

# Magic Alert

Grupo 7

Recuperación de Información y Recomendaciones en la  
WEB

Valentina Coggan	4741285-4
Nahuel Hourcade	4535586-0
Javier Morales	4780020-9
Santiago Pacheco	4894168-4
Santiago Serantes	5027617-8

<b>1. Introducción</b>	<b>3</b>
1.1. Motivación	3
<b>2. Objetivo</b>	<b>4</b>
<b>3. Enfoque de la solución</b>	<b>6</b>
3.1. Tecnologías	6
3.1.1. Python + Django	6
3.1.2. SQLite	7
3.1.2.1. Estructura de Datos	7
3.1.3. HTML + CSS + Javascript	7
3.1.3 Beautiful Soup 4	7
3.1.4. Scrapy	8
Arquitectura	8
Chequeo de Alertas	9
Obtención de Precios	9
Problemas encontrados	10
<b>4. Funcionalidades y uso</b>	<b>10</b>
4.1. Login	11
4.2. Dashboard	12
4.3. Nueva alerta	13
4.4. Servicios de scraping y envío de alertas	13
<b>5. Evaluación y resultados</b>	<b>14</b>
<b>6. Conclusiones</b>	<b>15</b>
<b>7. Trabajo futuro</b>	<b>16</b>
<b>8. Referencias</b>	<b>16</b>

# 1. Introducción

Magic: The Gathering<sup>1</sup> es un juego de cartas coleccionables creado en 1993. De hecho, es el juego de cartas coleccionables que inventó el rubro del TCG (Trading Card Game). El juego consiste de dos o más jugadores que compiten entre sí, cada uno con su propio mazo de cartas. Las reglas del juego no son de interés alguno a este trabajo, pero puede encontrarse un resumen [aquí](#).

Las cartas son producidas por la empresa Wizards of the Coast y comercializadas principalmente en sobres cerrados aleatorizados más conocidos como “*booster packs*” o solamente “*boosters*”. Las cartas tienen una rareza asociada (común, infrecuente, rara o mítica), y además, según su jugabilidad en torneos, resultan más o menos deseadas por los jugadores. Así es como nace el mercado secundario: múltiples vendedores, tanto empresas como particulares, revenden cartas individuales. En base a simple oferta y demanda, los precios tanto se estabilizan entre los distintos vendedores como fluctúan en el tiempo. Son estos precios fluctuantes los que motivan el desarrollo del proyecto.

## 1.1. Motivación

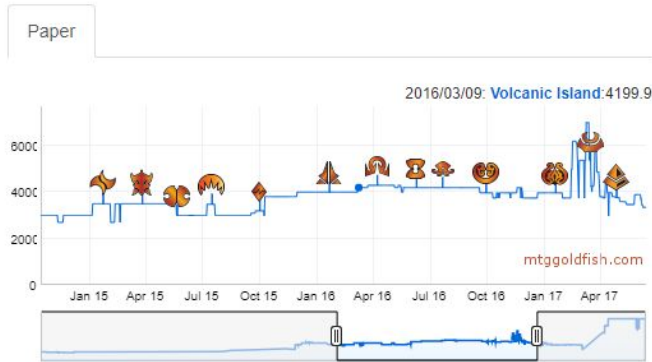


Como se ve en la figura, el precio de la carta Pelt Collector subió repentinamente. Quienes vendieron durante ese pico salieron beneficiados, mientras que ahora la carta recuperó su precio anterior. Existen numerosas páginas web que disponen de esta información y la presentan de manera gráfica y visualmente agradable, pero requieren que sea el usuario quien revise manualmente los precios. Con cientos de cartas en una colección, esto no es viable.

Por otro lado, existen cartas con un valor tan alto que incluso pequeñas variaciones porcentuales significan cambios significativos en valor, como se puede ver en la siguiente foto,

<sup>1</sup> <https://magic.wizards.com/en>

## Volcanic Island B Limited Edition Beta Reserved List




La carta Volcanic Island, por ejemplo, presenta varios cambios de aproximadamente un 15% en el rango de tiempo mostrado, pero al valer por arriba de los 3000 dólares en ese momento(y unos 8500 al día de hoy), un cambio de 15%, que porcentualmente puede parecer poco, resulta en unos 450 dólares.

## 2. Objetivo

El objetivo de nuestro proyecto es crear una aplicación que mantenga al usuario informado de la fluctuación de precios de un subconjunto de cartas previamente elegido por él. El propósito es dual: tanto si el usuario desea comprar una carta como venderla, el objetivo es conseguir el mejor precio posible.

Por ejemplo, un usuario desea comprar la carta “Chandra, Torch of Defiance”, pero no tiene una restricción de tiempo para la compra, solamente desea pagar menos de diez dólares por ella. En dos distintos sitios mostrados aquí, se muestran dos precios distintos para la carta. Lo que se busca es recibir una alerta cuando/si el precio baja a menos del precio indicado.



**Chandra, Torch of Defiance**  
Kaladesh (M)

Planeswalker - Chandra

[+1]: Exile the top card of your library. You may cast that card. If you don't, Chandra, Torch of Defiance deals 2 damage to each opponent.

[+1]: Add to your mana pool.

[-3]: Chandra, Torch of Defiance deals 4 damage to target creature.

**\$11.99**

NM EX VG G

8 available

[Add to Cart](#)

Chandra, Torch of Defiance en CardKingdom.com

Name	Category	Mana	Type	P/T	Rarity	Condition	Stock	Price
<b>Chandra, Torch of Defiance</b>	Kaladesh	2	Legendary Planeswalker 4 - Chandra		M	NM/M	15	\$10.99

Chandra, Torch of Defiance en StarCityGames.com

### 3. Enfoque de la solución

Para notificar a los usuarios, seguimos los precios de dos páginas web especializadas en la compra y venta de cartas, y notificamos al usuario cuando se cumplan las condiciones que él especificó previamente sobre las cartas de su interés.

Al usuario se le permite configurar las alertas sobre cada carta usando las siguientes condiciones:

- Mayor que X
- Menor que X
- Variación % en Y días (Y = 1, 3, 7)

Donde X es un valor fijo determinado por el usuario.

Los sitios de los que se toma información de precios son un subconjunto de los siguientes, que fueron considerados. Se marcan en negrita los que se utilizaron.

1. **CardKingdom.com**. Este sitio es de especial importancia en Uruguay porque es el que se toma como referencia en el mercado local.
2. Starcitygames.com
3. Trollandtoad.com
4. Abugames.com
5. Channelfireball.com
6. **MtgMintCard.com**

### Diseño e Implementación

Para desarrollar el proyecto se utilizaron las siguientes tecnologías:

## 3.1. Tecnologías

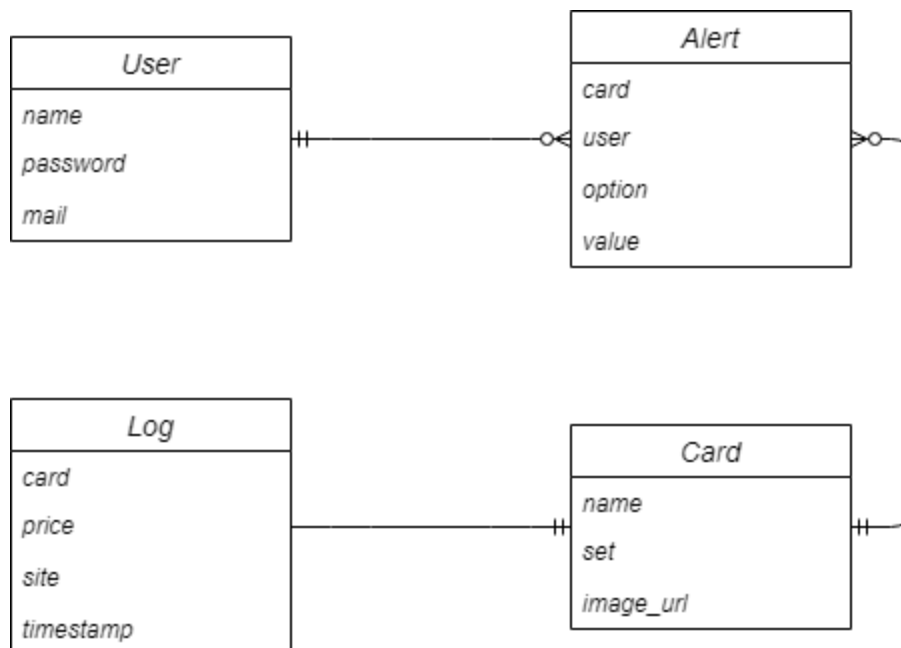
### 3.1.1. Python + Django

El lenguaje Python es apto para el desarrollo por su fácil integración con las herramientas de scrapping. El framework web Django se eligió pues cuenta con un esquema de plantillas, modelo MVC, ORM (*Object-Relation Mapper*, para acceso a base de datos), panel de administración y otras funciones para la construcción rápida de aplicaciones de gestión.

### 3.1.2. SQLite

Para la persistencia de los datos se utilizó SQLite por la compatibilidad nativa de Django con esta tecnología y la simplicidad de su uso. Si la aplicación fuera lanzada a producción, habría de usarse otra base de datos, por ejemplo Postgres, por razones de escalabilidad.

#### 3.1.2.1. Estructura de Datos



El campo option en las alertas es el que indica el tipo de alerta (mayor, menor, cambio porcentual en X días).

### 3.1.3. HTML + CSS + Javascript

Se utilizan las tecnologías clásicas para las vistas de la aplicación.

### 3.1.3 BeautifulSoup 4

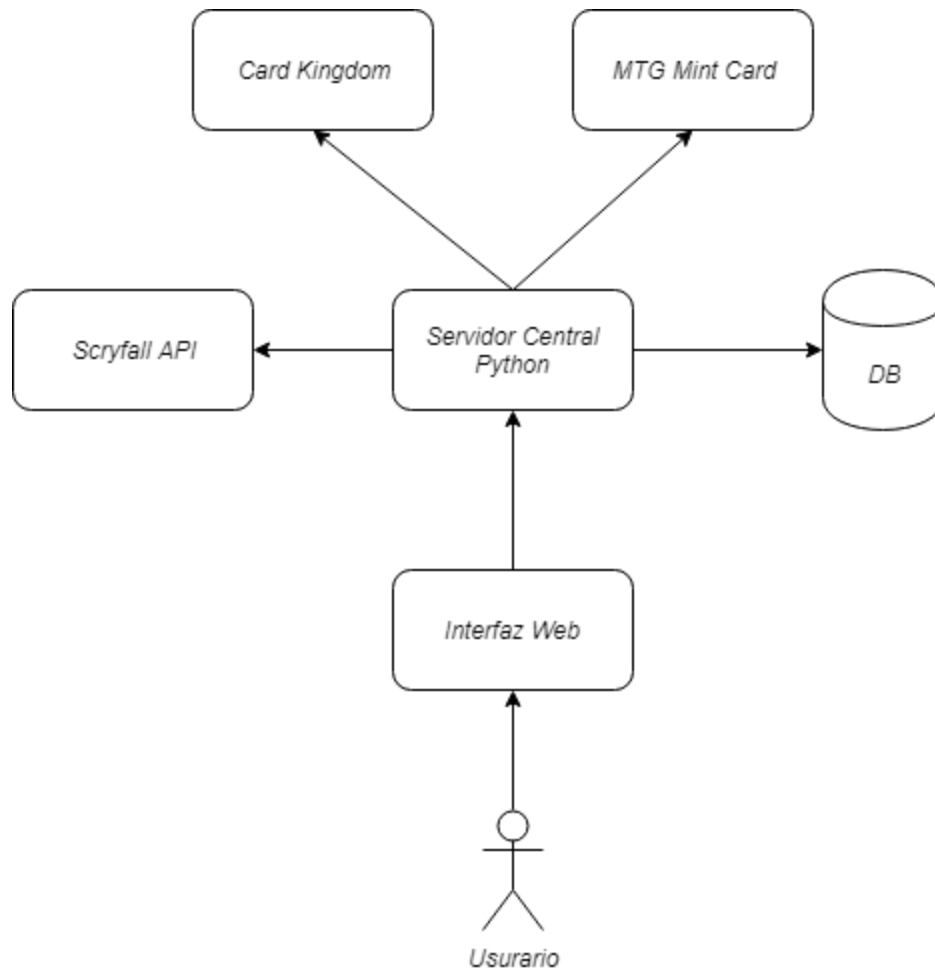
BeautifulSoup es una biblioteca de Python para extraer datos de archivos HTML y XML. Esta biblioteca crea un árbol con todos los elementos del documento y puede ser utilizado para extraer información. Proporciona formas idiomáticas de navegar, buscar y modificar este árbol. Es una herramienta muy útil para realizar web scraping y con una baja curva de aprendizaje.

### 3.1.4. Scrapy

Scrapy es un framework de Python para la extracción de información de páginas web. Si bien originalmente su función era la anteriormente dicha, también puede ser usada para extraer la información de APIs, o como un software de rastreo de la web. Ejemplos de aplicaciones en las que se utiliza Scrapy incluyen data mining, procesamiento de información o registros históricos. El funcionamiento de Scrapy se basa en Spiders o "arañas". Cada una de estas arañas contiene un conjunto de instrucciones que son las que indican de dónde deben extraerse los datos y cómo deben procesarse los datos obtenidos para la extracción de la información deseada.

## Arquitectura

Se presenta el diagrama de arquitectura del sistema.



A nivel lógico, se identifican cinco secciones principales:

- Plantillas o *templates* de las páginas del sitio.
- Lógica de las vistas de la aplicación, encargadas de recibir los pedidos y rellenar los templates.
- Lógica de los modelos, que determinan las estructuras de datos. Aquí se encuentra la lógica de cómo se disparan las alertas.
- Módulo de scrapping de precios.
- Módulo de acceso a la API de Scryfall<sup>2</sup> para información de cartas.

---

<sup>2</sup> Sitio que provee información de las cartas mediante una API. No provee datos de precios, pero ayuda a la validación de qué nombres de cartas existen, por ejemplo.



## Chequeo de Alertas

En esta sección se explica cómo el sitio obtiene los datos de precios y cómo los procesa. Las alertas que programan los usuarios se ingresan a la base de datos. Cada alerta contiene una carta relacionada y una condición. Por ejemplo, “Chandra, Torch of Defiance del set Kaladesh”, con condición “< \$10”.

Cada 24 horas, el proceso de **chequeo de alertas** corre en el servidor.

Su primer subrutina es la de **obtención de precios**. Para cada carta en el sistema (al registrar una alarma, si la carta a la que refiere no existe todavía en el sistema, se la agrega al mismo), obtiene su precio en cada sitio y lo almacena en la base de datos. Luego, para cada alerta se revisa si se cumple la condición con los precios almacenados actuales (y de los días anteriores para las condiciones de cambio porcentual en cierto tiempo), y se envía un correo al usuario correspondiente para las alertas que cumplan la condición.

## Obtención de Precios

Para obtener los precios se realiza un *scrapping* contra las páginas utilizadas, Card Kingdom y Mtg Mint Card en nuestro caso, utilizando la biblioteca Requests de Python, y luego se extrae del HTML el span correspondiente al precio, utilizando BeautifulSoup para el procesamiento, guardándose finalmente un registro en la base de datos con el precio encontrado.

Si bien la lógica es similar entre los dos sitios, lo que difiere es como se arma la url con la que se accede a la publicación de cada carta, y luego cual es el span del cual hay que sacar el precio.

Para el caso de Card Kingdom, la url es simplemente “/mtg/nombre-de-set/nombre-de-carta”, teniendo cuidado de cambiar espacios por “-”, y el eliminar símbolos especiales como contrabarra, apóstrofes, comas, etc. Una vez obtenido el HTML de la página, se extrae el precio a guardar buscando el span con la clase “stylePrice”.

MtgMintCard tiene un formato similar, siendo la url “mtg/singles/código-de-set/eng-reg/nombre-de-carta”. A diferencia de en Card Kingdom, este sitio utiliza un código de 3 letras que existe para todos los sets (por ejemplo, el set “Hour of Devastation” tiene como código “HOU”). Estos códigos los obtenemos desde Scryfall, pero como el sitio solo nos regresa la información de todos los códigos, realizamos un pre-proceso, guardando los datos en un diccionario de la forma “Nombre -> Código”, guardando además un timestamp. Dado que los códigos no cambian, y nuevos sets de cartas salen como mucho cada un par de meses, no es necesario actualizar este diccionario constantemente, para lo que se utiliza el timestamp mencionado, y solo se actualiza, al momento de realizar una consulta, si el diccionario es mayor a un cierto tiempo, en nuestro caso un día (para no demorar mucho en caso de que salga un set nuevo). Una vez obtenido el código del set, el procedimiento es similar al anterior, buscándose un span con itemprop “price”.

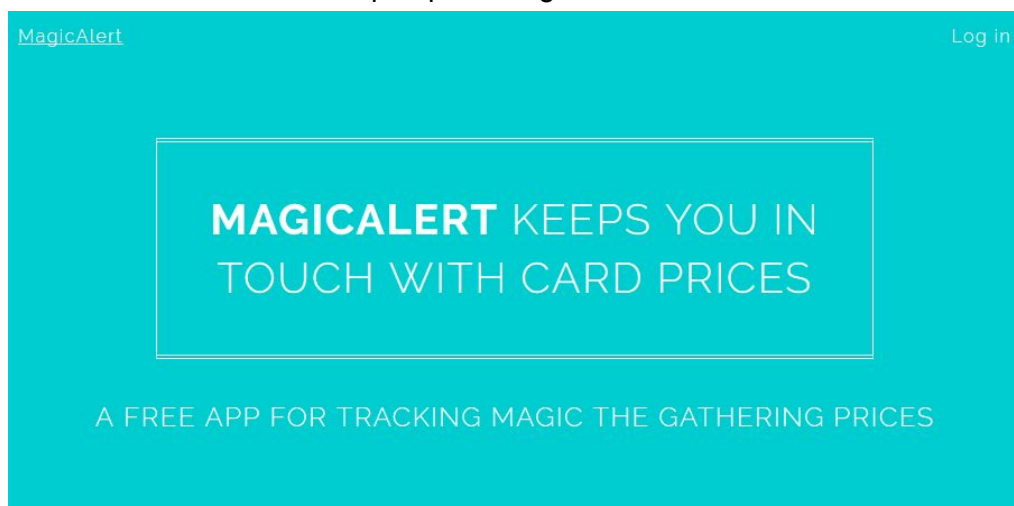
## Problemas encontrados

Algunos sitios de los que se mencionó utilizan carga dinámica: primero se carga el esqueleto de la página, y luego mediante Javascript y AJAX se obtiene el cuerpo de la misma. BeautifulSoup no soporta este tipo de carga, por lo tanto no se podía obtener con dicha biblioteca los precios de las cartas. Se investigaron otras opciones, por ejemplo el uso de Selenium, pero por razones de alcance, finalmente se decidió por la opción simple de utilizar otros sitios.

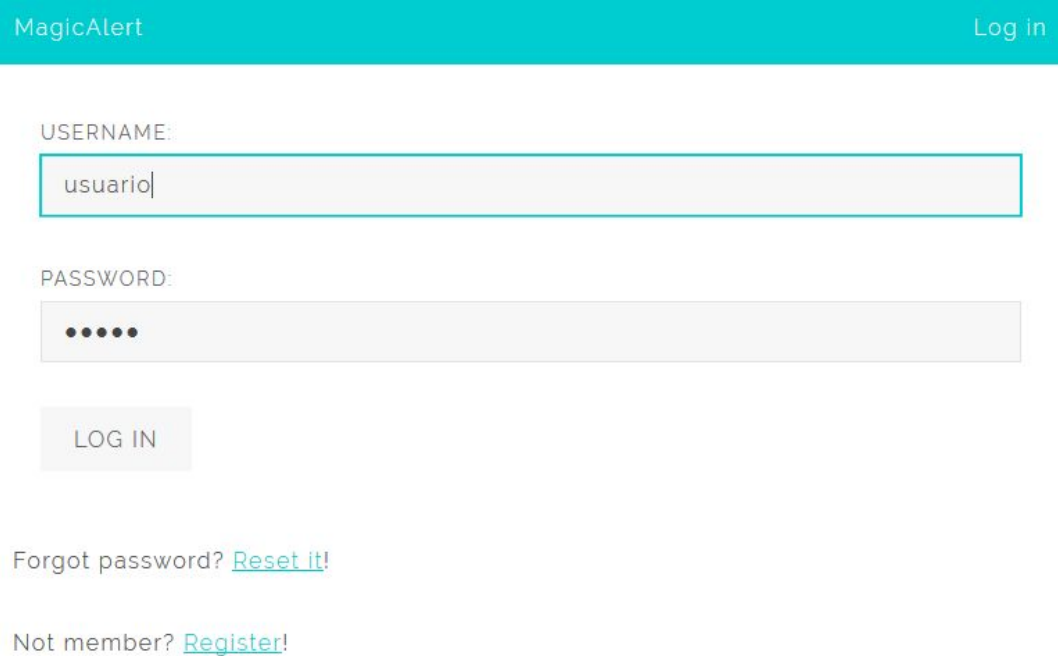
Otro problema encontrado fue que algunos de los nombres de los sets se encontraban en su forma completa en Scryfall (por ejemplo "Limited Edition Alpha"), pero tanto en el uso normal como en la página de Card Kingdom, se refiere a los mismos de manera abreviada ("Alpha"). Dado que el número de casos donde esto pasa es muy pequeño, se solucionó simplemente haciendo que el nombre corto fuera un alias del nombre completo.

## 4. Funcionalidades y uso

La aplicación tiene una interfaz web con la cual el usuario podrá interactuar para agregar y eliminar las alertas sobre las cartas que quiere seguir.



## 4.1. Login



MagicAlert Log in

USERNAME:

PASSWORD:

Forgot password? [Reset it!](#)




Not member? [Register!](#)

Es necesario mantener un perfil para cada usuario, donde se almacena su correo y al cual se asocian todos los datos que vaya generando a través de la página: qué cartas se siguen y con qué condiciones.

## 4.2. Dashboard

MagicAlert Dashboard Logged in: admin Log out Change password

MY ALERTS

Card	Name	Option	Value	Card Kingdom	Mint Card	
	Shock [Battlebond]	MENOR	U\$ 3.0	U\$ 0.25	U\$ 0.29	DELETE
	Walking Ballista [Aether Revolt]	MAYOR	U\$ 10.0	U\$ 17.99		DELETE
	Accomplished Automaton [Kaladesh]	MENOR	U\$ 3.0	U\$ 0.25	U\$ 0.24	DELETE

ADD ALERT

Una vez que el usuario realiza el login puede ver en el dashboard las alertas que ya haya ingresado previamente. Además se muestra en cada alerta el último precio obtenido de la carta indicada. Se muestran por separado los precios según el sitio. En caso de que el sitio no tenga stock no se publica el precio.

En este dashboard el usuario tiene las funciones para crear y borrar alertas. Puede crear una nueva alerta usando el botón *Add Alert* y puede borrar una alerta usando el botón *Delete* de la alerta correspondiente.

### 4.3. Nueva alerta

NUEVA ALERTA

CARD:

SET:

OPTION:

VALUE:

Para crear una nueva alerta el usuario debe ingresar el nombre de la carta, el set al que pertenece, una opción (tipo de condición para alertar) y el valor. Al momento de guardar se validará que el nombre de la carta y el set sean válidos. Para esto se utiliza la API ofrecida por Scryfall que permite corroborar que la carta ingresada existe y pertenece al set ingresado.

### 4.4. Servicios de scraping y envío de alertas

Periódicamente se estará corriendo un servicio que utilizando la configuración de cada alerta recuperará la información de precios de las cartas relevantes. Cuando el sistema detecte que se cumplen las condiciones definidas en la alerta le enviará un correo al usuario con la información de la carta, condición de la alerta y el precio.

## MagicAlert



*El precio de la carta Accomplished Automaton es menor a US\$ 3.0 y actualmente vale US\$ 0.25 en Card Kingdom*

## 5. Evaluación y resultados

Se evaluaron las distintas funcionalidades por separado: obtención de precios, definición de alertas, evaluación y envío de alertas.

La obtención de precios se implementó primero para el sitio Card Kingdom (el más relevante por ser referencia de precios en Uruguay). Se ejecutaron pruebas para obtener precios de cartas específicas utilizando la herramienta para hacer scraping. Por ser una funcionalidad crítica y la que tenía un mayor riesgo en la implementación fue la primera que se atacó y se validó su correcto funcionamiento.

En paralelo se trabajó la definición de alertas y la evaluación de las mismas. Se implementó la interfaz web que permitiera el ingreso de alertas y por otro lado la lógica que permite evaluar cuándo se debe enviar cada alerta según sus variables.

Cuando se terminó este trabajo se integraron las distintas partes para evaluar un flujo completo del uso de la aplicación. En este proceso se vio la necesidad de agregar validaciones sobre las alertas ya que fácilmente el usuario podría definir alertas sobre cartas inexistentes por error: ingresar mal el nombre de la carta o el set al que pertenece, o ingresar una carta y un set que no se corresponden. Para esto se utilizó la API de Scryfall que valida al momento de guardar la alerta que la carta definida sea correcta.

Sobre el sistema completo se probaron distintas alertas combinando las distintas condiciones y se corroboró que el comportamiento fuera correcto, sabiendo de antemano cuando debía o no enviarse una notificación por correo electrónico.

Finalmente se agregó la información de otro sitio: Mtg Mint Card. Sobre este cambio se volvió a testear el flujo completo ahora utilizando la información de ambos sitios. Los resultados finales fueron exitosos validando que el sistema se comporta de la forma esperada.

## 6. Conclusiones

Gracias a la metodología del scraping se pudo centralizar la información, obteniendo precios de distintos sitios de ventas de cartas. Este proceso además se programó para que se ejecute periódicamente así que es algo que también se actualiza en el tiempo. Esto permite no solamente obtener los valores actuales de las cartas en las alertas si no también calcular variaciones porcentuales a lo largo de varios días.

Se pudo implementar exitosamente la obtención de datos de dos sitios mediante scraping: Card Kingdom y MTG Mint Card. Ambos se implementaron con la herramienta BeautifulSoup que permite parsear el html de la página y mediante selectores rápidamente obtener los datos específicos que se quieren extraer del sitio. Se intentó extraer información de otros sitios que en la implementación de sus páginas utilizan carga dinámica, pero esto no es soportado por la herramienta que utilizamos. Para este proyecto se decidió sólo trabajar con sitios que no utilicen este tipo de tecnologías.

Además se logró implementar un sistema que permite definir alertas con distintos tipos de condiciones a evaluar sobre el precio: mayor que; menor que; cambio porcentual en 1, 3 o 7 días. El usuario tiene la flexibilidad para crear alertas sobre cualquier carta con cualquiera de las condiciones anteriores y fijando el valor de precio que le parezca apropiado. Combinando la información que se centralizó de los distintos sitios con la definición de las alertas el sistema puede evaluar en cada ejecución si se cumple la condición de cada alerta y en este caso enviar un correo al usuario correspondiente, el objetivo final de este proyecto.

El proyecto se implementó con herramientas que todo el mundo tiene acceso, software Open Source, y también por las prestaciones que tienen estas tecnologías elegidas, como django que permite generar sitios con poca cantidad de código, la complejidad del proyecto no es muy alta. A su vez podemos ver que la aplicación de este tipo de sistemas tiene un gran potencial, donde se le puede ofrecer al usuario un visión más compacta, acotada y ordenada de la información de la web sobre un tema en concreto. En este caso aplicado al mercado de cartas Magic, pero potencialmente sobre cualquier otro tema en el cual se quiera enfocar.

## 7. Trabajo futuro

Existen numerosas áreas donde mejorar la aplicación, pues es un servicio que no satisfacen otras aplicaciones en la web y más aún es solo una prueba de concepto del mismo. Estas son funcionalidades que se podrían implementar, además de la adición de más sitios que provean información:

- Alertas adaptativas: si una alerta se cumplió, por ejemplo, Chandra, Torch of Defiance bajó a menos de diez dólares, que el sitio sugiera al notificar que la alerta se cumplió, modificar el valor: *“ahora quiero enterarme cuando baje de nueve”*.
- Aplicación móvil con soporte de notificaciones PUSH.
- Alertas con consideración (opcional) de costos de envío.
- Alertas para solamente algunos de los sitios soportados, por ejemplo: *“no me interesa MtgMintCard”*.
- Alertas combinadas para múltiples cartas: *“avísame cuando todo este mazo valga menos de 150 dólares”*.
- Soportar búsqueda en distintos idiomas (las cartas no solo se imprimen en Inglés, sino que también en varios otros lenguajes).

## 8. Referencias

1. Magic: The Gathering - <https://magic.wizards.com/en>
2. Magic gameplay - <https://magic.wizards.com/en/gameplay/how-to-play>
3. Django - <https://www.djangoproject.com/>
4. BeautifulSoup - <https://www.crummy.com/software/BeautifulSoup/>
5. Scryfall API - <https://scryfall.com/docs/api>
6. Card Kingdom - <https://www.cardkingdom.com/>
7. Mtg Mint Card - <http://www.mtgmintcard.com/>