

# Blockchain y Criptomonedas II

UCEMA - QUANT

Clase II: 12/11/25

# Agenda

- Módulo V: Librería asyncio para Python
- Módulo VI: Discusión de la tarea propuesta
- Break
- Modulo VII: Repaso de mercados centralizados
- Modulo VIII: API de Binance
- Módulo IX: Librería CCXT
- Conclusión y resumen
- Tarea propuesta

# Módulo V

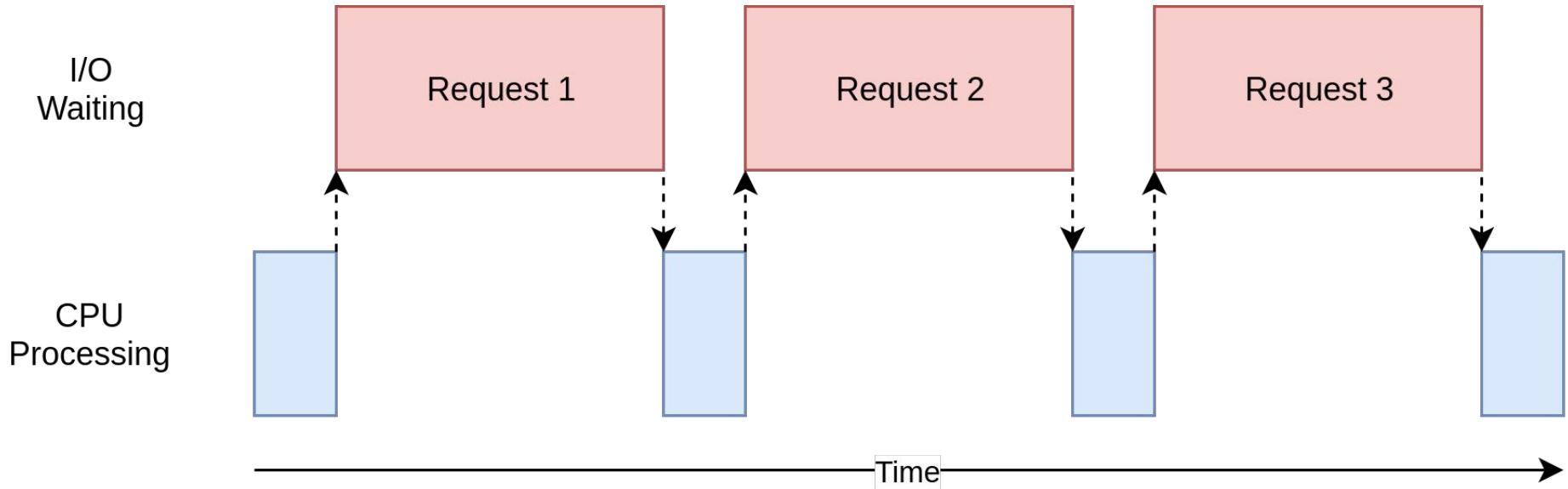
## Librería asyncio para Python

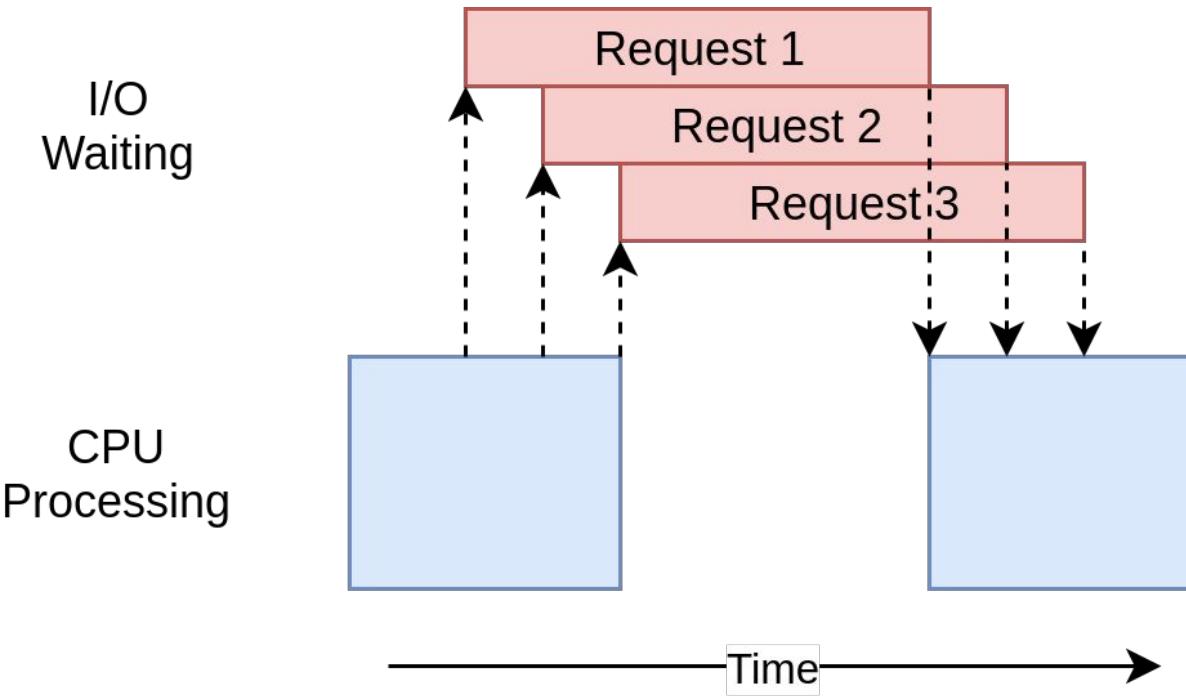
## 5.1 ¿Qué es asyncio?

- Librería de Python que permite escribir código **concurrente**
- Código concurrente:
  - Permite comenzar una operación antes de que finalice la anterior!
  - Se busca que las tareas se dividen en sub tareas que se puedan ejecutar en simultáneo.
  - Muy utilizado para:
    - I/O
    - Network requests

- Ejemplo: conectarse a n exchanges para obtener información de precios instantánea
  - Caso sincronico:
    - Request al exchange 1
    - Response del exchange 1
    - ...
    - Request al exchange n
    - Response del exchange n
  - Caso asincrónico:
    - Request al exchange 1, 2, ..., n
    - Respuesta del exchange 1, 2, ..., n
  - Comparación

## 4.2 Comparación de performance





# Módulo VI

## Discusión de las tarea propuesta

# Break

# Módulo VII

# Repaso de mercados centralizados

# 7.1 Mercados centralizados

- ¿Qué son?
  - Son mercados con reglas más parecidas a los mercados financieros tradicionales
- Características
  - Funcionan con libros de órdenes
  - Se puede operar las 24 horas del día
  - Permiten operar:
    - *Spot*: settlement instantáneo, de monedas. Similar al FX
    - *Margin*: permite operar tomando margen prestado del mercado. Se colateraliza la posición con otra moneda o con la cartera
    - *Derivatives*: tienen una sección de derivados donde se pueden encontrar generalmente:
      - Futuros con delivery en *token*
      - Futuros con delivery en *stablecoin*
      - Futuros perpetual, sin delivery

- Perpetuals
  - Son futuros a perpetuidad, sin vencimiento
  - Siguen a un índice. Por ejemplo, *ETH/USDT*.
  - Tienen un *funding rate* asociado. El cálculo de este funding rate depende de cada protocolo
    - Busca hacer converger el precio del perpetual y el precio del índice
    - Por ejemplo: si muchos usuarios están *long* en el perpetual, el precio va a tender a subir probablemente por encima del índice. Eso genera un valor de funding rate tal que los usuarios que están *long* le van a pagar a los usuarios que están *short*. Por otro lado, el funding rate incentiva a los usuarios a tomar la posición contraria para recibir esta tasa.
- ¿Ventajas y diferencias del *margin* y los futuros?
- ¿Ideas de trading con esto?
  - ¿Futuros con delivery?
  - ¿*Perpetuals*?

## 7.2 Comparación entre Dexes y Cexes

	Centralized Exchanges	Decentralized Exchanges
Privacidad	KYC	P2P
Seguridad	Discutible en ambos casos. Historial de hackeos y estafas en ambos lados	
Liquidez	Mayor liquidez, menos variedad	Menor liquidez, más variedad
Fees	Más caros	Más baratos
Velocidad	Muchísimo más rápido	Lento
UX	Más amigable para aquellos acostumbrados al trading tradicional	Hoy en día, suficientemente amigable para un usuario convencional
Trading	Contra el exchange	“Contra otro usuario”
Marco regulatorio"	“Sujeto a regulaciones”	No regulado, a veces auditado

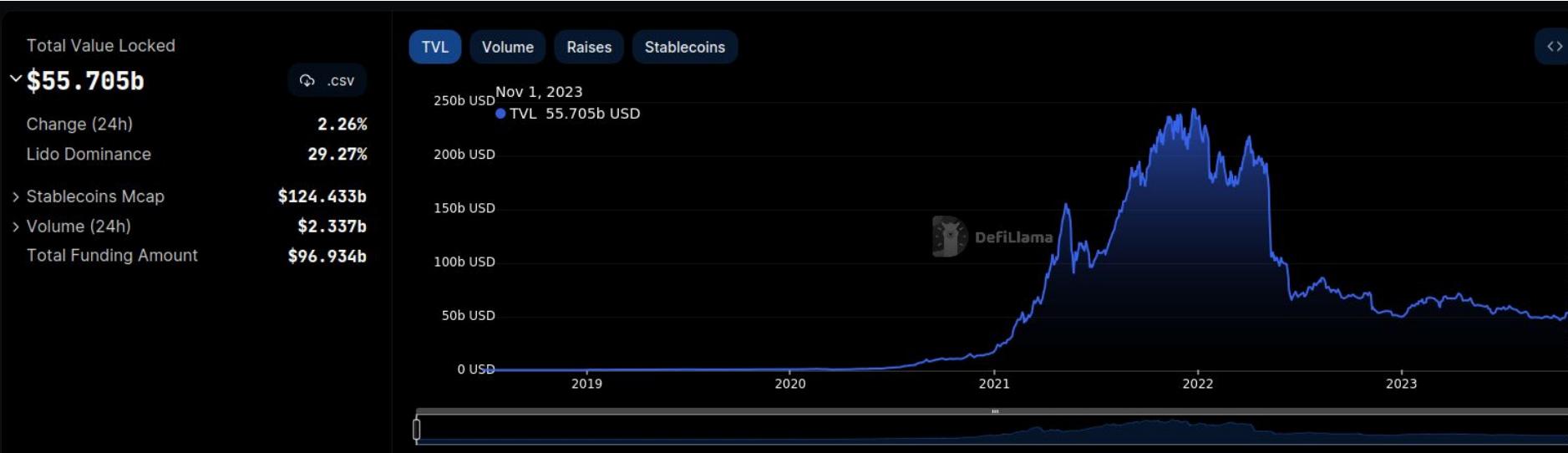
Fuente: [post](#)

- Defillama.com

- Cexes:

Name	Assets ⚡	Clean Assets ⚡	24h Inflows ⚡	7d Inflows ⚡	1m Inflows ⚡	Custom range Inflows ⚡	Auditor ⚡	Last audit date ⚡	Spot Volume ⚡
1 Binance	\$65.971b	②\$57.908b	\$95.38m	\$695.28m	-\$142.33m	-\$16.43m			\$7.629b
2 OKX	\$11.297b	②\$11.297b	\$24.16m	-\$59.53m	\$620.32m	\$562.86m			\$2.305b
3 Bitfinex	\$10.811b	②\$8.268b	-\$75.36m	-\$77.06m	\$24.45m	-\$14.41m			\$139.45m
4 Bybit	\$4.106b	②\$4.106b	-\$9.95m	-\$41.19m	\$143.58m	\$149.87m			\$2.384b
5 Kraken	\$3.619b	②\$3.619b	\$68.88m	-\$57.81m	\$17.49m	-\$559,975			\$881.74m

- Dexes:



# Módulo VIII

## API de Binance

# 8.1 Documentación

- Documentación
- Generación de API Key
- Métodos:
  - Publicos, no autenticados
  - Privados, autenticados
    - Cómo autenticar:
      - Tomar el body del request. Por ejemplo:  
`"symbol=LTCBTC&side=BUY&type=LIMIT&timeInForce=GTC&quantity=1&price=0.1&recvWindow=5000&timestamp=1499827319559"`
      - Usar el API Key como clave para firmar con openssl  
`openssl dgst -sha256 -hmac "NhqPtmdSJYdKjVHjA7PZj4Mge3R5YNiP1e3UZjlnClVN65XAbvqqM6A7H5fATj0j"`
      - Usar eso en el header del request bajo el nombre X-MBX-APIKEY
    - Referencia

## 8.2 Métodos Públicos

- Son aquellos que involucran:
  - Obtener información pública
    - Precios
    - *Order book*
    - Información de los mercados disponibles
    - Mining
    - Staking
    - Funding rates
    - ...

## 8.3 Métodos Privados

- Son todos aquellos que involucran:
  - Movimientos de fondos
  - Consulta de balances
  - *Trading*
  - Manejo de subcuentas
- Requieren autenticación

# Módulo IX

## Librería CCXT

## 9.1 ¿Qué es CCXT?

- Librería de Python que permite conectarse con múltiples exchanges
- API Rest o Websockets (premium)
- Estandarización!

```
exchange_id = 'binance'
exchange_class = getattr(ccxt, exchange_id)
exchange = exchange_class({
    'apiKey': 'YOUR_API_KEY',
    'secret': 'YOUR_SECRET',
})

hitbtc_markets = hitbtc.load_markets()

print(hitbtc.id, hitbtc_markets)
print(bitmex.id, bitmex.load_markets())
print(huobipro.id, huobipro.load_markets())

print(hitbtc.fetch_order_book(hitbtc.symbols[0]))
print(bitmex.fetch_ticker('BTC/USD'))
print(huobipro.fetch_trades('LTC/USDT'))

print(exmo.fetch_balance())
|
# sell one ₩ for market price and receive $ right now
print(exmo.id, exmo.create_market_sell_order('BTC/USD', 1))

# limit buy BTC/EUR, you pay €2500 and receive ₩1 when the order is closed
print(exmo.id, exmo.create_limit_buy_order('BTC/EUR', 1, 2500.00))

# pass/redefine custom exchange-specific order params: type, amount, price, flags, etc...
kraken.create_market_buy_order('BTC/USD', 1, {'trading_agreement': 'agree'})
```

# Conclusión y resumen

# Conclusión y resumen

- Módulo V: Librería asyncio para Python
  - Comentario rápido, ventajas de asyncio
- Módulo VI: Discusión de la tarea propuesta
  - Implementación de la interfaz
  - Muchos exchanges son forks de uni! es un buen lugar para empezar
- Modulo VII: Repaso de mercados centralizados
  - Gran tamaño, gran liquidez, gran velocidad. Menos privacidad
- Modulo VIII: API de Binance
  - Integración, métodos públicos y privados
- Módulo IX: Librería CCXT
  - Definición
  - Implementación interfaz
  - Ejemplos!

# Tarea propuesta

# Tarea propuesta

- Desarrollar un servicio que, ante cada evento de swap generado por un par determinado (por ejemplo, wMATIC/USDC), muestre el precio actual del par en Sushiswap, Quickswap, Binance, y OKX.
- Guía:
  - Para obtener la dirección del pool y de los tokens:
    - Crear el contrato factory de UNIV2 para Sushiswap y QuickSwap
    - Usar el método getPair con el address de cada uno de los tokens
    - Para obtener el address de los tokens:
      - En sushiswap, ir a la sección de pools
      - Filtrar por polygon
      - Buscar el pool wMATIC/USDC, y ahí está la información
  - Después, crear un filtro de web3 con el evento de swap y los addresses del par de los dos protocolos
  - Finalmente, en cada evento, imprimir los precios de los 4 exchanges

