

## PRACTICA DE SCRAPPING

Elaborado:

Ing. David Vilatuña

Objetivo:

En la presente practica se elabora establece como parámetros la búsqueda automatizada en un sitio web de criterios para que el programa se enlace y entregue, datos a telegram en bot, subir información mongodb

Análisis:

Se busca en que pagina podemos realizar la automatización y tenemos la siguiente:

<https://srienlinea.sri.gob.ec/sri-en-linea/SriRucWeb/ConsultaRuc/Consultas/consultaRuc>

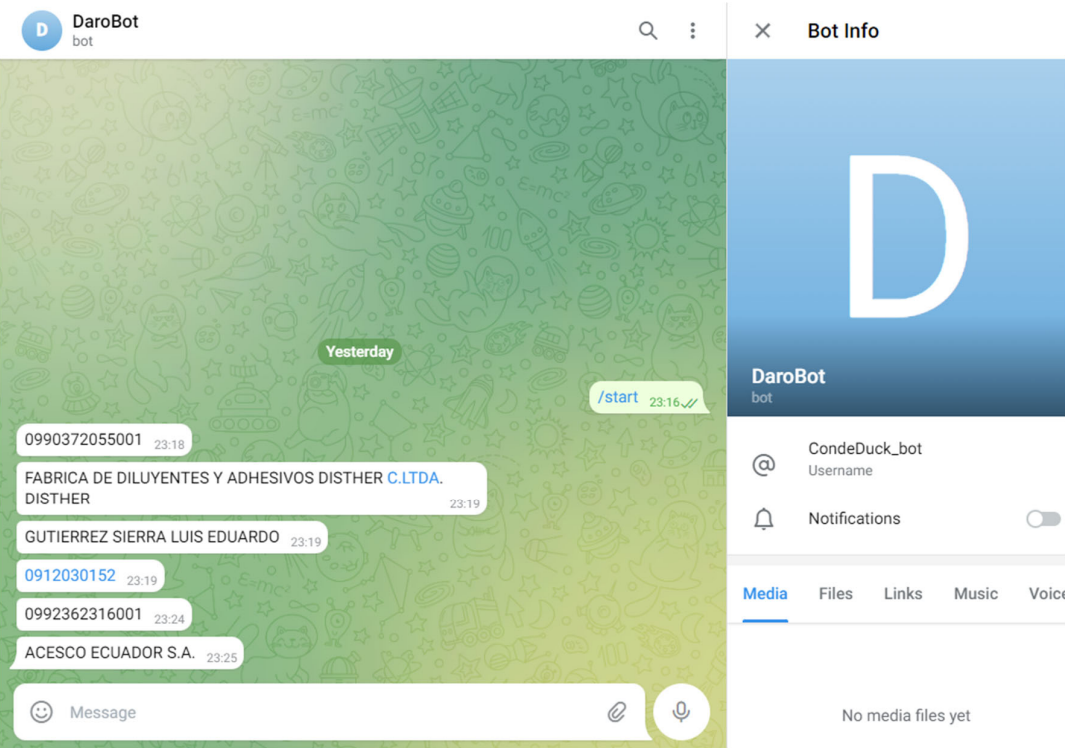


El funcionamiento de la misma requiere se ingrese los datos de un RUC comercial y esta nos entrega la información de la empresa.

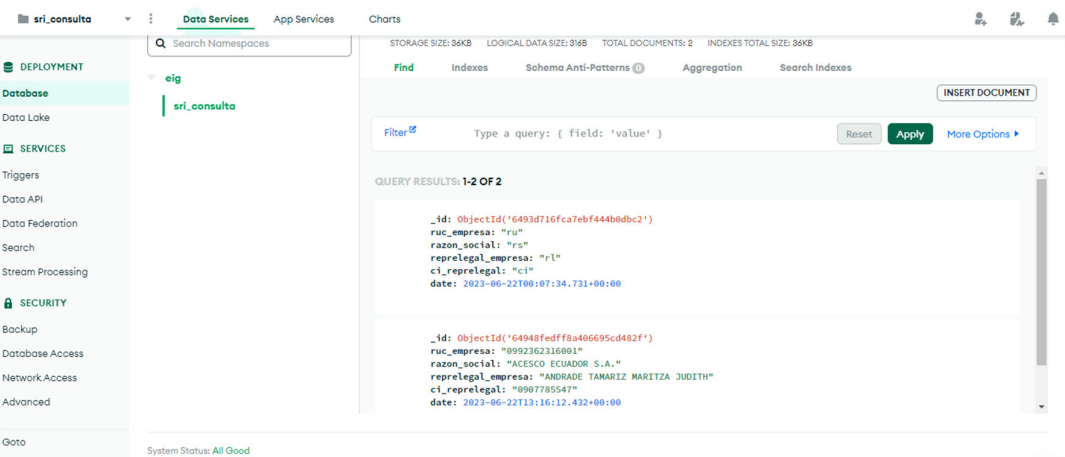


Esta información, es lo que realiza la entrega el sitio web, como manejo en segundo plano por parte del curso vamos a ingresar estos valores resultantes en el bot y guardarlo en la base de datos mongodb.

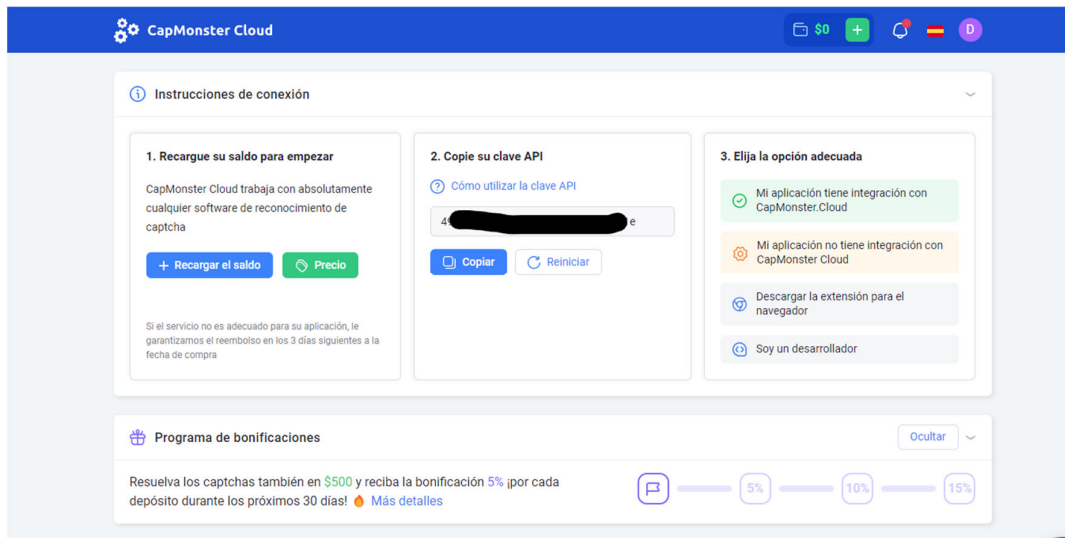
Información enviada a Telegram



Y la información enviada a MongoDB



Anexo a ello para poder saltar la seguridad de recaptcha, se ha realizado la consulta y se ha obtenido los siguientes recursos.



Recurso visual:

<https://www.youtube.com/watch?v=-TMNh64ubyM>

Recurso de codificación:


<https://gist.github.com/alperensert>

En el modulo main.py, se encuentra toda la codificación confirmada, el problema que tenemos es que es pago y no se tiene saldo en la cuenta.

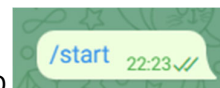
### **CONFIGURACION DE UN NUEVO BOT.**

1. Ingresamos en Telegram



BotFather   Wed  
Done! Congratulations on your new...

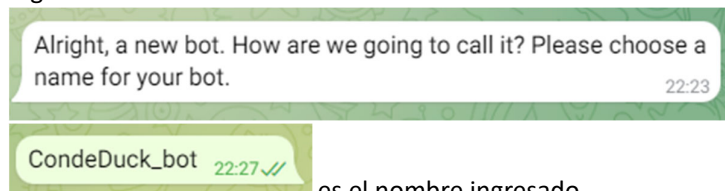
2. Buscamos el bot



3. Ingresamos, Presionamos el botón de INICIO

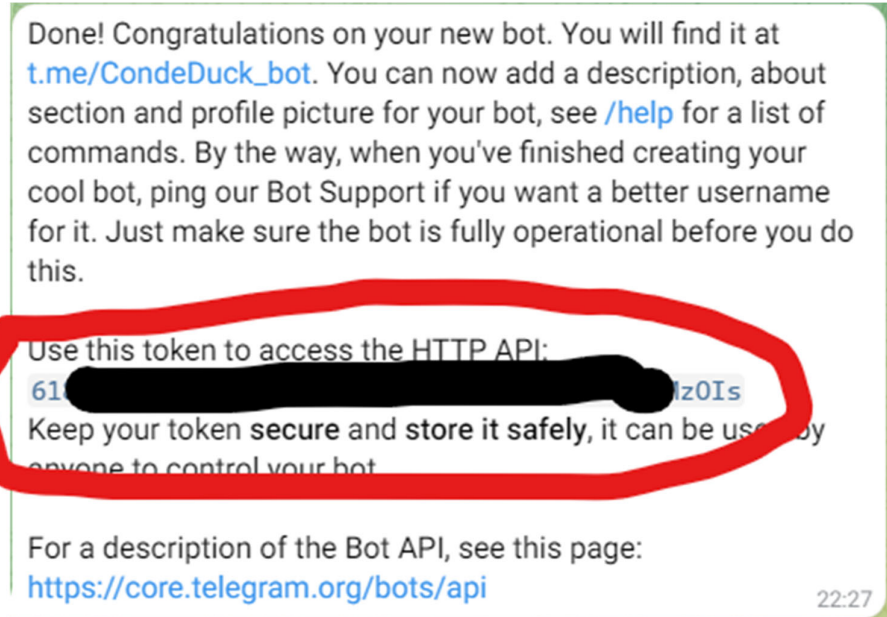
4. Vamos a [/newbot](#) - create a new bot

5. Ingresamos un nombre:

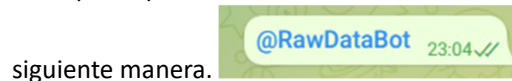


6. es el nombre ingresado.

7. Nos genera el mensaje:



8. De este mensaje necesitamos el token lo copiamos y lo pegamos en el archivo de referencias a variables globales que no se suben a git .env
9. Se requiere para el uso de bot la identificación de nuestro usuario en telegram de la



siguiente manera.

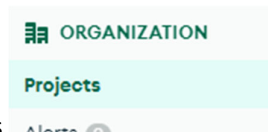
10. Nos vamos al siguiente chat:
11. Presionamos INICIO
12. Nos genera el ID del usuario este id es unico asi se cambie de nombre:



- 13.
14. Estos datos se requieren para el desarrollo del programa

## **CONFIGURACION DE MONGODB**

1. Ingresamos a <https://account.mongodb.com/account/login?signedOut=true>
2. Nos logueamos con Google o git



DARO'S ORG - 2023-06-10

## Projects

3. Nos vamos a proyectos
4. Creamos un nuevo proyecto,
5. Nombre del proyecto

### Name Your Project

Project names have to be unique within the organization (and other restrictions).

← ChatGPT Writer ⚙

Cancel

Next

6. En el caso de ingresar mas usuario al proyecto

[DARO'S ORG - 2023-06-10 > PROJECTS](#)

## Create a Project

✓ Name Your Project

Add Members

### Add Members and Set Permissions

Give your members access permissions below.

darovilas13@gmail.com  
(you)


Project Owner ▼

Back

Cancel

Create Project

7. Creamos la base de datos.




## Create a database

Choose your cloud provider, region, and specs.

**Build a Database**

Once your database is up and running, live migrate an existing MongoDB database into Atlas with our [Live Migration Service](#).

8. Elegimos la opcion free

 **MongoDB.**

### Deploy your database

Use a template below or set up [advanced configuration options](#). You can also edit these configuration options once the cluster is created.

STORAGE	RAM	vCPU
10 GB	2 GB	2 vCPUs

**M10** **\$0.08/hour**

For production applications with sophisticated workload requirements.

STORAGE	RAM	vCPU
Up to 1TB	Auto-scale	Auto-scale

**SERVERLESS** **\$0.10/1M reads**

For application development and testing, or workloads with variable traffic.




STORAGE	RAM	vCPU
512 MB	Shared	Shared


**M0** **FREE**


For learning and exploring MongoDB in a cloud environment.

9. Elegimos el proveedor y el datacenter mas cercano


**Provider**


  

**Region** ★ Recommended region 

 **Iowa (us-central1)** ★

**Americas**

 Iowa (us-central1) ★

 **Sao Paulo (southamerica-east1)** ★

10. Mantenemos en nombre del cluster

**Name**  
You cannot change the name once the cluster is created.

11. Elegimos la seguridad

1 How would you like to authenticate your connection?

Your first user will have permission to read and write any data in your project.


**Username and Password**


**Certificate**


12. Creamos el usuario es importante tener en cuenta que el password es la unica vez que se presentara por lo cual se debe grabar

Create a database user using a username and password. Users will be given the *read and write to any database privilege* by default. You can update these permissions and/or create additional users later. Ensure these credentials are different to your MongoDB Cloud username and password.

**Username**

**Password** 

 Autogenerate Secure Password

 Copy



**Create User**

13. Añadimos, las ips que tengan acceso a esta base de datos.

Only an IP address you add to your Access List will be able to connect to your project's clusters. You can manage existing IP entries via the [Network Access Page](#).

IP Address	Description	
<input type="text" value="Enter IP Address"/>	<input type="text" value="Enter description"/>	<b>Add My Current IP Address</b>
<b>Add Entry</b>		

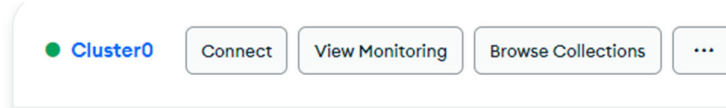
---

IP Access List	Description	
0.0.0.0/0		 <b>EDIT</b>  <b>REMOVE</b>

En este caso ponemos que todo el mundo tenga acceso a nuestra base de datos.



14. Con ello creamos la base de datos



15. Con esas opciones nos entregan las formas de conectarnos ver el monitor y mirar las colecciones que serian los datos

