

PROYECTO DE BLOCKCHAIN

Crypto Actas

INTEGRANTES
Nombre y Apellido
Andrés Daniel Chimuris Gimenez
Juan Cristian Miguel
Matias Leonel Bassi
Nicolas Matías Garrido

INDICE

Instructivo de instalación	3
Ejemplo de uso	12
Generación de planes de estudio	12
Persistencia de Plan de estudios en Blockchain	13

Instructivo de instalación

1. Descargarse los archivos fuente del proyecto, ya sea clonando el repositorio o bajarlo en un [zip](#). Por línea de comandos puede usarse la siguiente instrucción:

```
git clone https://gitlab.com/m3851/cryptoactas.git
```

Nota: Es necesario tener instalado git, si y sólo si, la descarga de los fuentes lo hace a través de la clonación del repositorio.

2. Posicionarse en el directorio del proyecto. En nuestro ejemplo el directorio es el siguiente:

```
C:\src\blockchain\Proyecto\cryptoactas
```

3. Abrir Node.Js Command Prompt como administrador y escribir el siguiente comando

```
cd C:\src\blockchain\Proyecto\cryptoactas
```

4. Ejecutar el comando:

```
npm run install
```

Esto descargará todos los paquetes necesarios para ejecutar el servidor y el cliente react de la aplicación.

5. Instalar mongodb del sitio <https://www.mongodb.com/try/download/compass>

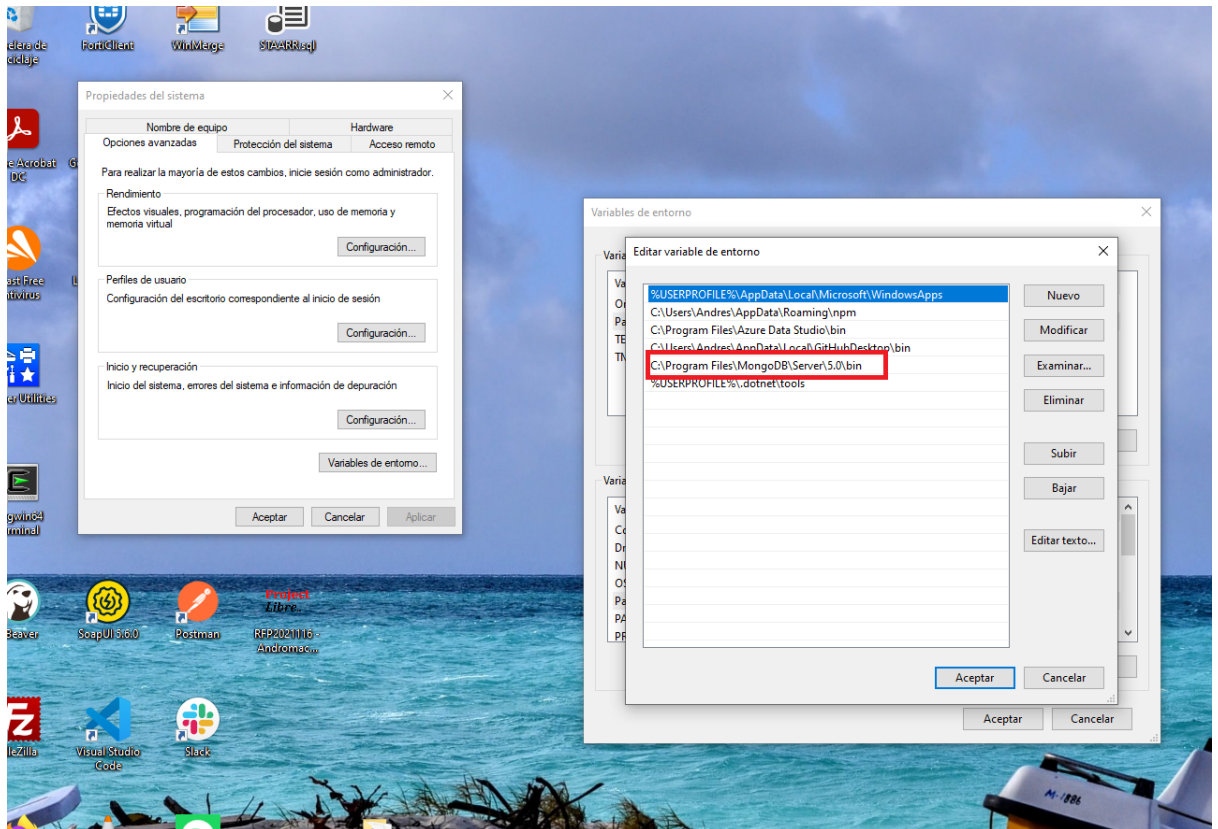
MongoDB Compass

Easily explore and manipulate your database with Compass, the GUI for MongoDB. Intuitive and flexible, Compass provides detailed schema visualizations, real-time performance metrics, sophisticated querying abilities, and much more.

Please note that MongoDB Compass comes in three versions: **a full version** with all features, **a read-only version** without write or delete capabilities, and **an isolated edition**, whose sole network connection is to the MongoDB instance.

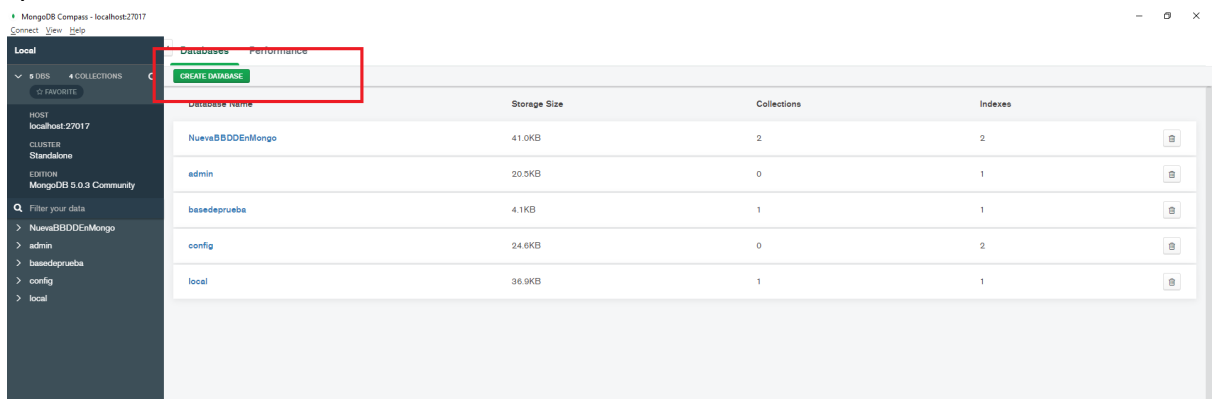
6. Crear la carpeta vacía en la ruta **c:/data/db** de Windows.
7. Agregar en la variable path la ruta a la carpeta bin de la instalación. Por ejemplo, para la versión 5.0 puede ser:

```
C:\Program Files\MongoDB\Server\5.0\bin
```

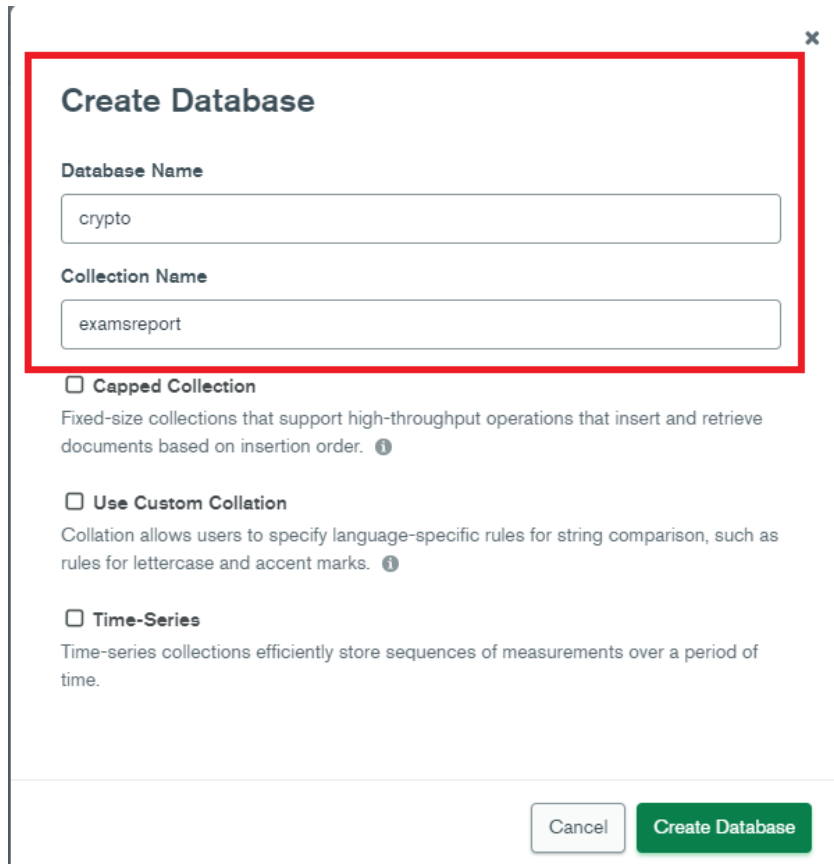


Con esto, finalizamos la instalación de MongoDB

8. Abrir MongoDB Compass.
9. Presionar el botón *Create Database* para crear la base de datos que utilizará la aplicación.



10. Definir una base con el nombre *crypto* con una collection de nombre *examsreport* como se muestra en la imagen.



Create Database

Database Name

crypto

Collection Name

examsreport

☐ **Capped Collection**
Fixed-size collections that support high-throughput operations that insert and retrieve documents based on insertion order. ⓘ

☐ **Use Custom Collation**
Collation allows users to specify language-specific rules for string comparison, such as rules for lettercase and accent marks. ⓘ

☐ **Time-Series**
Time-series collections efficiently store sequences of measurements over a period of time.

Cancel Create Database

Donde *examsreport* es una de las colecciones de datos requeridas por la aplicación.

11. Crear las siguientes colecciones:

- a. majors
- b. users
- c. syllabus
- d. universities

Se muestra a continuación un ejemplo donde se crea la colección *universities*.

x

Create Collection

Collection Name

universities

☐ Capped Collection

Fixed-size collections that support high-throughput operations that insert and retrieve documents based on insertion order. ⓘ

☐ Use Custom Collation

Collation allows users to specify language-specific rules for string comparison, such as rules for lettercase and accent marks. ⓘ

☐ Time-Series

Time-series collections efficiently store sequences of measurements over a period of time.

Cancel

Create Collection

Luego de estas operaciones, la base quedará configurada de esta forma:

MongoDB Compass - localhost:27017/crypto
Connect View Help

Local

6 DBS 9 COLLECTIONS

☆ FAVORITE

HOST
localhost:27017

CLUSTER
Standalone

EDITION
MongoDB 5.0.3 Community

Filter your data

> NuevaBBDDenMongo

> admin

> basedepueba

> config

✓ crypto

- examsreport
- majors
- syllabus
- universities
- users

> local

Collections

CREATE COLLECTION

Collection Name ^	Documents
examsreport	0
majors	0
syllabus	0
universities	0
users	0

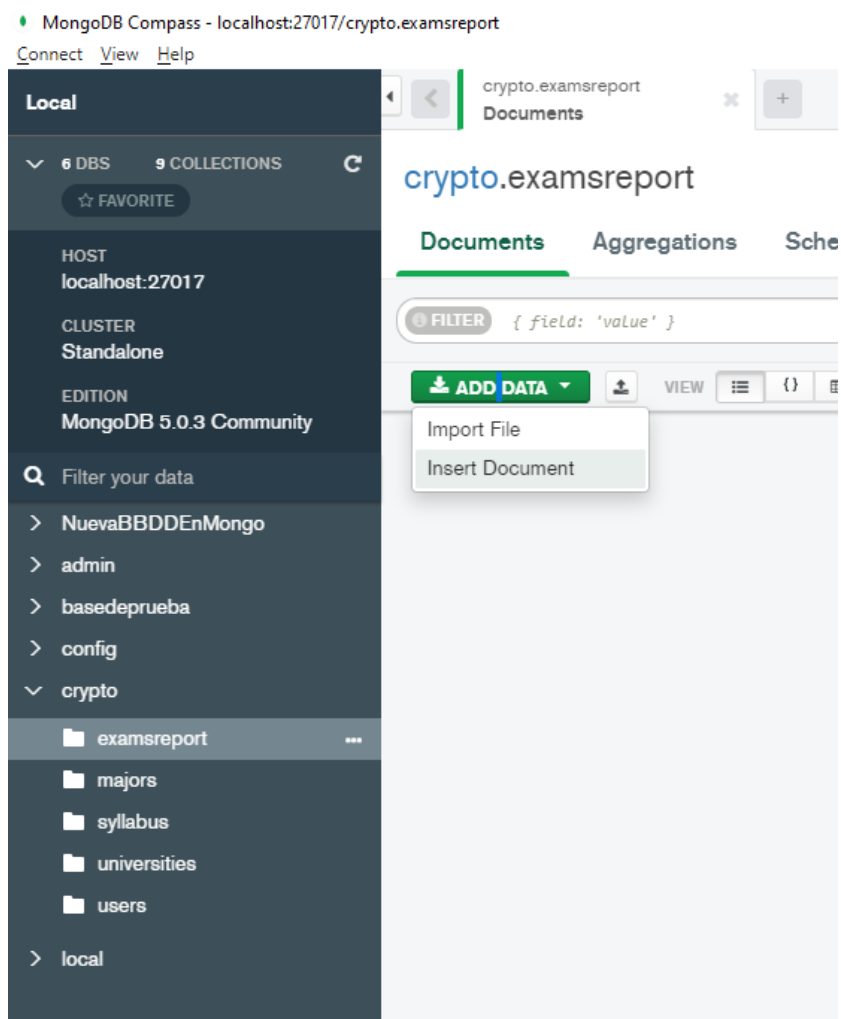
- Extraer los archivos ubicados dentro de la sub-carpeta DataBlockChain en el directorio del proyecto. Esta ruta podría ser por ejemplo como la que sigue:

```
C:\src\blockchain\Proyecto\cryptoactas\DataBlockChain
```

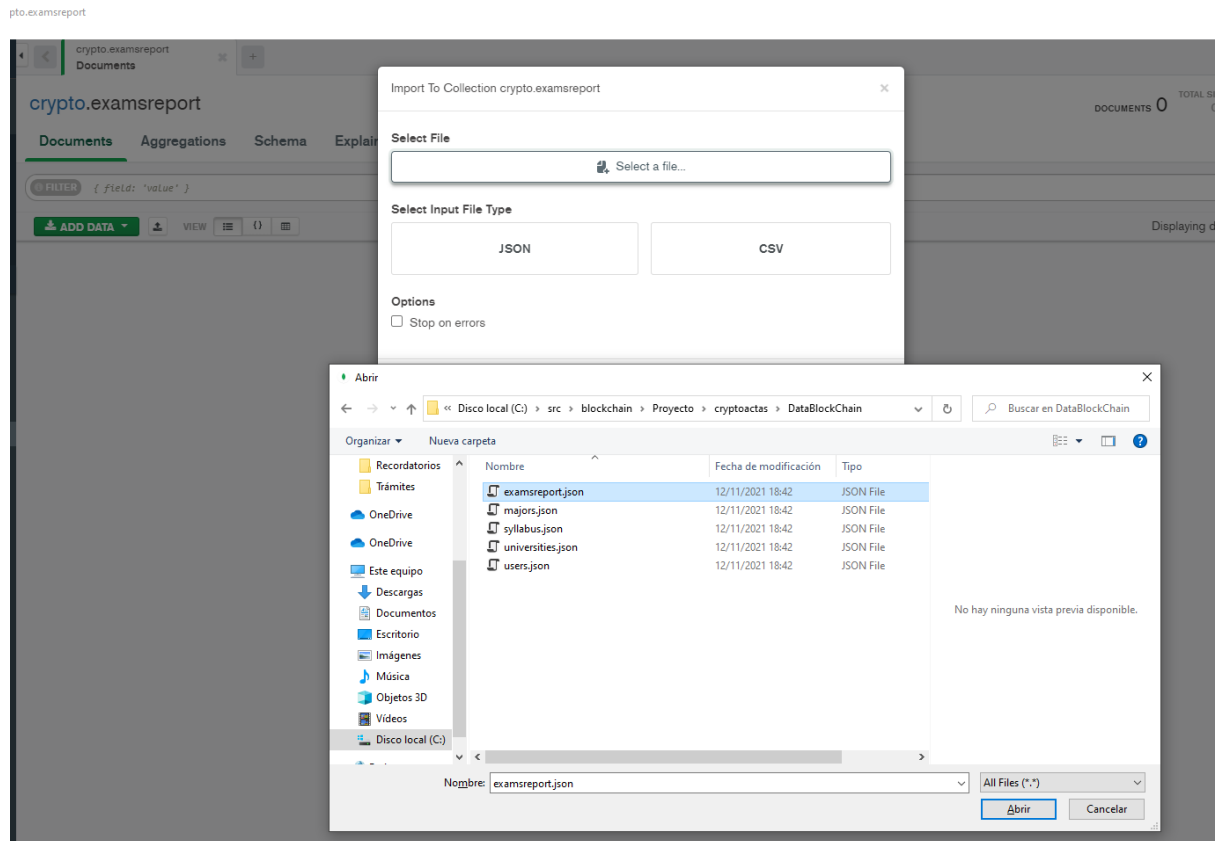
Dentro de este directorio se encuentran los archivos JSON necesarios para efectuar la carga inicial en la colecciones de base de datos de MongoDB.

13. Para cargar los datos iniciales de una colección se debe seleccionar la deseada y hacer clic en la opción *Add Data*. Se muestra a continuación un ejemplo en el que se ingresan los datos de la colección *examsreport*, a través de la opción importar archivo (*Import File*).

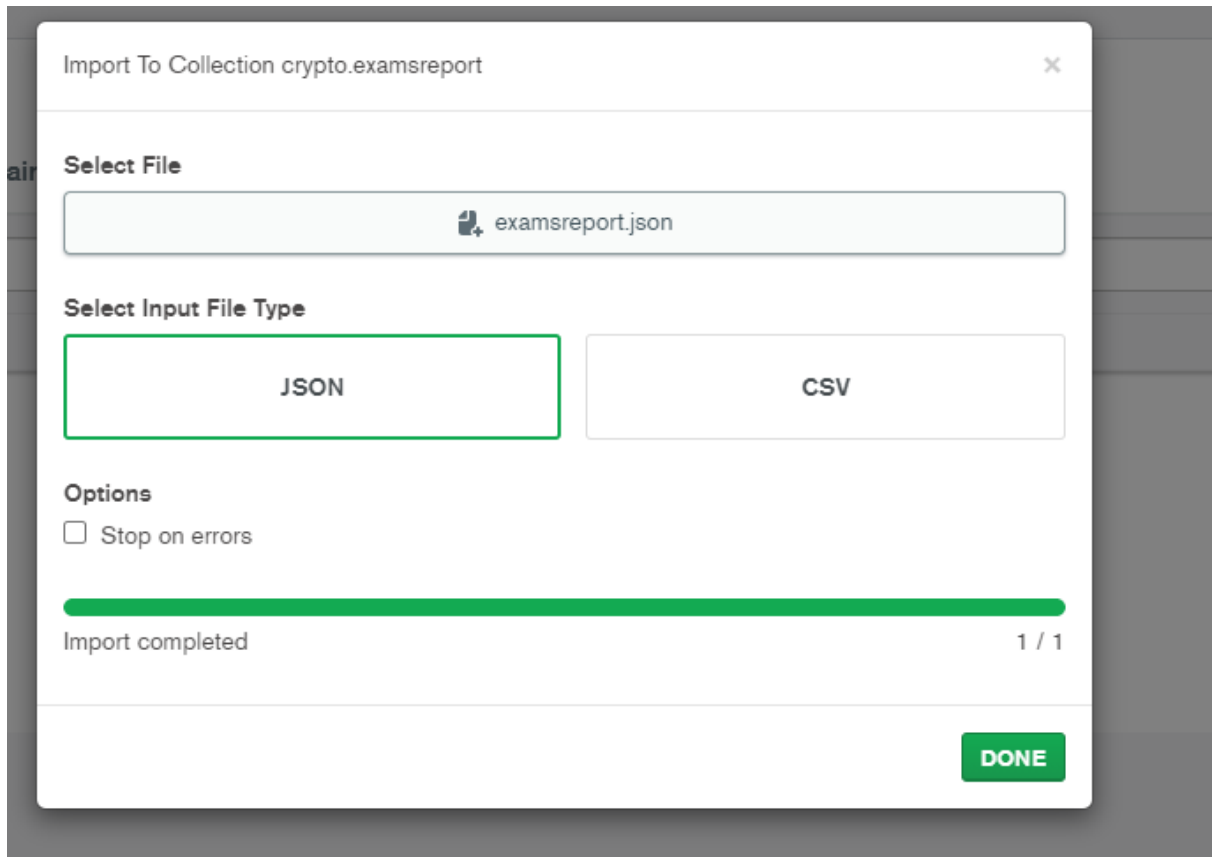
- a. Se selecciona Import File.



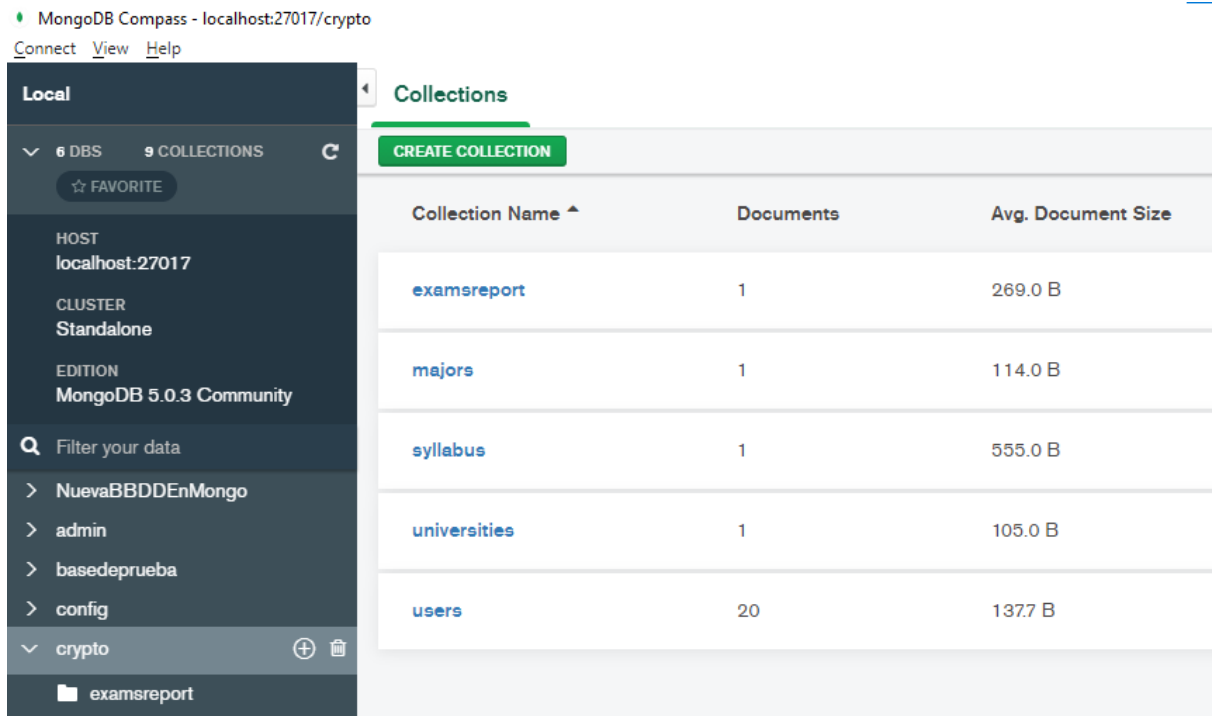
- b. Se selecciona el archivo con los datos.



c. Indicar formato JSON e importar.

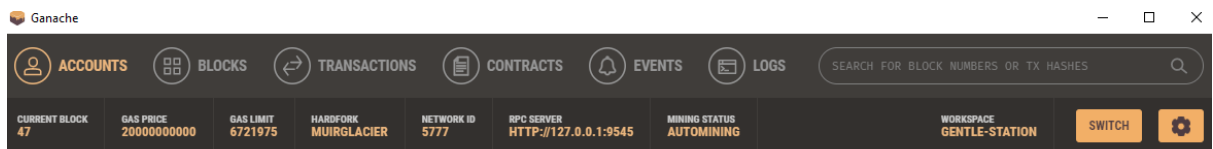


14. Repetir el paso 13 con las demás collections de la base de datos. De esta forma, se completa la carga inicial.
15. Una vez finalizado, debería quedar la base de datos crypto similar a la siguiente pantalla:



16. Instalar Ganache desde el sitio web oficial <https://www.trufflesuite.com/ganache>.

17. Ejecutar Ganache y crear un nuevo espacio de trabajo (Workspace) apuntando al puerto 9545, con Network ID 5777.



18. Posicionarse en la subcarpeta contracts, compilar y desplegar los smart contracts en una red de prueba local con los siguientes comandos:

```
cd contracts
    (ejemplo de la ubicación resultante C:\src\blockchain\Proyecto\cryptoactas\contracts)

truffle console
truffle migrate --reset
```

19. Ejecutar por línea de comandos la instrucción:

```
mongod
```

Esto habilita el servicio de acceso a la base de datos

```
C:\src\blockchain\Proyecto\cryptoactas>mongod
{"t":{"$date":"2021-11-17T20:35:10.200-03:00"},"s":"I",  "c":"NE
specification","attr":{"spec":{"incomingExternalClient":{"minWi
t":{"minWireVersion":0,"maxWireVersion":13},"outgoing":{"minWire
}}
{"t":{"$date":"2021-11-17T20:35:10.206-03:00"},"s":"I",  "c":"CO
```

No cerrar la ventana

20. Abrir otra terminal y posicionarse en la carpeta *server* del proyecto y ejecutar el comando:

```
npm start
```

Una vez levantado el server, permanecerá activo escuchando peticiones en la dirección <http://localhost:8080/>.

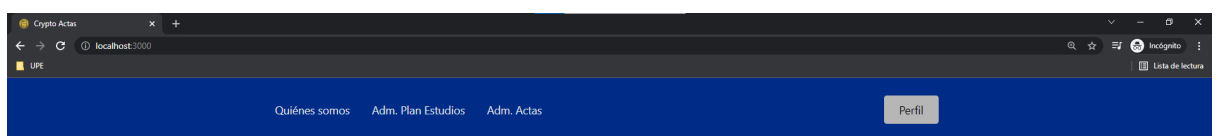
No cerrar la ventana

21. Abrir otra terminal y posicionarse en la carpeta *client* del proyecto y ejecutar el comando:

```
npm start
```

No cerrar la ventana

22. Con la ejecución del paso anterior, se abrirá una pestaña en el navegador con la aplicación cliente y su frontend desarrollado en React, ejecutándose en la dirección <http://localhost:3000/>.

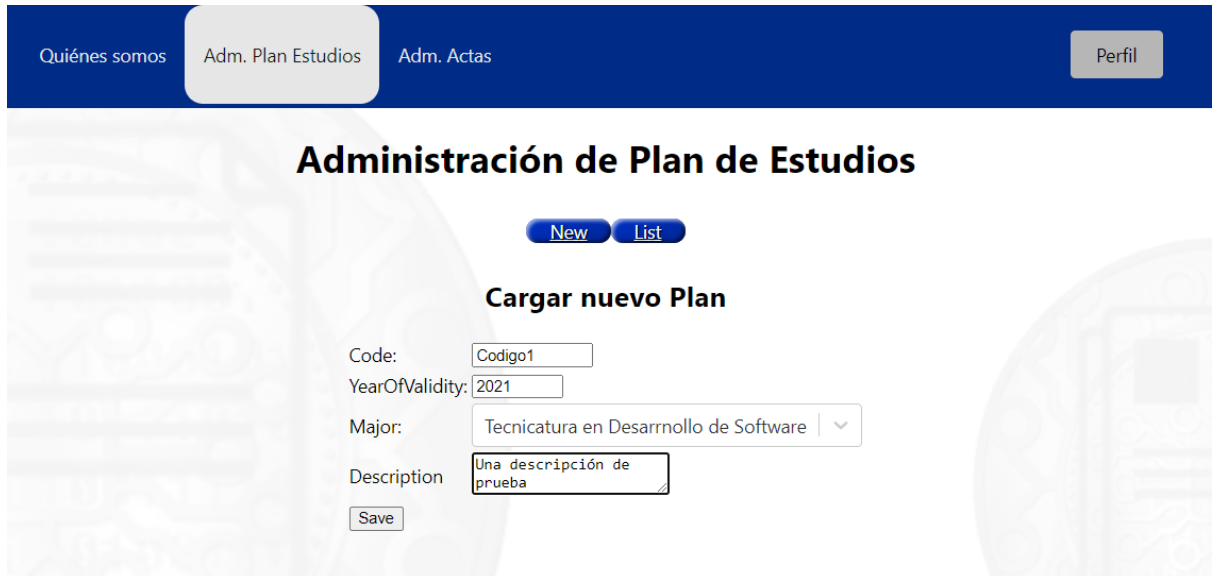


23. ¡Enhorabuena! Ha finalizado el proceso de instalación de cryptoactas 😊😊

Ejemplo de uso

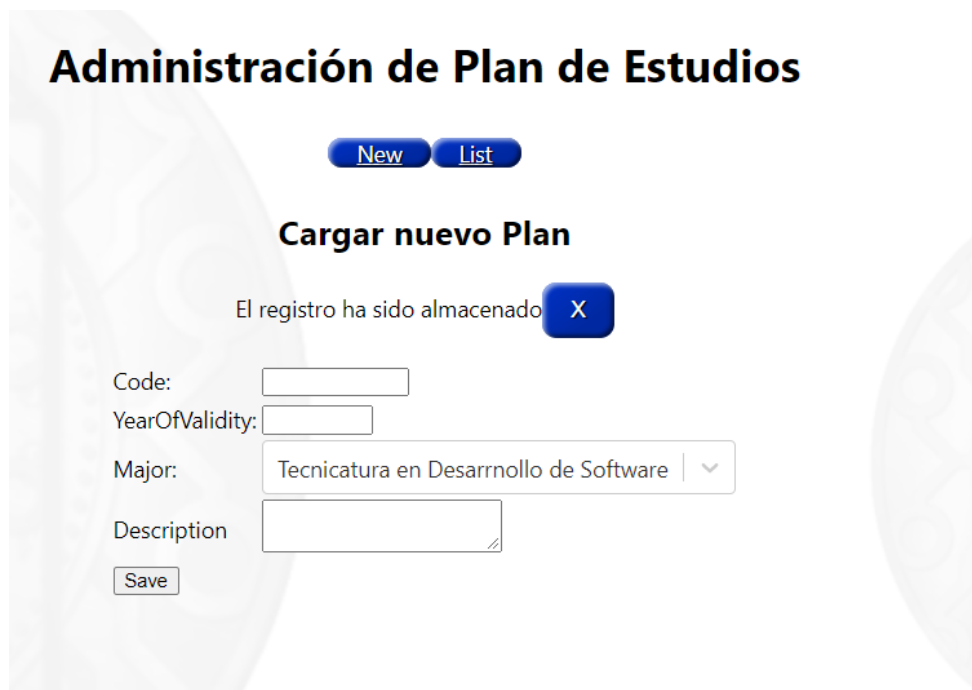
Generación de planes de estudio

1. Ubicado en el menú principal, hacer clic *Adm. Plan Estudios* y seleccionar *New* para agregar una nueva entrada en la base.



The screenshot shows the 'Administración de Plan de Estudios' interface. The top navigation bar includes 'Quiénes somos', 'Adm. Plan Estudios' (selected), 'Adm. Actas', and a 'Perfil' button. Below the navigation bar, the title 'Administración de Plan de Estudios' is displayed. There are two buttons: 'New' and 'List'. The 'Cargar nuevo Plan' section contains the following fields: 'Code:' with the value 'Codigo1', 'YearOfValidity:' with the value '2021', 'Major:' with a dropdown menu showing 'Tecnicatura en Desarrollado de Software', and 'Description' with the value 'Una descripción de prueba'. A 'Save' button is located at the bottom of the form.

2. Hacer clic en Save para persistir los datos en la base de MongoDB.



The screenshot shows the 'Administración de Plan de Estudios' interface after the 'Save' button was clicked. A success message 'El registro ha sido almacenado' is displayed next to a close button (X). The form fields are now empty: 'Code:', 'YearOfValidity:', 'Major:' (dropdown menu showing 'Tecnicatura en Desarrollado de Software'), and 'Description'. The 'Save' button is still visible at the bottom of the form.

- Para consultar los Planes de estudios almacenados en la base, se debe recurrir a la opción *List*.



Administración de Plan de Estudios

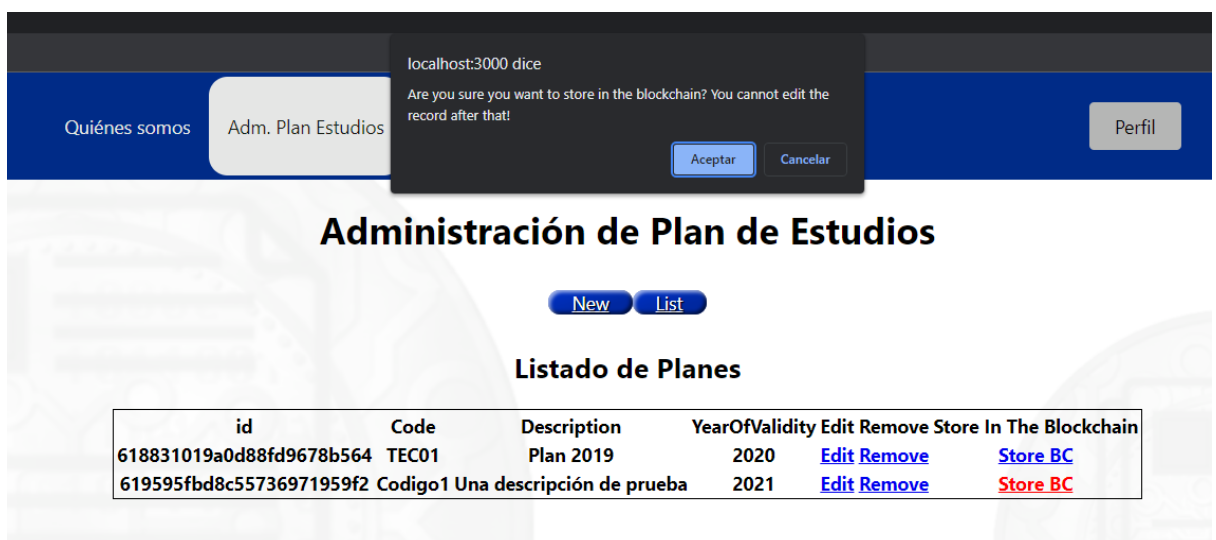
[New](#)
[List](#)

Listado de Planes

id	Code	Description	YearOfValidity	Edit	Remove	Store In The Blockchain
618831019a0d88fd9678b564	TEC01	Plan 2019	2020	Edit	Remove	Store BC
619595fbd8c55736971959f2	Codigo1	Una descripción de prueba	2021	Edit	Remove	Store BC

Persistencia de Plan de estudios en Blockchain

- Ubicado en el menú principal, hacer clic *Adm. Plan Estudios* y seleccionar *List* para consultar los planes cargados en la base de MongoDB.
- Seleccionar el plan deseado y hacer clic en la opción *Store BC*. Se desplegará un *pop up* solicitando confirmación de lo requerido.



localhost:3000 dice
Are you sure you want to store in the blockchain? You cannot edit the record after that!

[Aceptar](#)
[Cancelar](#)

Administración de Plan de Estudios

[New](#)
[List](#)

Listado de Planes

id	Code	Description	YearOfValidity	Edit	Remove	Store In The Blockchain
618831019a0d88fd9678b564	TEC01	Plan 2019	2020	Edit	Remove	Store BC
619595fbd8c55736971959f2	Codigo1	Una descripción de prueba	2021	Edit	Remove	Store BC

- Aparecerá en pantalla un botón titulado *Save in the BC*. Al hacer clic en el mismo, el dato se persistirá en forma definitiva en la Blockchain.

Administracion de Plan de Estudios

[New](#) [List](#)

Listado de Planes

[Save in the BC](#)

id	Code	Description	YearOfValidity	Edit	Remove	Store In The Blockchain
618831019a0d88fd9678b564	TEC01	Plan 2019	2020	Edit	Remove	Store BC
619595fbd8c55736971959f2	Codigo1	Una descripción de prueba	2021	Edit	Remove	Store BC

4. Comprobar que la cantidad de transacciones en Ganache se haya incrementado.