Issuer

1. Descripción:
   El issuer permite a distintas entidades autorizadas por el didi-server a generar y emitir certificados que podrán ser accedidos por los dueños de los mismos desde didi.
   Este módulo está compuesto por el **issuer-front**, un front-end desarrollado en reactjs y **issuer-back,** un backend desarrollado en nodejs con una base de datos local mongodb Donde se almacena la información de modelos de certificados y certificados a ser emitidos.

2. Ejecución
   a. Instalar Node:
      Este módulo utiliza npm como gestor de paquetes, el cual viene integrado con Node: https://nodejs.org/en/
   b. Instalar mongodb:
      El backend utiliza mongodb para almacenar la información de modelos de certificados y certificados a ser emitidos por el issuer:
      https://docs.mongodb.com/manual/installation/
   c. Instalar Robo3t (opcional):
      Se recomienda instalar algún gestor de bases de datos para mejor mantenimiento y detección de errores dentro de la misma:
      https://robomongo.org/
   d. Instalar dependencias:
      Abrir una terminal en la carpeta principal de **issuer-front** e **issuer-back** (a la altura del archivo 'package.json') y ejecutar el comando **'npm install'**
   e. Ejecutar **issuer-front** e **issuer-back**:
      A la misma altura, ejecutar el comando **'npm run start $ENV_VARS'**, donde $ENV_VARS son las distintas variables de entorno del módulo.
      Ej:
      Issuer-front

      'PORT=3502 REACT_APP_API_URL=http://localhost:3500/api/1.0/didi_issuer npm run start '

      Issuer-back

      'DEBUGG_MODE=true MONGO_DIR=127.0.0.1 MONGO_PORT=27017 MONGO_DB=didi_issuer ISSUER_SERVER_DID=0xf1232f840f3ad7d23fcdaa84d6c66dac24efb198 ISSUER_SERVER_PRIVATE_KEY=d8b595680851765f38ea5405129244ba3cbad84467d190859f4c8b20c1ff6c75 ADDRESS=http://192.168.2.113 PORT=3500 DIDI_API=http://192.168.2.113:3000/api/1.0/didi npm run start'

f. Variables de entorno:

issuer-front

Nombre	Obligatorio	Descripción	notas	ej
PORT	Si	*Puerto en que corre issuer-front*	-	*3502*
REACT_APP_API_URL	Si	*dirección en la que está corriendo issuer-back*	-	*http://localhost:3500/api/1.0/didi_issuer*

issuer-back

Nombre	Obligatorio	Descripción	notas	ej
DEBUGG_MODE	No	*si está en true se muestran más cosas por consola*	-	*true*
MONGO_DIR	Si	*dirección en la que está corriendo mongo*	*La instancia de mongodb no necesariamente es la misma que para el didi-server!!*	*127.0.0.1*
MONGO_PORT	Si	*puerto en que corre mongo*	-	*27017*
MONGO_USERNAME	No	*usuario admin en mongo*	*se lo puede omitir si la base de datos no requiere autenticación (inseguro!)*	*didi_admin*
MONGO_PASSWORD	No	*clave de usuario admin en mongo*	*se la puede omitir si la base de datos no requiere autenticación (inseguro!)*	*uIERvZXMiLCJpYXQiOjE1MTYyMzkwMjJ9*
MONGO_DB	Si	*nombre de la base de datos dentro de mongo*	-	*didi_issuer*
ISSUER_DELEGATOR_DID	No	*Nombre del delegador en caso que el issuer no esté autorizado a emitir certificados, pero haya sido autorizado por alguien que si lo esta.*	-	*0x36Fdb5032d8e42b8cd14C70A9c7Aef4a6086D8a3*
ISSUER_SERVER_DID	Si	*did del issuer-back*	-	*0xf1232f840f3ad7d23fcdaa84d6c66dac24efb198*

ISSUER_SERVER_PRIVATE_KEY	Si	*clave privada del issuer-back*	-	*d8b595680851765f38ea5405129244ba3cbad84467d190859f4c8b20c1ff6c75*
ADDRESS	Si	*dirección en que corre issuer-back*	*Se usa únicamente para el callback del código qr*	*http://192.168.2.113*
PORT	Si	*puerto en que corre issuer-back*	-	*3500*
DIDI_API	Si	*dirección en la que está corriendo didi-server*	*Para emision y validacion de certificados*	*http://192.168.2.113:3000/api/1.0/didi*
BLOCK_CHAIN_URL	Si	*Url donde se encuentra corriendo la blockchain*	-	*https://did.testnet.rsk.co:4444*
BLOCK_CHAIN_CONTRACT	Si	*Identificador del contrato de ethr-did-registry dentro de la blockchain*	-	*0xdca7ef03e98e0dc2b855be647c39abe984fcf21b*
BLOCK_CHAIN_DELEGATE_DURATION	no	*Tiempo en segundos durante el cual una delegación de did es válida.*	*Default 1300000*	*1300000 (1año)*
BLOCK_CHAIN_SET_ATTRIBUTE	no	*Tiempo en segundos durante el cual una registración de nombre en la blockchain será vigente.*	*Default 999999999*	*999999999*
HASH_SALT	Si	*Salt a ser usado al encriptar datos indexables (mail, tel)*	-	*$2b$11$ggDdDiXNBkuEiQWTdHQ.hu*
RSA_PRIVATE_KEY	Si	*Clave privada a utilizarse al encriptar los datos con rsa*	-	*Ver abajo...*

RSA_PRIVATE_KEY:

-----BEGIN RSA PRIVATE KEY-----
MIIBOwIBAAJBAO64JHU8uFH4ZsmhbkGUFysU/GktjV3vT84TfGjKljw6npCbFMeN
QZmmSa+TQb+oYurlO2YOaw5GI4LfQbkjnpcCAwEAAQJACEWrRcC/5l9EOJJ2fqFC
GEUW5hIlBu9HBa4ZLoxqmUmhmPhov5if2ga1IE9vpcTOzmGIMW/LPBGS184wW6tG
IQIhAPfwR5vkrSYkDudw6Sp5uwsTv1t2sMT3bWwhDJDquv1RAiEA9nsgNrc1Y95c
+YVHr6e4v4bZYcxsHV5WE8ZCC5EDw2cCIQCmPZ9l8W//UNIFcHmGF1TIWpdFlIFz
34qoo4gvapOx4QIhAIdVc4qXbak4HrSiiYnY/Yer8w/Pvk0hzwFsijbvmLFhAiBB
JoZfex8gJUKiNfKXYtJkBFy2fTOQ7c1iEDS6hnIKSA==
-----END RSA PRIVATE KEY-----

3. Activar autenticación en base de datos mongo (opcional):

Lanzar servicio de mongo:

```
mongod
```

Arrancar cliente de mongo:

```
mongo
```

En el cliente, crear usuario root:

```
> use admin
> db.createUser(
  {
    user: "administrator",
    pwd: "$PASSWORD",
    roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]
  }
)
```

Volver a lanzar el servicio de mongo con la opción --auth (activar validación de usuarios)

```
mongod --auth
```

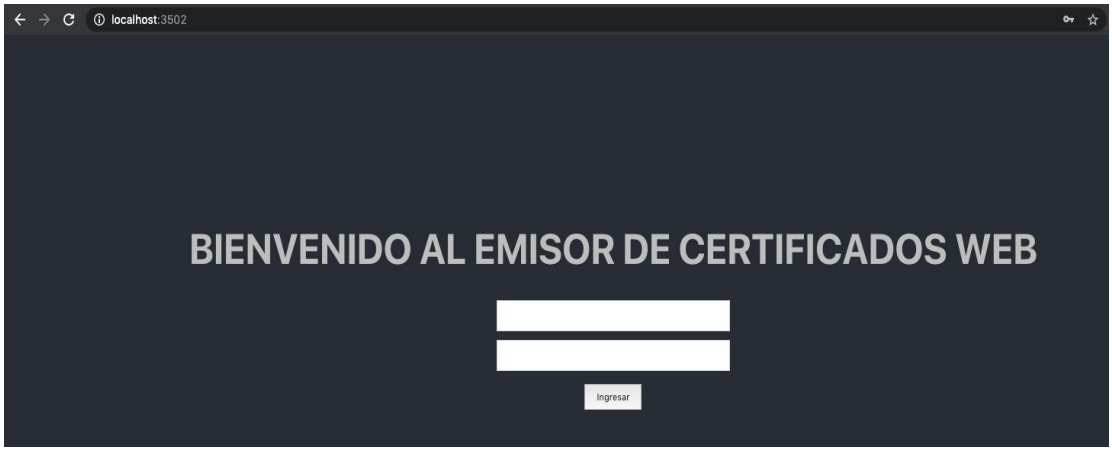
Arrancar cliente de mongo con el usuario previamente creado:

```
mongo -u "administrator" -p "$PASSWORD" --authenticationDatabase "admin"
```

En el cliente, crear usuario para la base de datos del issuer:

```
> use didi-issuer
> db.createUser({
        user: "$MONGO_USERNAME",
        pwd: "$MONGO_PASSWORD",
        roles: [
                {
                        role: "readWrite",
                        db: "didi-issuer"
                }
        ]
});
```

4. Una vez iniciado, en el **Issuer-front** se mostrará la url desde la cual se puede acceder al módulo**:**



En estos momentos, para crear el usuario es necesario realizar una llamada al **Issuer-back** (esto puede cambiar en el futuro), el método a llamar es el “/user”, y se debe pasar en el body el nombre de usuario y contraseña.

Ej: utilizando “curl”

```
curl -d '{"name":"admin", "password":"DRFNTFBM9ThkNGJCME"}' -H "Content-Type: application/json" -X POST http://localhost:3500/api/1.0/didi_issuer/user
```

Una vez logueado, se verán tres secciones:

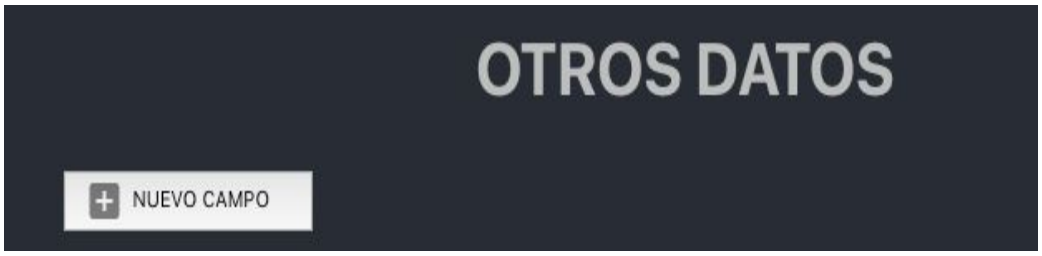
- ‘Templates’ o ‘Modelos de certificados’: Permite generar modelos de certificados genéricos, fijando los datos que contendrán los mismos y la forma en que se visualizan los mismos en la aplicación.

Al editar un template se muestran varias secciones para las que se pueden agregar campos genéricos, decidir si estos serán obligatorios u opcionales y definirles valores por defecto.

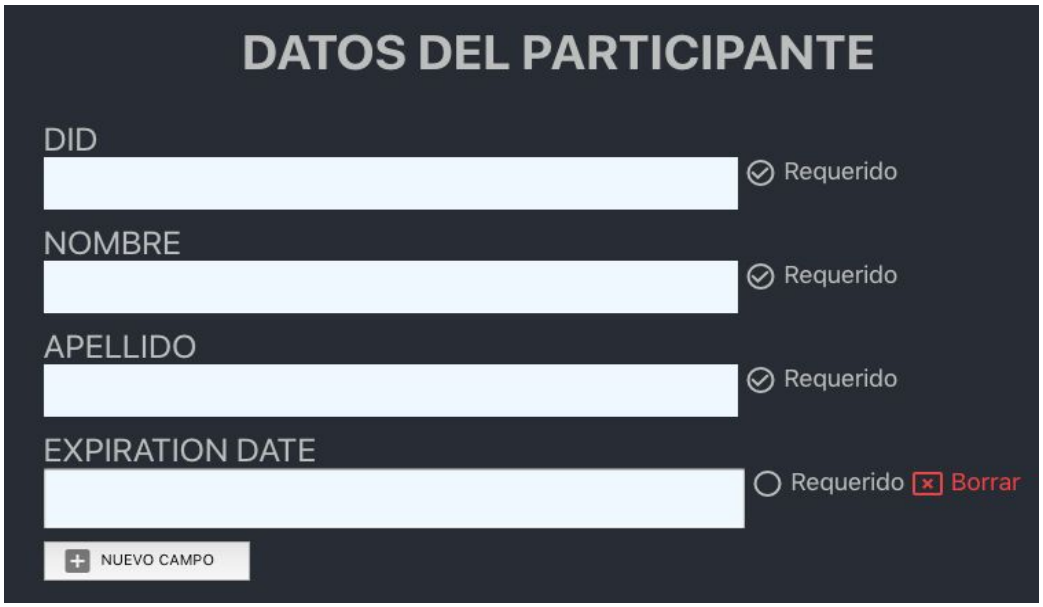
Las secciones son:

- Campos a previsualizar: Permite elegir entre varias formas en que puede mostrarse el certificado emitido en la aplicación y también permite definir que campos se mostrarán por defecto en el certificado, mostrándose el resto solo en el detalle del certificado.

- Datos del Certificado y Otros Datos:
  Definen información global del certificado, si se emiten certificados “en batch” todos los certificados tendrán en estos campos los mismos datos.



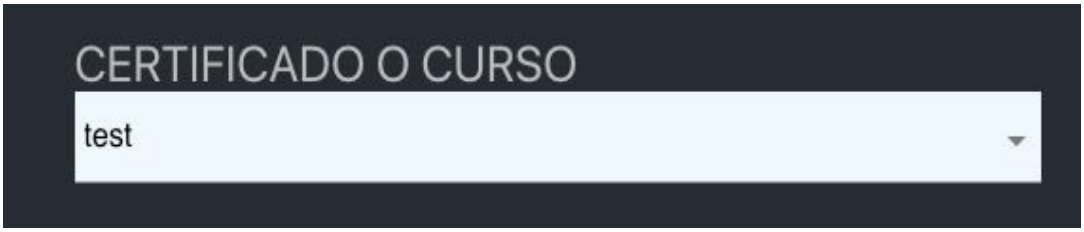
- Datos del Participante:
  Definen información del dueño del certificado si se emiten certificados “en batch”, son los datos que serán únicos para cada certificado particular.



- ‘Certificados’: Permite crear, modificar, emitir y revocar certificados partiendo de un template previamente creado en ‘Templates’.

Al editar un certificado se muestran varias secciones:

  - ‘Selección deTemplate’: Permite definir el modelo de certificado que se desea emitir



  - ‘Selección de microcredenciales’: Permite definir si se emitirán credenciales completas o se las dividirá en microcredenciales, Permitiendo seleccionar el nombre y los datos que irán en cada una de esas microcredenciales.

- Datos del Certificado, Datos del Participante y Otros Datos:
  Definen los datos de las credenciales a emitir, son análogas a las del template, con la diferencia que en la sección “Datos del Participante”, se cargará un set de datos por cada certificado a emitir.

- También existen opciones para cargar los datos del certificado a partir de un csv, generar un csv de ejemplo, cargar participante desde código qr o cargar participante previamente guardado en la sección “Participantes”

- ‘Participantes’: Permite realizar la carga por qr o perdido de datos a usuarios de didi para tenerlos precargados al momento de completar los certificados.