

## Instrucciones para personalizar CiupCoin

---

### 1. Nombre de Moneda:

Renombre la Carpeta CiupCoin con el nombre que desee, e.g. AbcCoin, y ábrala desde un editor de texto. Desde allí, cambie el nombre del archivo `CiupCoin-qt.pro` a `AbcCoin-qt.pro`. También modifique los nombres dentro de los siguientes archivos:

- `LearnCoinqt.pro`
- `/src/bitcoinrpc.cpp`
- `/src/irc.cpp`
- `/src/init.cpp`
- `/src/main.cpp`
- `/src/net.cpp`
- `/src/rpcdump.cpp`
- `/src/util.cpp`

### 2. Puertos:

Puedes modificar algún puerto para evitar el conflicto con otras aplicaciones, solo modifique lo siguiente:

(a) Edita `.../AbcCoin/src/Bitcoinrpc.cpp`

i. Línea 2892:

```
ip::tcp::endpoint endpoint(bindAddress, GetArg("-rpcport", 34332));
```

ii. línea 3168

```
if(!d.connect(GetArg("-rpconnect", "127.0.0.1"),GetArg("-rpcport", "34332")))
```

(b) Edita `.../AbcCoin/src/init.cpp`

i. Línea 235:

```
" -port=<port>          " + _("Listen for connections on <port>  
(default: 34332 or testnet: 36337)") + "\n" +
```

### 3. Recompensas:

Modificamos el monto de la coins que se recompensa al minero por cada bloque minado.

(a) Edita `.../AbcCoin/src/main.cpp`

i. línea 831

```
int64 nSubsidy = 10 * COIN;
```

ii. línea 841

```
static const int64 nTargetTimespan = 4 * 60;
```

iii. línea 842

```
static const int64 nTargetSpacing = 30;
```

En las dos últimas líneas puedes modificar el TargetTimespan que sirve para determinar el tiempo en el que se reajusta el minado de acuerdo a la cantidad de coins emitidos, y el TargetSpacing que es el tiempo ideal que le debería tomar a un bloque ser encontrado.

### 4. Confirmaciones:

Modificamos el número mínimo de confirmaciones que debe tener un bloque para ser aceptada en la cadena de bloques, para este ejemplo son 4; recuerda que nunca deben ser menor que 3.

(a) Edita `.../AbcCoin/src/main.h`

i. Línea 45:

```
static const int COINBASE_MATURITY = 3;
```

(b) Edita `.../AbcCoin/src/main.cpp`

i. Línea 701:

```
return max(0, (COINBASE_MATURITY+1) - GetDepthInMainChain());}
```

(c) Edita `.../AbcCoin/src/ qt/transactionrecord.h`

i. Línea 73:

```
static const int NumConfirmations = 3;
```

### 5. Árbol Merkle

(a) Extraiga la fecha en Unix. En el ejemplo de abajo es: 1509667324.

```
~$ date +%s  
> 1509667324
```

(b) Copie el resultado y en: `.../AbcCoin/src/main.cpp` reemplace las siguientes líneas.

i. línea 2032

```
block.nTime = 1509667324;
```

ii. línea 2038

```
block.nTime = 1509667324;
```

(c) Para crear el árbol Merkel ubíquese en la carpeta `.../AbcCoin/src` desde el terminal y ejecute:

```
~$ make -f makefile.unix  
~$ ./AbcCoin
```

- (d) Se desplegará un mensaje de error referente al hash del árbol. Debes dirigirte al directorio del sistema operativo donde se habrá creado una carpeta llamada `.AbcCoin` que se puede abrir con cualquier editor de texto. Dentro de la carpeta encontrarás un archivo llamado `debug.log` con el siguiente contenido:

```
AbcCoin version v0.6.3.0-unk-beta (2017-10-12 14:11:05 -0500)
Startup time: 11/15/17 08:21:32
Default data directory /home/linux/.AbcCoin
Used data directory /home/linux/.AbcCoin
Bound to [::]:36329
Bound to 0.0.0.0:36329
Loading block index...
dbenv.open LogDir=/home/linux/.AbcCoin/database ErrorFile=/home/linux/.AbcCoin/db.log
a1df3f68cf658bdd4bccf1176e3abca57c8e0d5c4a66d7904cbcb3af288ef2d6
3bd3bc36f1eda38897e79b3fed3b42fffcf266d2b522093f47f30cd681b84614
be47273044f018b7a78a562d05334114e02757dd1c42c8e198015011b29b29fe
```

- (e) Edita `.../AbcCoin/src/main.cpp`

- i. Línea 2047:

```
assert(block.hashMerkleRoot ==
uint256("0xbe47273044f018b7a78a562d05334114e02757dd1c42c8e198015011b29b29fe"));
```

- ii. línea 2049

```
if (true && block.GetHash() != hashGenesisBlock)
```

- (f) Recompilas el código:

```
~$ make -f makefile.unix
~$ ./AbcCoin
```

- (g) Se mostrara otro mensaje de error referente al hash del árbol, así que te diriges archivo `debug.log` que contiene lo siguiente:

```
block.nTime = 1509667324
block.nNonce = 2087879834
block.GetHash = f16ab9cede99d863f8568b8e46330a9ca0fe8ad30d26da4aeb79a017dcd68eb1
CBlock(hash=f16ab9cede99d863f856, PoW=000009097b37e273f950, ver=1,
hashPrevBlock=00000000000000000000, hashMerkleRoot=be47273044,
nTime=1509667324, nBits=1e0ffff0, nNonce=2087879834, vtx=1)
CTransaction(hash=be47273044, ver=1, vin.size=1, vout.size=1, nLockTime=0)
CTxIn(COutPoint(0000000000, -1),
coinbase 04ffff001d0104147072696d657220646961206465204a6b436f696e)
CTxOut(error)
vMerkleTree: be47273044
```

Copias el `block.nNonce` y el `block.GetHash` y los rempazan en los siguiente archivos:

- i. `.../AbcCoin/src/main.cpp`

- A. Línea 32:

```
uint256 hashGenesisBlock(
"0xf16ab9cede99d863f8568b8e46330a9ca0fe8ad30d26da4aeb79a017dcd68eb1");
```

- B. línea 2034

```
block.nNonce = 2087879834;
```

C. línea 2039

```
block.nNonce = 2087879834;
```

D. línea 2039

```
if (false && block.GetHash() != hashGenesisBlock)
```

ii. .../AbcCoin/src/checkpoints.cpp

A. Línea 27:

```
(0,uint256(
"0xf16ab9cede99d863f8568b8e46330a9ca0fe8ad30d26da4aeb79a017dcd68eb1"))
```

## 6. Compartir en Github

- (a) Para subir a GitHub debes tener una cuenta y crear allí un repositorio llamado AbcCoin y luego ejecutar desde la terminal de tu computadora lo siguiente:

```
~$ rm -r -f .git
~$ git init
~$ git add -A *
~$ git commit -m "My AbcCoin"
~$ git remote add origin https://github.com/yourusername/AbcCoin.git
~$ git push origin master
```

- (b) Para iniciar el minado siga los pasos descritos en:  
<https://github.com/franciscorosales-marticorena/CiupCoin>