

Anexo: Probando la Prueba de Concepto (PoC)

En este anexo se explican los pasos a dar para configurar la Prueba de Concepto realizada y descrita en las secciones previas:

1. **Crear el chat de Telegram que usaremos para notificación.** Como la aplicación *notifcryptovalues* se ejecuta en PC, lo más apropiado de usar para configurar esta parte es Telegram Desktop (<https://desktop.telegram.org/?setln=es>). Hay un bot en Telegram que es el encargado de manejar todos nuestros bots. Se puede encontrar como @botfather.

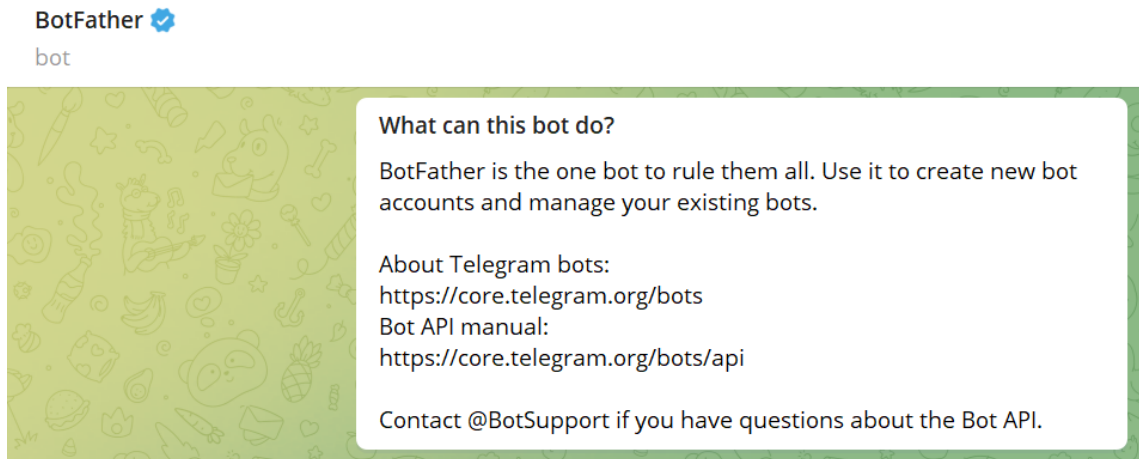


Figura 1. Bot @BotFather en Telegram.

Una vez comencemos la comunicación, deberemos mandar el comando `/newbot` y añadir el nombre y *username* que nos permitirá diferenciarlo del resto (este nombre de usuario deberá acabar en bot).

`/newbot`

Alright, a new bot. How are we going to call it? Please choose a name for your bot.

13:49

Figura 2. Elegir un nombre para el chat a crear.

Ejemplo: `notifcryptovalues_bot`

Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot.

13:51

Figura 3. Elegir un *username* para el chat a crear.

Ejemplo: `notifcryptovalues_bot`

Una vez hecho esto, el bot ya estará creado y nos aparecerá un mensaje con toda la información que necesitamos al respecto. Entre esta información tenemos el *token*.

Done! Congratulations on your new bot. You will find it at [REDACTED]. You can now add a description, about section and profile picture for your bot, see [/help](#) for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:

[REDACTED]
Keep your token **secure** and **store it safely**, it can be used by anyone to control your bot.

For a description of the Bot API, see this page:

<https://core.telegram.org/bots/api>

13:52

Figura 4. Información con el *token* (se ha ocultado en negro el utilizado).

/help

I can help you create and manage Telegram bots. If you're new to the Bot API, please [see the manual](#).

You can control me by sending these commands:

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

[/mybots](#) - edit your bots **[beta]**

Edit Bots

[/setname](#) - change a bot's name

[/setdescription](#) - change bot description

[/setabouttext](#) - change bot about info

[/setuserpic](#) - change bot profile photo

[/setcommands](#) - change the list of commands

[/deletebot](#) - delete a bot

Bot Settings

[/token](#) - generate authorization token

[/revoke](#) - revoke bot access token

[/setinline](#) - toggle [inline mode](#)

[/setinlinegeo](#) - toggle [inline location requests](#)

[/setinlinefeedback](#) - change [inline feedback](#) settings

[/setjoiningroups](#) - can your bot be added to groups?

[/setprivacy](#) - toggle [privacy mode](#) in groups

Games

[/mygames](#) - edit your [games](#) **[beta]**

[/newgame](#) - create a new [game](#)

[/listgames](#) - get a list of your games

[/editgame](#) - edit a game

[/deletgame](#) - delete an existing game

20:12

Figura 5. Comandos para gestionar el bot.

/mybots

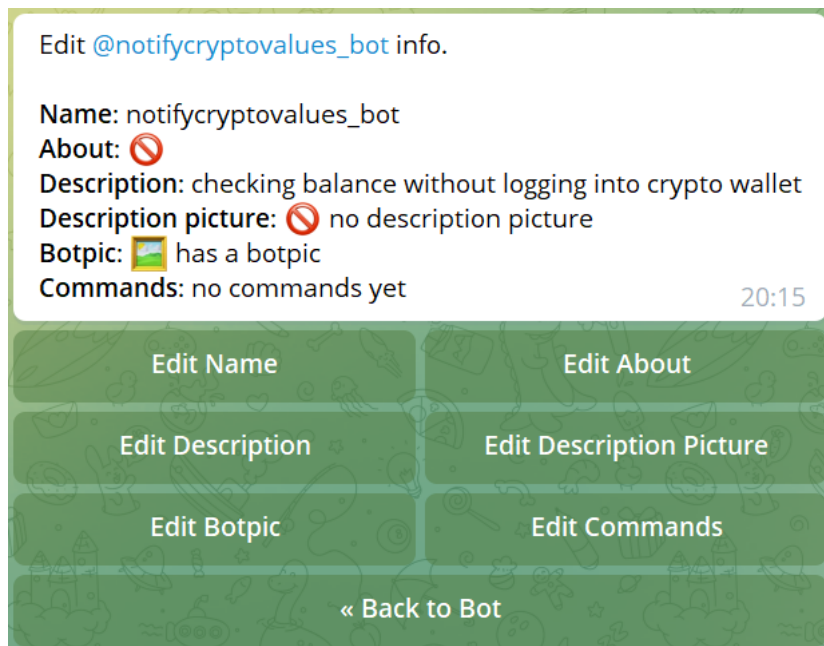


Figura 6. Editando el chat creado.

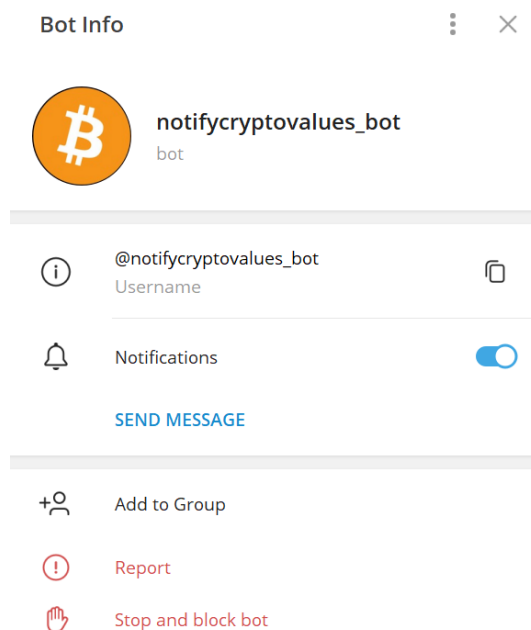


Figura 7. Información del chat creado en Telegram.

Obtener el ID del chat de Telegram creado. Para poder mandar una notificación vía Telegram desde nuestra Prueba de Concepto necesitamos: el *token* y el ID del chat al que queremos enviar la notificación. El primero ya lo tenemos, por lo que vamos a ver cuál es el identificador de nuestra conversación. Iniciamos una conversación con nuestro bot. Esto lo podemos hacer utilizando el enlace que nos aparece en el mensaje y confirmación de creación o buscándolo en Telegram por el nombre de usuario que le hemos asignado.

Una vez hecho esto pasaremos a acceder a la página con todas las actualizaciones que ha recibido. Abrimos cualquier navegador Web y accedemos a la siguiente página:

```
https://api.telegram.org/bot [token] /getUpdates
```

Deberemos cambiar [token] por el de nuestro propio bot. Por ejemplo, un ejemplo de URL podría quedar de la siguiente forma:

```
https://api.telegram.org/bot1725563685:AAGQNBsLvzA-  
ssf0QsqNQ9K3BUIB1NVnbJ8/getUpdates
```

Una vez accedamos a esta dirección, nos aparecerá un fichero JSON con un array de un solo objeto (si sólo hemos enviado un mensaje). Dentro de este objeto se encuentra otro objeto llamado chat. Si accedemos a él veremos que cuenta con la propiedad id. Ése es el dato que nos interesa y que deberemos guardar.

2. **Crear el chat de WhatsApp que usaremos para notificación.** Como la aplicación *notifycryptovalues* se ejecuta en PC, lo más apropiado de usar para configurar esta parte es WhatsApp Web (<https://web.whatsapp.com/>).



Figura 8. Página de WhatsApp Web.

Utilizamos la app WhatsApp de nuestro dispositivo móvil y vamos a Dispositivos vinculados para vincular nuestro dispositivo móvil.



Figura 9. Dispositivos vinculados de WhatsApp.

Escanemos el código QR con nuestro dispositivo móvil. Una vez esto, estamos con el PC dentro de WhatsApp Web.

Creamos un nuevo grupo.

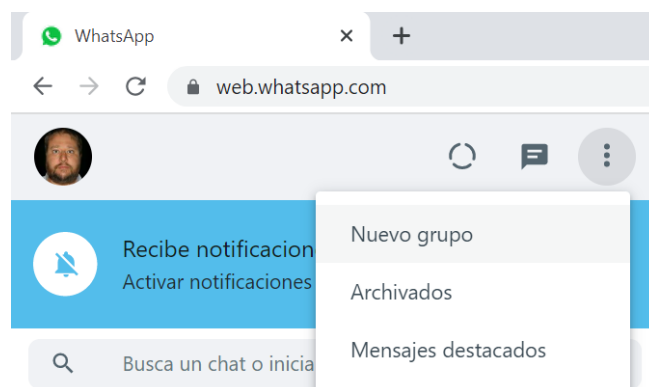


Figura 10. Nuevo grupo de WhatsApp.

En la siguiente figura se ve el grupo creado en WhatsApp.

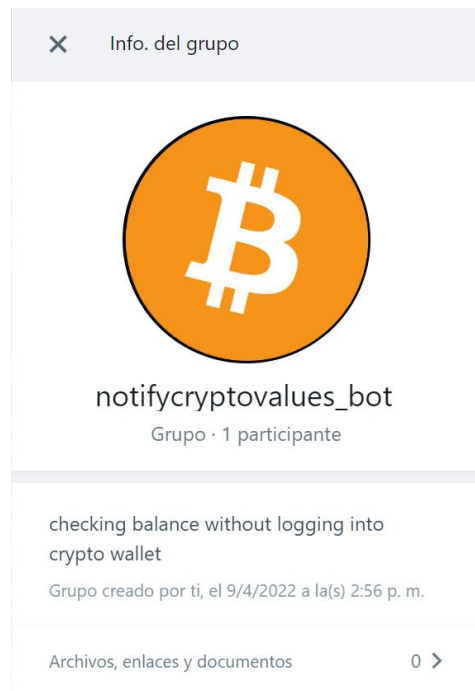


Figura 11. Información del grupo creado en WhatsApp.

3. Descargar el fichero .zip con la aplicación *notifcryptovalues* del siguiente enlace de OneDrive:

<https://1drv.ms/u/s!AhkSKZxSDxNhgRnTLfgCwmA7Gu5k>

4. Descomprimir el fichero .zip
5. Editar el código fuente con estos valores que habíamos guardado anteriormente:

```
chatNameWhatsApp
chatIdTelegram
apiToken
```

En:

```
notifcryptovalues\src\main\java\execution\Main.java
```

```
public class Main {

    public static void main(String[] args) {

        String chatNameWhatsApp = "to complete";
        String chatIdTelegram = "to complete";
```

En:

```
notifcryptovalues\src\main\java\sendmessage\Unit.java
```

```
public class Unit {

    public static void main(String[] args) {
```

```
String chatNameWhatsApp = "to complete";  
String chatIdTelegram = "to complete";
```

En:

```
notifycryptovalues\src\main\java\sendmessage\TelegramEncode.java
```

```
String apiToken = "to complete";
```

En:

```
notifycryptovalues\src\test\java\execution\TestCase1.java
```

```
@DisplayName("Testing sendmessage::WhatsApp")  
@Test  
void test2() {  
    String chatNameWhatsApp = "to complete";  
  
    @DisplayName("Testing sendmessage::Telegram")  
    @Test  
    void test3() {  
        String chatIdTelegram = "to complete";
```

6. Damos a guardar los cambios.
7. Generamos el fichero `.jar` de la aplicación *notifycryptovalues*. Lo podemos hacer con Maven por línea de comandos. O desde el entorno de Eclipse IDE una vez importado el proyecto.
8. Movemos el fichero `.jar` generado en `notifycryptovalues\target\` de nombre `notifycryptovalues-0.0.1-SNAPSHOT-jar-with-dependencies.jar` un directorio arriba a `notifycryptovalues\`
9. Ejecutamos el fichero `.jar` (que contiene la aplicación *notifycryptovalues*) por línea de comandos: `java -jar notifycryptovalues-0.0.1-SNAPSHOT-jar-with-dependencies`

Como vemos esta ejecución es muy parecida a la prueba de integración realizada. Y donde incluso facilitamos un vídeo a modo de demostración.