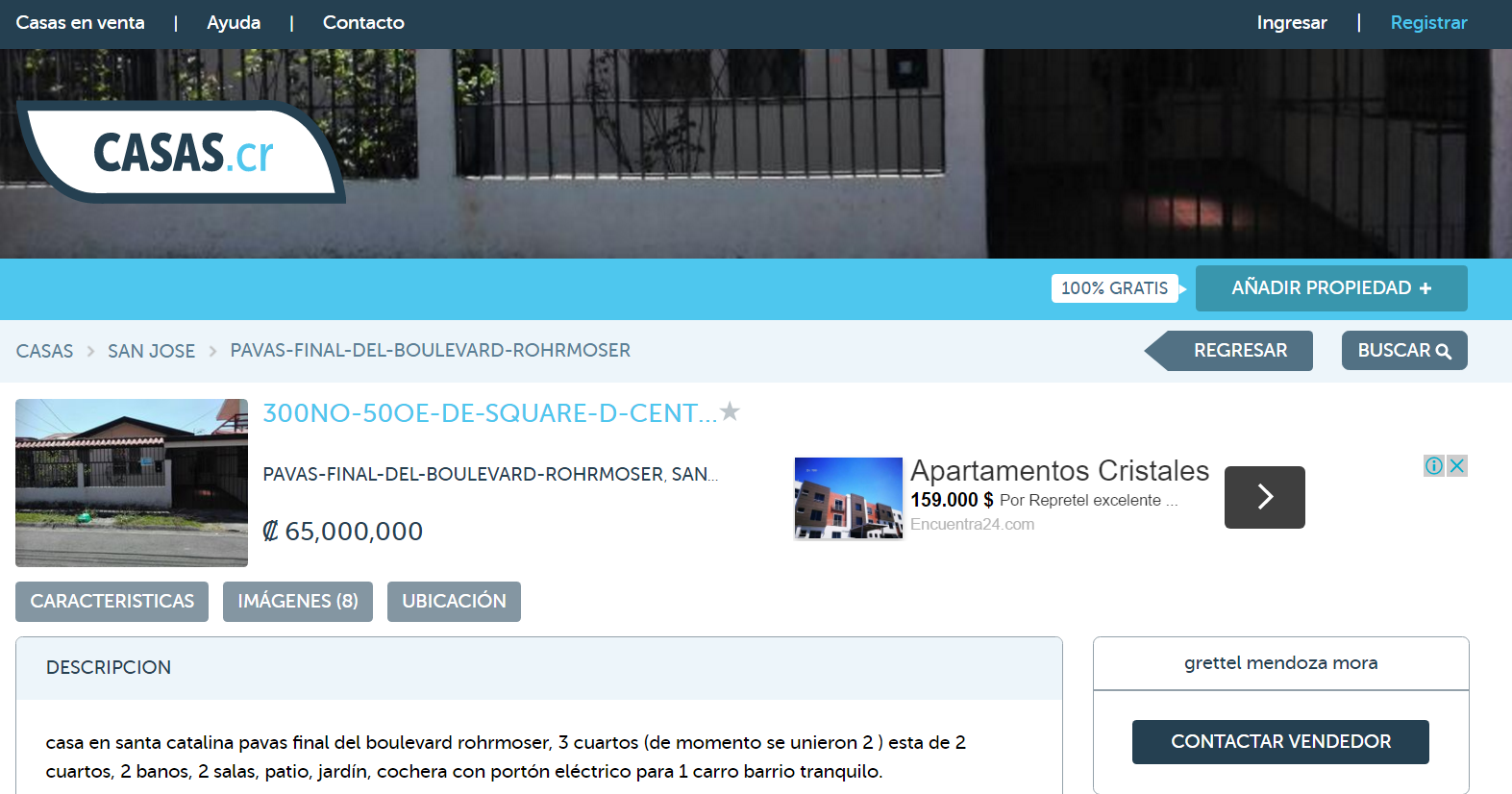
Cada vez que se termina la página de casas se seleccionará el siguiente link para recorrer la siguiente página y recolectar la información de cada casa, los links a seleccionar son los botones que se muestran a la siguiente imagen.

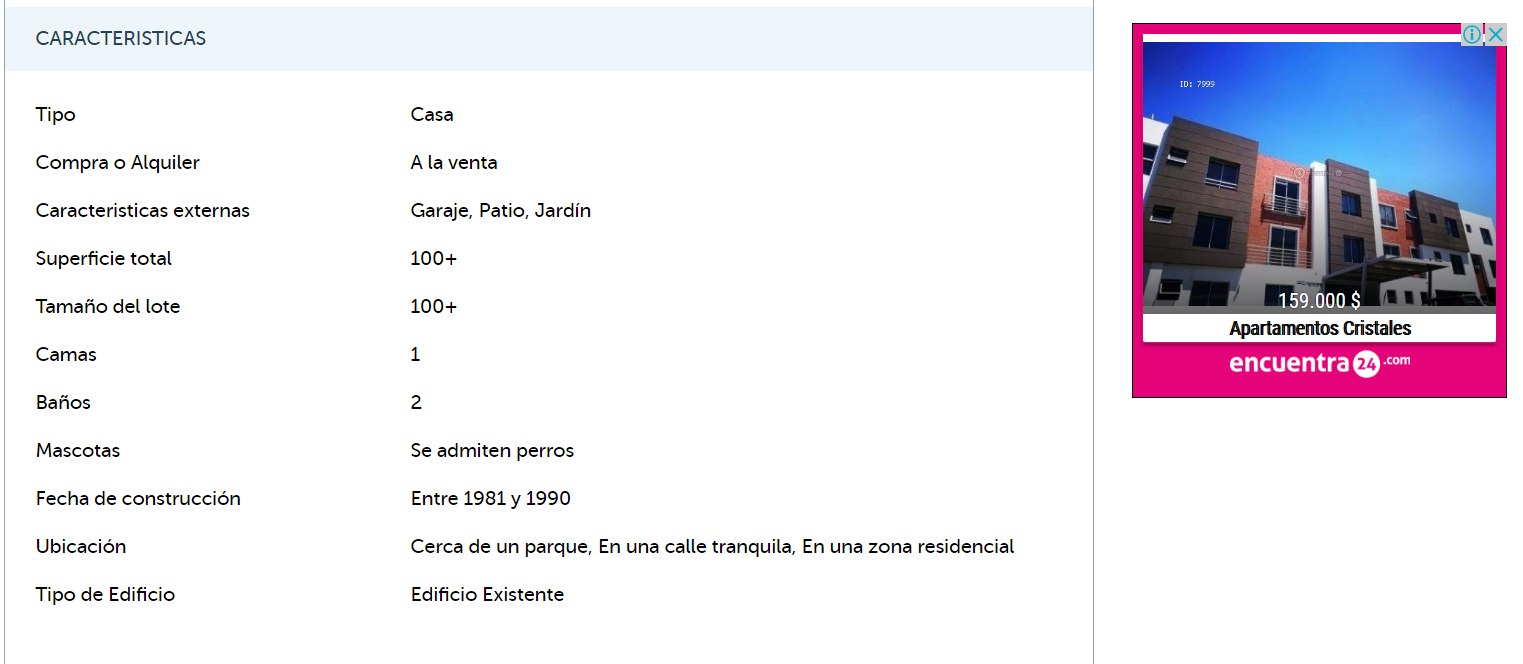


Cada página cuenta con una lista de 12 casas, cada una con la información que se muestra en la imagen de abajo.



Pero para poder obtener toda la información necesaria, se necesitará ir a los links específicos de cada casa, es decir, seleccionar cada celda y redirigirse a la página de casa en sí, como lo muestran las imágenes de abajo.







Descripción y xpaths de los atributos:

**Imagen**: Atributo que se utiliza para conocer la imagen de la casa.

XPATH: /html/body/div/article/div[1]/div[1]/div/a

**Titulo**: Atributo que se utiliza para conocer el título de la casa.

XPATH: //\*[@id="tekst1"]/h1/a/span

**Precio**: Atributo que se utiliza para conocer el precio de la casa.

XPATH: //\*[@id="text2"]

**Tipo**: Atributo que se utiliza para conocer si es una casa o apartamento.

XPATH: //\*[@id="mainarticles"]/div/section[1]/ul/li[1]/p

**Venta**: Atributo que se utiliza para conocer si es una casa de venta o alquiler.

XPATH: //\*[@id="mainarticles"]/div/section[1]/ul/li[2]/p

**Externas**: Atributo que se utiliza para conocer las características externas de la casa.

XPATH: //\*[@id="mainarticles"]/div/section[1]/ul/li[4]/p

**Superficie:** Atributo para conocer para conocer la superficie total de la casa.

XPATH: //\*[@id="mainarticles"]/div/section[1]/ul/li[5]/p

**Tamano:** Atributo que se utiliza para conocer el tamaño de la casa.

XPATH: //\*[@id="mainarticles"]/div/section[1]/ul/li[6]/p

**Camas**: Atributo que se utiliza para conocer el número de camas de la casa.

XPATH: //\*[@id="mainarticles"]/div/section[1]/ul/li[7]/p

**Banos**: Atributo que se utiliza para conocer el número de banos de la casa.

XPATH: //\*[@id="mainarticles"]/div/section[1]/ul/li[8]/p

**Mascotas**: Atributo para conocer si se permiten mascotas en la casa.

XPATH: //\*[@id="mainarticles"]/div/section[1]/ul/li[9]/p

**Fecha de Construccion:** Atributo para conocer la fecha de construcción de la casa.

XPATH: //\*[@id="mainarticles"]/div/section[1]/ul/li[10]/p

**Ubicación**: Atributo que se utiliza para conocer la ubicación de la casa.

XPATH: //\*[@id="mainarticles"]/div/section[1]/ul/li[11]/p

**Tipo de Edificio**: Atributo que se utiliza para conocer si el edificio es nuevo o no.

XPATH: //\*[@id="mainarticles"]/div/section[1]/ul/li[12]/p

**Visto**: Atributo que se utiliza para conocer cuántas veces han visto es casa.

XPATH: //\*[@id="mainarticles"]/div/div[2]/p

**Fecha Visto**: Atributo para conocer la última vez que esa casa fue vista.

XPATH: //\*[@id="mainarticles"]/div/div[2]/p

# Conclusiones

El realizar web scraping se ha convertido muy popular a nivel mundial ya que trae muchos beneficios si se sabe utilizar bien, se gana una mayor visibilidad en redes sociales, controlar la imagen y visibilidad de una marca en internet e incluso hasta generar un mejor marketing de contenidos, entre otros.

Una vez realizado un buen web scraping, se tiene que hacer un buen uso de datos obtenidos, para eso es importante tener conocimientos sobre software que ayude a visualizar mejor los datos y el uso de regex o también como popularmente se llaman expresiones regulares, que permiten trabajar con grandes cantidades de datos para corregir o depurar dicha información.

Por otro lado, es importante siendo el dueño de alguna página web asesorarse adecuadamente y tomar medidas para contrarrestar estás técnicas.

# Recomendaciones

* Tener bien claro el propósito del proyecto para saber cuál técnica de web scraping es la apropiada en dicha situación.
* Saber trabajar con grandes volúmenes de datos.
* Informarse sobre las implicaciones legales al realizar web scraping.
* Investigar sobre lenguajes de programación para realizar web scraping que se ajusten a las necesidades tomando en cuenta las características y potencia del lenguaje y su grado de aprendizaje.

# Bibliografía

*Feliciano Borrego*. (4 de Noviembre de 2016). Obtenido de http://felicianoborrego.com/alternativas-para-realizar-web-scraping/

Glez-Peña, D., Lourenço, A., López-Fernández , H., Reboiro-Jato, M., & Fdez-Riverola, F. (2013). Web scraping technologies in an API world. *BRIEFINGS IN BIOINFORMATICS*, 10.

Massimino, B. (2016). Accessing Online Data: Web-Crawling and Information-Scraping Techniques to Automate the Assembly of Research Data. *Journal of Business Logistics*, 9.

*sitelabs*. (8 de Abril de 2016). Obtenido de https://sitelabs.es/web-scraping-introduccion-y-herramientas/